

RocketMQ学习笔记

MQ介绍

- 为什么要用MQ
 - 应用解耦
 - 流量削峰
 - 数据分发
- MQ的优点和缺点

RocketMQ快速入门

- 下载RocketMQ
- 安装
- 目录介绍
- 配置环境变量
- 启动RocketMQ
 - 启动NameServer
 - 启动Broker
- 测试RocketMQ
 - 发送消息
 - 接收消息
- 关闭RocketMQ

RocketMQ集群搭建

- 各角色介绍
- 集群搭建方式
 - 集群特点
 - 集群模式
 - 单Master模式
 - 多Master模式
 - 多Master多Slave模式（异步）
 - 多Master多Slave模式（同步）
- 双主双从集群搭建
 - 集群工作流程
 - 创建消息存储路径
 - broker配置文件
 - master1
 - slave2
 - master2
 - slave1
 - 服务启动
 - 启动NameServer
 - 启动Broker集群
 - 启动脚本
 - 查看进程状态

mqadmin管理工具

- 集群监控平台搭建
 - 克隆开源项目
 - 配置集群地址
 - 打包
 - 启动
 - 访问
- 命令介绍
 - Topic相关
 - 集群相关

- Broker相关
- 消息相关
- 消费者、消费组相关
- 连接相关
- NameServer相关
- 其他

消息发送与消费

- RocketMQ客户端依赖

- 基本消息的发送与接收

- 消息发送

- 发送同步消息

- 发送异步消息

- 单向发送消息

- 消费消息

- 负载均衡模式

- 广播模式

- 顺序消息

- 生产

- 消费

- 延时消息

- 生产者

- 消费者

- 启动

- 批量消息

- 生产

- 消费

- 启动

- 过滤消息

- SQL基本语法

- 消息生产者

- 消息消费者

- 启动测试

- 消息生产者

- 消息消费者

- 启动测试

- 事务消息

- 流程分析

- 事务消息发送及提交

- 事务补偿

- 事务消息状态

- 发送事务消息

- 创建事务性生产者

- 实现事务的监听接口

- 接收事务消息

- 启动测试

- 使用限制

整合spring boot

- 依赖

- Yml文件配置

- 消息发送

- 发送同步消息

- 发送批量消息

- 发送带tag的消息

- 发送同步消息，并指定队列

- 发送异步消息

- 发送单向消息

- 消息消费

高级功能

消息存储

存储介质

关系型数据库

文件系统

性能对比

消息的存储和发送

消息存储

消息发送

消息存储结构

刷盘机制

同步刷盘

异步刷盘

高可用性机制

消息消费高可用

消息发送高可用

消息主从复制

总结

负载均衡

Producer负载均衡

Consumer负载均衡

集群模式

广播模式

消息重试

顺序消息的重试

无序消息的重试

重试次数

配置方式

死信队列

死信特性

消费幂等

消费幂等的必要性

处理方式

小结

MQ介绍

为什么要用MQ

消息队列是一种“先进先出”的数据结构

其应用场景主要包含以下3个方面

应用解耦

系统的耦合性越高，容错性就越低。以电商应用为例，用户创建订单后，如果耦合调用库存系统、物流系统、支付系统，任何一个子系统出了故障或者因为升级等原因暂时不可用，都会造成下单操作异常，影响用户使用体验。

使用消息队列解耦合，系统的耦合性就会提高了。比如物流系统发生故障，需要几分钟才能来修复，在这段时间内，物流系统要处理的数据被缓存到消息队列中，用户的下单操作正常完成。当物流系统回复后，补充处理存在消息队列中的订单消息即可，终端系统感知不到物流系统发生过几分钟故障。

流量削峰

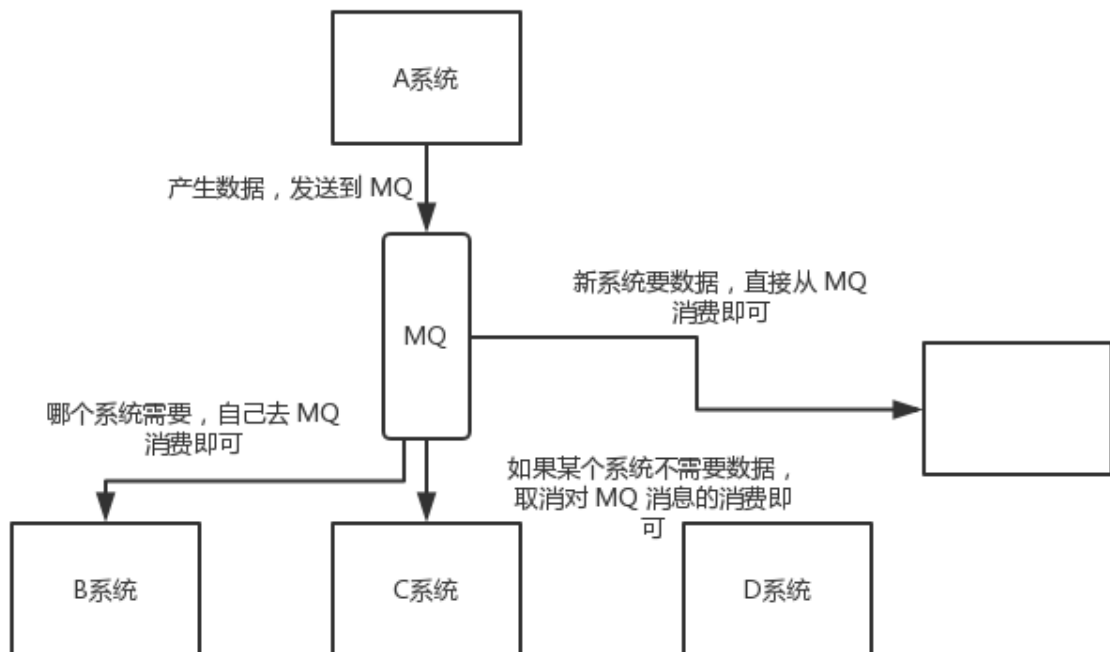
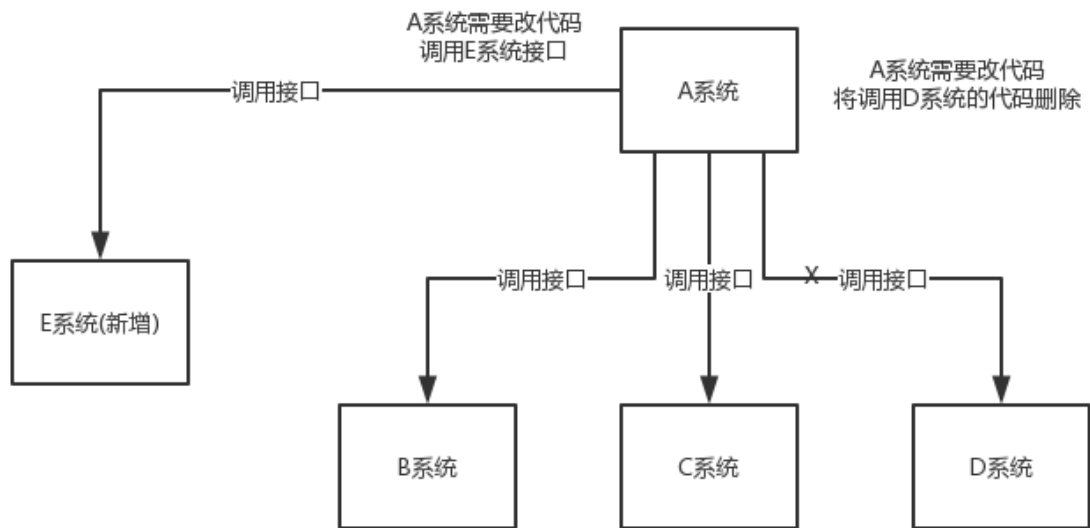
应用系统如果遇到系统请求流量的瞬间猛增，有可能会将系统压垮。有了消息队列可以将大量请求缓存起来，分散到很长一段时间处理，这样可以大大提到系统的稳定性和用户体验。

一般情况，为了保证系统的稳定性，如果系统负载超过阈值，就会阻止用户请求，这会影响用户体验，而如果使用消息队列将请求缓存起来，等待系统处理完毕后通知用户下单完毕，这样总不能下单体验要好。

业务系统正常时段的QPS如果是1000，流量最高峰是10000，为了应对流量高峰配置高性能的服务器显然不划算，这时可以使用消息队列对峰值流量削峰

数据分发

通过消息队列可以让数据在多个系统更加之间进行流通。数据的产生方不需要关心谁来使用数据，只需要将数据发送到消息队列，数据使用方直接在消息队列中直接获取数据即可



MQ的优点和缺点

优点：

- 应用解耦
- 流量削峰
- 数据分发

缺点:

- 系统可用性降低: 系统引入的外部依赖越多, 系统稳定性越差。一旦MQ宕机, 就会对业务造成影响
- 系统复杂度提高: MQ的加入大大增加了系统的复杂度, 以前系统间是同步的远程调用, 现在是通过MQ进行异步调用
- 一致性问题: A系统处理完业务, 通过MQ给B、C、D三个系统发消息数据, 如果B系统、C系统处理成功, D系统处理失败

RocketMQ快速入门

下载RocketMQ

[下载地址](#)

安装

解压版, 需要解压到某一个文件夹里

```
1 PS H:\opensoft> ls
2
3
4 目录: H:\opensoft
5
6
7 Mode                LastWriteTime         Length Name
8 ----                -
9 d-----            2022/5/5         18:34      apache-jmeter-5.4.3
10 d-----            2022/6/2          13:25      cerebro-0.9.4
11 d-----            2022/5/26         20:37      elasticsearch-analysis-ik-
12          8.2.0
13 d-----            2022/6/1          23:19      elasticsearch-cluster
14 d-----            2022/5/3          10:55      kibana-8.1.3
15 d-----            2022/5/30         23:02      logstash-8.1.3
16 d-----            2022/11/22        14:58      MongoDB
17 d-----            2022/7/20         12:25      mycat
18 d-----            2022/6/15         12:23      mycat-1.6
19 d-----            2022/7/19         15:35      nacos
20 d-----            2022/7/16         14:30      naocs-cluster
21 d-----            2022/7/16         21:38      nginx-1.21.6
22 d-----            2022/5/6          23:16      pvzpak-master.git
```

```

22 d----- 2022/11/29 21:21 rocketmq
23 d----- 2021/4/25 16:01 seata-server-1.4.2
24 d----- 2022/7/28 20:20 seata-server-cluster
25 d----- 2022/7/23 16:36 Sentinel
26 d----- 2022/8/11 20:14 浏览器主页

```

```

27
28
29 PS H:\opensoft> cd .\rocketmq\
30 PS H:\opensoft\rocketmq> ls

```

目录: H:\opensoft\rocketmq

Mode	LastWriteTime	Length	Name
d-----	2022/11/29 21:21		benchmark
d-----	2022/11/29 21:21		bin
d-----	2022/11/29 21:21		conf
d-----	2022/11/29 21:21		lib
-a----	2019/3/28 17:08	17336	LICENSE
-a----	2019/5/21 10:44	1337	NOTICE
-a----	2022/11/29 21:19	2523	README.md

```

45
46
47 PS H:\opensoft\rocketmq> cd .\bin\
48 PS H:\opensoft\rocketmq\bin> ls

```

目录: H:\opensoft\rocketmq\bin

Mode	LastWriteTime	Length	Name
d-----	2022/11/29 21:21		dledger
-a----	2019/3/28 17:08	1654	cachedog.sh
-a----	2019/3/28 17:08	845	cleancache.sh
-a----	2019/3/28 17:08	1116	cleancache.v1.sh
-a----	2019/3/28 17:08	1398	mqadmin
-a----	2019/3/28 17:08	1029	mqadmin.cmd
-a----	2019/3/28 17:08	1422	mqadmin.xml
-a----	2019/3/28 17:08	1394	mqbroker
-a----	2019/3/28 17:08	1084	mqbroker.cmd
-a----	2019/3/28 17:08	1373	mqbroker.numanode0
-a----	2019/3/28 17:08	1373	mqbroker.numanode1
-a----	2019/3/28 17:08	1373	mqbroker.numanode2
-a----	2019/3/28 17:08	1373	mqbroker.numanode3
-a----	2019/3/28 17:08	1388	mqbroker.xml
-a----	2019/3/28 17:08	1396	mqnamesrv
-a----	2019/3/28 17:08	1088	mqnamesrv.cmd
-a----	2019/3/28 17:08	1390	mqnamesrv.xml
-a----	2019/3/28 17:08	1571	mqshutdown
-a----	2019/3/28 17:08	1398	mqshutdown.cmd
-a----	2019/3/28 17:08	2222	os.sh
-a----	2019/3/28 17:08	1148	play.cmd
-a----	2019/3/28 17:08	1008	play.sh
-a----	2019/3/28 17:08	772	README.md
-a----	2019/3/28 17:08	2206	runbroker.cmd

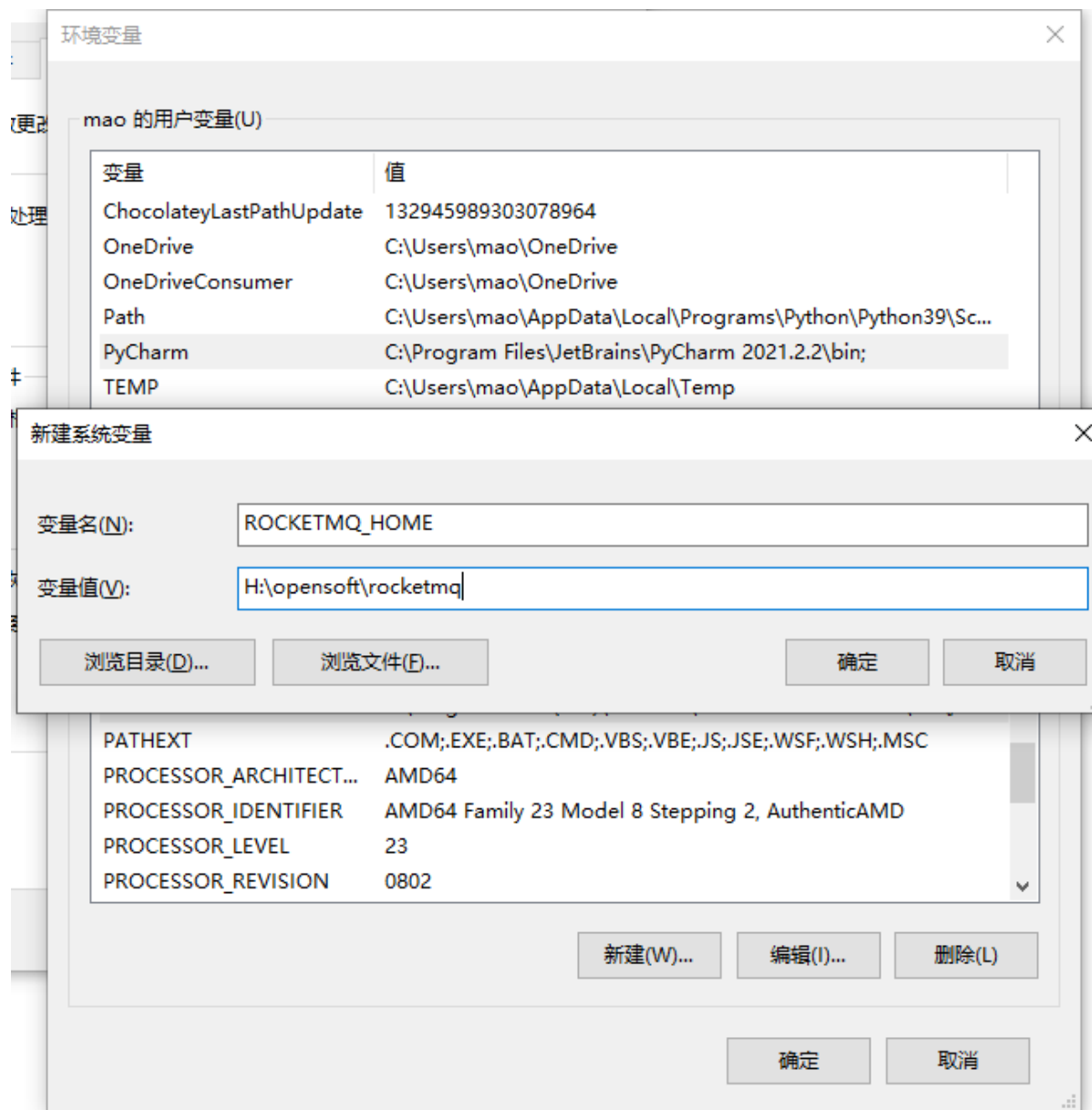
```
80 -a---- 2019/3/28 17:08 2849 runbroker.sh
81 -a---- 2019/3/28 17:08 1816 runserver.cmd
82 -a---- 2019/3/28 17:08 2443 runserver.sh
83 -a---- 2019/3/28 17:08 1156 setcache.sh
84 -a---- 2019/3/28 17:08 1408 startfsrv.sh
85 -a---- 2019/3/28 17:08 1601 tools.cmd
86 -a---- 2019/3/28 17:08 1870 tools.sh
87
88
89 PS H:\opensoft\rocketmq\bin>
```

目录介绍

- bin: 启动脚本, 包括shell脚本和CMD脚本
- conf: 实例配置文件, 包括broker配置文件、logback配置文件等
- lib: 依赖jar包, 包括Netty、commons-lang、FastJSON等

配置环境变量





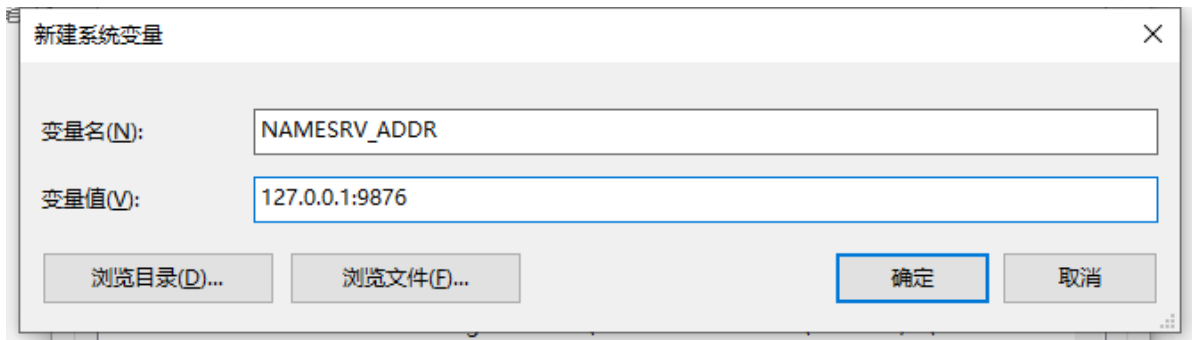
启动RocketMQ

启动NameServer

```
1 PS H:\opensoft\rocketmq\bin> .\mqnamesrv
2 OpenJDK 64-Bit Server VM warning: Using the DefNew young collector with the
  CMS collector is deprecated and will likely be removed in a future release
3 OpenJDK 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated
  and will likely be removed in a future release.
4 The Name Server boot success. serializeType=JSON
```

启动Broker

配置环境变量



```
1 PS H:\opensoft\rocketmq\bin> .\mqbroker
2 The broker[mao, 172.27.80.1:10911] boot success. serializeType=JSON and name
  server is 127.0.0.1:9876
```

测试RocketMQ

发送消息

```
1 | .\tools org.apache.rocketmq.example.quickstart.Producer
```

```
1 PS H:\opensoft\rocketmq\bin> .\tools
  org.apache.rocketmq.example.quickstart.Producer
2 OpenJDK 64-Bit Server VM warning: ignoring option PermSize=128m; support was
  removed in 8.0
3 OpenJDK 64-Bit Server VM warning: ignoring option MaxPermSize=128m; support
  was removed in 8.0
4 21:52:00.739 [main] DEBUG i.n.u.i.l.InternalLoggerFactory - Using SLF4J as
  the default logging framework
5 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E342180000,
  offsetMsgId=AC1B500100002A9F0000000000000000, messageQueue=MessageQueue
  [topic=TopicTest, brokerName=mao, queueId=3], queueOffset=0]
6 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E342380001,
  offsetMsgId=AC1B500100002A9F000000000000000B2, messageQueue=MessageQueue
  [topic=TopicTest, brokerName=mao, queueId=0], queueOffset=0]
```

```
7  SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3423B0002,
offsetMsgId=AC1B500100002A9F0000000000000164, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=0]
8  .....
9  .....
10 .....
11 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482103D6,
offsetMsgId=AC1B500100002A9F0000000000002B20A, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=245]
12 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482203D7,
offsetMsgId=AC1B500100002A9F0000000000002B2BE, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=2], queueOffset=245]
13 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482303D8,
offsetMsgId=AC1B500100002A9F0000000000002B372, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=3], queueOffset=246]
14 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482303D9,
offsetMsgId=AC1B500100002A9F0000000000002B426, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=0], queueOffset=246]
15 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482803DA,
offsetMsgId=AC1B500100002A9F0000000000002B4DA, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=246]
16 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482903DB,
offsetMsgId=AC1B500100002A9F0000000000002B58E, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=2], queueOffset=246]
17 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482A03DC,
offsetMsgId=AC1B500100002A9F0000000000002B642, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=3], queueOffset=247]
18 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482B03DD,
offsetMsgId=AC1B500100002A9F0000000000002B6F6, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=0], queueOffset=247]
19 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482C03DE,
offsetMsgId=AC1B500100002A9F0000000000002B7AA, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=247]
20 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482C03DF,
offsetMsgId=AC1B500100002A9F0000000000002B85E, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=2], queueOffset=247]
21 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482D03E0,
offsetMsgId=AC1B500100002A9F0000000000002B912, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=3], queueOffset=248]
22 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482E03E1,
offsetMsgId=AC1B500100002A9F0000000000002B9C6, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=0], queueOffset=248]
23 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3482F03E2,
offsetMsgId=AC1B500100002A9F0000000000002BA7A, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=248]
24 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3483003E3,
offsetMsgId=AC1B500100002A9F0000000000002BB2E, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=2], queueOffset=248]
25 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3483103E4,
offsetMsgId=AC1B500100002A9F0000000000002BBE2, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=3], queueOffset=249]
26 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3483203E5,
offsetMsgId=AC1B500100002A9F0000000000002BC96, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=0], queueOffset=249]
27 SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3483203E6,
offsetMsgId=AC1B500100002A9F0000000000002BD4A, messageQueue=MessageQueue
[topic=TopicTest, brokerName=mao, queueId=1], queueOffset=249]
```

```
28  SendResult [sendStatus=SEND_OK, msgId=76FBC10A4DDC60E53B9394E3483303E7,
    offsetMsgId=AC1B500100002A9F000000000002BDFE, messageQueue=MessageQueue
    [topic=TopicTest, brokerName=mao, queueId=2], queueOffset=249]
29  21:52:03.132 [NettyClientSelector_1] INFO  RocketmqRemoting - closeChannel:
    close the connection to remote address[127.0.0.1:9876] result: true
30  21:52:03.133 [NettyClientSelector_1] INFO  RocketmqRemoting - closeChannel:
    close the connection to remote address[172.27.80.1:10911] result: true
31  PS H:\opensoft\rocketmq\bin>
```

接收消息

```
1 | .\tools org.apache.rocketmq.example.quickstart.Consumer
```

```

1 .....
2 .....
3 .....
4 ConsumeMessageThread_13 Receive New Messages: [MessageExt [queueId=0,
storeSize=180, queueOffset=113, sysFlag=0, bornTimestamp=1669729922426,
bornHost=/172.27.80.1:61263, storeTimestamp=1669729922426,
storeHost=/172.27.80.1:10911, msgId=AC1B500100002A9F00000000000013E16,
commitLogOffset=81430, bodyCRC=1655802458, reconsumeTimes=0,
preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0,
properties={MIN_OFFSET=0, MAX_OFFSET=250, CONSUME_START_TIME=1669730050849,
UNIQ_KEY=76FBC10A4DDC60E53B9394E3457A01C5, WAIT=true, TAGS=TagA}, body=[72,
101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 52, 53, 51],
transactionId='null'}]]]
5 ConsumeMessageThread_10 Receive New Messages: [MessageExt [queueId=0,
storeSize=180, queueOffset=112, sysFlag=0, bornTimestamp=1669729922418,
bornHost=/172.27.80.1:61263, storeTimestamp=1669729922418,
storeHost=/172.27.80.1:10911, msgId=AC1B500100002A9F00000000000013B46,
commitLogOffset=80710, bodyCRC=461328901, reconsumeTimes=0,
preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0,
properties={MIN_OFFSET=0, MAX_OFFSET=250, CONSUME_START_TIME=1669730050849,
UNIQ_KEY=76FBC10A4DDC60E53B9394E3457201C1, WAIT=true, TAGS=TagA}, body=[72,
101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 52, 52, 57],
transactionId='null'}]]]
6 ConsumeMessageThread_5 Receive New Messages: [MessageExt [queueId=0,
storeSize=180, queueOffset=111, sysFlag=0, bornTimestamp=1669729922412,
bornHost=/172.27.80.1:61263, storeTimestamp=1669729922412,
storeHost=/172.27.80.1:10911, msgId=AC1B500100002A9F00000000000013876,
commitLogOffset=79990, bodyCRC=315170350, reconsumeTimes=0,
preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0,
properties={MIN_OFFSET=0, MAX_OFFSET=250, CONSUME_START_TIME=1669730050848,
UNIQ_KEY=76FBC10A4DDC60E53B9394E3456C01BD, WAIT=true, TAGS=TagA}, body=[72,
101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 52, 52, 53],
transactionId='null'}]]]
7 ConsumeMessageThread_2 Receive New Messages: [MessageExt [queueId=0,
storeSize=180, queueOffset=110, sysFlag=0, bornTimestamp=1669729922404,
bornHost=/172.27.80.1:61263, storeTimestamp=1669729922405,
storeHost=/172.27.80.1:10911, msgId=AC1B500100002A9F000000000000135A6,
commitLogOffset=79270, bodyCRC=363125303, reconsumeTimes=0,
preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0,
properties={MIN_OFFSET=0, MAX_OFFSET=250, CONSUME_START_TIME=1669730050848,
UNIQ_KEY=76FBC10A4DDC60E53B9394E3456401B9, WAIT=true, TAGS=TagA}, body=[72,
101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 52, 52, 49],
transactionId='null'}]]]
8 ConsumeMessageThread_9 Receive New Messages: [MessageExt [queueId=0,
storeSize=180, queueOffset=109, sysFlag=0, bornTimestamp=1669729922399,
bornHost=/172.27.80.1:61263, storeTimestamp=1669729922399,
storeHost=/172.27.80.1:10911, msgId=AC1B500100002A9F000000000000132D6,
commitLogOffset=78550, bodyCRC=864479685, reconsumeTimes=0,
preparedTransactionOffset=0, toString()=Message{topic='TopicTest', flag=0,
properties={MIN_OFFSET=0, MAX_OFFSET=250, CONSUME_START_TIME=1669730050848,
UNIQ_KEY=76FBC10A4DDC60E53B9394E3455F01B5, WAIT=true, TAGS=TagA}, body=[72,
101, 108, 108, 111, 32, 82, 111, 99, 107, 101, 116, 77, 81, 32, 52, 51, 55],
transactionId='null'}]]]
9 .....
10 .....
11 .....

```

关闭RocketMQ

```
1 | .\mqshutdown namesrv
```

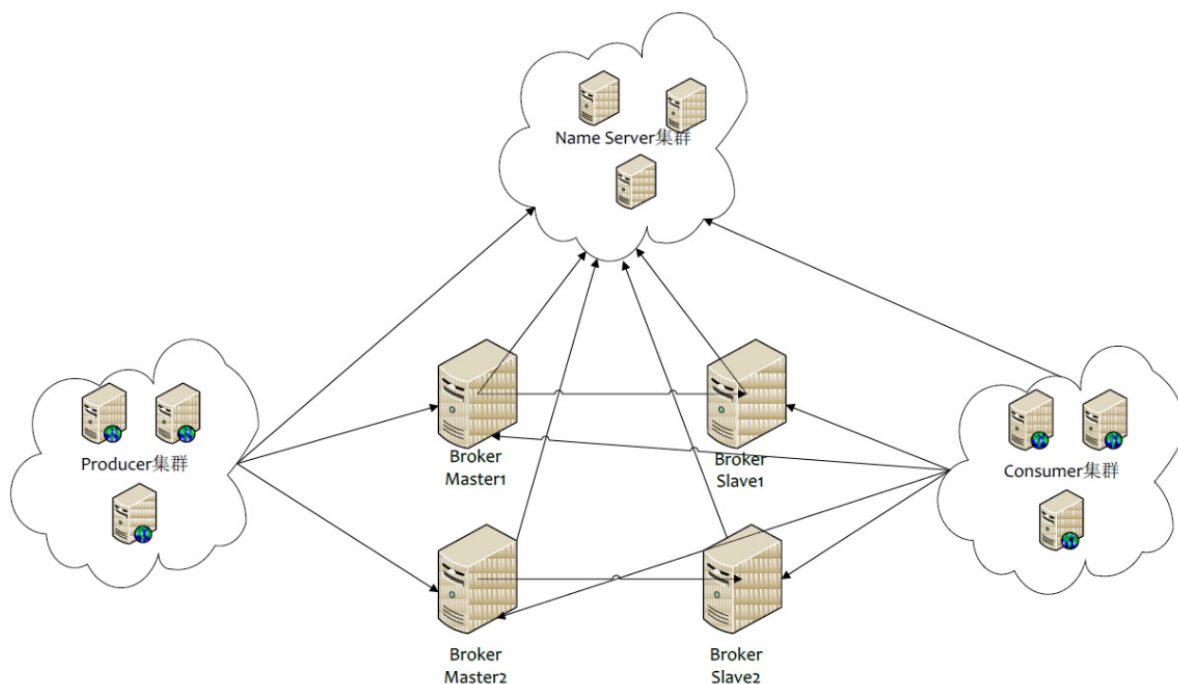
```
1 | .\mqshutdown broker
```

```
1 | PS H:\opensoft\rocketmq\bin> .\mqshutdown namesrv
2 | killing name server
3 | 成功: 已终止 PID 为 9444 的进程。
4 | Done!
5 | PS H:\opensoft\rocketmq\bin> .\mqshutdown broker
6 | killing broker
7 | 成功: 已终止 PID 为 21088 的进程。
8 | Done!
9 | PS H:\opensoft\rocketmq\bin>
```

RocketMQ集群搭建

各角色介绍

- Producer: 消息的发送者; 举例: 发信者
- Consumer: 消息接收者; 举例: 收信者
- Broker: 暂存和传输消息; 举例: 邮局
- NameServer: 管理Broker; 举例: 各个邮局的管理机构
- Topic: 区分消息的种类; 一个发送者可以发送消息给一个或者多个Topic; 一个消息的接收者可以订阅一个或者多个Topic消息
- Message Queue: 相当于是Topic的分区; 用于并行发送和接收消息



集群搭建方式

集群特点

- NameServer是一个几乎无状态节点，可集群部署，节点之间无任何信息同步。
- Broker部署相对复杂，Broker分为Master与Slave，一个Master可以对应多个Slave，但是一个Slave只能对应一个Master，Master与Slave的对应关系通过指定相同的BrokerName，不同的BrokerId来定义，BrokerId为0表示Master，非0表示Slave。Master也可以部署多个。每个Broker与NameServer集群中的所有节点建立长连接，定时注册Topic信息到所有NameServer。
- Producer与NameServer集群中的其中一个节点（随机选择）建立长连接，定期从NameServer取Topic路由信息，并向提供Topic服务的Master建立长连接，且定时向Master发送心跳。Producer完全无状态，可集群部署。
- Consumer与NameServer集群中的其中一个节点（随机选择）建立长连接，定期从NameServer取Topic路由信息，并向提供Topic服务的Master、Slave建立长连接，且定时向Master、Slave发送心跳。Consumer既可以从Master订阅消息，也可以从Slave订阅消息，订阅规则由Broker配置决定。

集群模式

单Master模式

这种方式风险较大，一旦Broker重启或者宕机时，会导致整个服务不可用。不建议线上环境使用，可以用于本地测试

多Master模式

一个集群无Slave，全是Master，例如2个Master或者3个Master，这种模式的优缺点如下：

- 优点：配置简单，单个Master宕机或重启维护对应用无影响，在磁盘配置为RAID10时，即使机器宕机不可恢复情况下，由于RAID10磁盘非常可靠，消息也不会丢（异步刷盘丢失少量消息，同步刷盘一条不丢），性能最高；
- 缺点：单台机器宕机期间，这台机器上未被消费的消息在机器恢复之前不可订阅，消息实时性会受到影响。

多Master多Slave模式（异步）

每个Master配置一个Slave，有多对Master-Slave，HA采用异步复制方式，主备有短暂消息延迟（毫秒级），这种模式的优缺点如下：

- 优点：即使磁盘损坏，消息丢失的非常少，且消息实时性不会受影响，同时Master宕机后，消费者仍然可以从Slave消费，而且此过程对应用透明，不需要人工干预，性能同多Master模式几乎一样；
- 缺点：Master宕机，磁盘损坏情况下会丢失少量消息。

多Master多Slave模式（同步）

每个Master配置一个Slave，有多对Master-Slave，HA采用同步双写方式，即只有主备都写成功，才向应用返回成功，这种模式的优缺点如下：

- 优点：数据与服务都无单点故障，Master宕机情况下，消息无延迟，服务可用性与数据可用性都非常高；
- 缺点：性能比异步复制模式略低（大约低10%左右），发送单个消息的RT会略高，且目前版本在主节点宕机后，备机不能自动切换为主机。

双主双从集群搭建

集群工作流程

1. 启动NameServer，NameServer起来后监听端口，等待Broker、Producer、Consumer连上来，相当于一个路由控制中心。
2. Broker启动，跟所有的NameServer保持长连接，定时发送心跳包。心跳包中包含当前Broker信息(IP+端口等)以及存储所有Topic信息。注册成功后，NameServer集群中就有Topic跟Broker的映射

关系。

3. 收发消息前，先创建Topic，创建Topic时需要指定该Topic要存储在哪些Broker上，也可以在发送消息时自动创建Topic。
4. Producer发送消息，启动时先跟NameServer集群中的其中一台建立长连接，并从NameServer中获取当前发送的Topic存在哪些Broker上，轮询从队列列表中选择一个队列，然后与队列所在的Broker建立长连接从而向Broker发消息。
5. Consumer跟Producer类似，跟其中一台NameServer建立长连接，获取当前订阅Topic存在哪些Broker上，然后直接跟Broker建立连接通道，开始消费消息。

创建消息存储路径

```
1 mkdir ./data
2 mkdir ./data/master1/commitlog
3 mkdir ./data/master1/consumequeue
4 mkdir ./data/master1/index
5 mkdir ./data/master2/commitlog
6 mkdir ./data/master2/consumequeue
7 mkdir ./data/master2/index
8 mkdir ./data/slave1/commitlog
9 mkdir ./data/slave1/consumequeue
10 mkdir ./data/slave1/index
11 mkdir ./data/slave2/commitlog
12 mkdir ./data/slave2/consumequeue
13 mkdir ./data/slave2/index
```

```
1 PS H:\opensoft\rocketmq> mkdir ./data
2
3
4 目录: H:\opensoft\rocketmq
5
6
7 Mode                LastWriteTime         Length Name
8 ----                -
9 d-----            2022/12/1    20:42      data
10
11
12 PS H:\opensoft\rocketmq> mkdir ./data/master1/commitlog
13
14
15 目录: H:\opensoft\rocketmq\data\master1
16
17
18 Mode                LastWriteTime         Length Name
19 ----                -
20 d-----            2022/12/1    20:42    commitlog
21
22
```

```

23 PS H:\opensoft\rocketmq> mkdir ./data/master1/consumequeue
24
25
26 目录: H:\opensoft\rocketmq\data\master1
27
28
29 Mode                LastWriteTime         Length Name
30 ----                -
31 d-----            2022/12/1     20:42             consumequeue
32
33
34 PS H:\opensoft\rocketmq> mkdir ./data/master1/index
35
36
37 目录: H:\opensoft\rocketmq\data\master1
38
39
40 Mode                LastWriteTime         Length Name
41 ----                -
42 d-----            2022/12/1     20:42             index
43
44
45 PS H:\opensoft\rocketmq> mkdir ./data/master2/commitlog
46
47
48 目录: H:\opensoft\rocketmq\data\master2
49
50
51 Mode                LastWriteTime         Length Name
52 ----                -
53 d-----            2022/12/1     20:42             commitlog
54
55
56 PS H:\opensoft\rocketmq> mkdir ./data/master2/consumequeue
57
58
59 目录: H:\opensoft\rocketmq\data\master2
60
61
62 Mode                LastWriteTime         Length Name
63 ----                -
64 d-----            2022/12/1     20:42             consumequeue
65
66
67 PS H:\opensoft\rocketmq> mkdir ./data/master2/index
68
69
70 目录: H:\opensoft\rocketmq\data\master2
71
72
73 Mode                LastWriteTime         Length Name
74 ----                -
75 d-----            2022/12/1     20:42             index
76
77
78 PS H:\opensoft\rocketmq> mkdir ./data/slave1/commitlog
79
80

```

```

81  目录: H:\opensoft\rocketmq\data\slave1
82
83
84  Mode                LastWriteTime         Length Name
85  ----                -
86  d-----            2022/12/1     20:42             commitlog
87
88
89  PS H:\opensoft\rocketmq> mkdir ./data/slave1/consumequeue
90
91
92  目录: H:\opensoft\rocketmq\data\slave1
93
94
95  Mode                LastWriteTime         Length Name
96  ----                -
97  d-----            2022/12/1     20:42             consumequeue
98
99
100 PS H:\opensoft\rocketmq> mkdir ./data/slave1/index
101
102
103  目录: H:\opensoft\rocketmq\data\slave1
104
105
106  Mode                LastWriteTime         Length Name
107  ----                -
108  d-----            2022/12/1     20:42             index
109
110
111 PS H:\opensoft\rocketmq> mkdir ./data/slave2/commitlog
112
113
114  目录: H:\opensoft\rocketmq\data\slave2
115
116
117  Mode                LastWriteTime         Length Name
118  ----                -
119  d-----            2022/12/1     20:42             commitlog
120
121
122 PS H:\opensoft\rocketmq> mkdir ./data/slave2/consumequeue
123
124
125  目录: H:\opensoft\rocketmq\data\slave2
126
127
128  Mode                LastWriteTime         Length Name
129  ----                -
130  d-----            2022/12/1     20:42             consumequeue
131
132
133 PS H:\opensoft\rocketmq> mkdir ./data/slave2/index
134
135
136  目录: H:\opensoft\rocketmq\data\slave2
137
138

```

```

139 Mode                LastWriteTime         Length Name
140 ----                -
141 d-----             2022/12/1      20:42             index
142
143
144 PS H:\opensoft\rocketmq>

```

```

1  PS H:\opensoft\rocketmq> ls
2
3
4      目录: H:\opensoft\rocketmq
5
6
7  Mode                LastWriteTime         Length Name
8  ----                -
9  d-----             2022/11/29      21:21      benchmark
10 d-----             2022/11/29      21:21         bin
11 d-----             2022/11/29      21:21        conf
12 d-----             2022/12/1      20:42        data
13 d-----             2022/11/29      21:21         lib
14 -a-----             2019/3/28       17:08     17336 LICENSE
15 -a-----             2019/5/21       10:44     1337 NOTICE
16 -a-----             2022/11/29      21:19     2523 README.md
17
18
19 PS H:\opensoft\rocketmq> cd .\data\
20 PS H:\opensoft\rocketmq\data> ls
21
22
23      目录: H:\opensoft\rocketmq\data
24
25
26 Mode                LastWriteTime         Length Name
27 ----                -
28 d-----             2022/12/1      20:42      master1
29 d-----             2022/12/1      20:42      master2
30 d-----             2022/12/1      20:42      slave1
31 d-----             2022/12/1      20:42      slave2
32
33
34 PS H:\opensoft\rocketmq\data> cd .\master1\
35 PS H:\opensoft\rocketmq\data\master1> ls
36
37
38      目录: H:\opensoft\rocketmq\data\master1
39
40
41 Mode                LastWriteTime         Length Name
42 ----                -
43 d-----             2022/12/1      20:42      commitlog
44 d-----             2022/12/1      20:42      consumequeue
45 d-----             2022/12/1      20:42      index
46
47
48 PS H:\opensoft\rocketmq\data\master1>

```

broker配置文件

配置文件使用的是软件目录下\conf\2m-2s-sync的配置文件

```
1 PS H:\opensoft\rocketmq> ls
2
3
4 目录: H:\opensoft\rocketmq
5
6
7 Mode                LastWriteTime         Length Name
8 ----                -
9 d-----            2022/11/29    21:21      benchmark
10 d-----            2022/11/29    21:21      bin
11 d-----            2022/11/29    21:21      conf
12 d-----            2022/11/30    18:37      data
13 d-----            2022/11/29    21:21      lib
14 -a-----            2019/3/28    17:08    17336 LICENSE
15 -a-----            2019/5/21    10:44    1337 NOTICE
16 -a-----            2022/11/29    21:19    2523 README.md
17
18
19 PS H:\opensoft\rocketmq> cd conf
20 PS H:\opensoft\rocketmq\conf> ls
21
22
23 目录: H:\opensoft\rocketmq\conf
24
25
26 Mode                LastWriteTime         Length Name
27 ----                -
28 d-----            2022/11/29    21:21      2m-2s-async
29 d-----            2022/11/29    21:21      2m-2s-sync
30 d-----            2022/11/29    21:21      2m-noslave
31 d-----            2022/11/29    21:21      dledger
32 -a-----            2019/3/28    17:08     949 broker.conf
33 -a-----            2019/3/28    17:08   14978 logback_broker.xml
34 -a-----            2019/3/28    17:08   3836 logback_namesrv.xml
35 -a-----            2019/3/28    17:08   3761 logback_tools.xml
36 -a-----            2019/5/21    10:44    1305 plain_acl.yml
37 -a-----            2019/5/21    10:44     834 tools.yml
38
39
40 PS H:\opensoft\rocketmq\conf> cd .\2m-2s-sync\
41 PS H:\opensoft\rocketmq\conf\2m-2s-sync> ls
42
43
44 目录: H:\opensoft\rocketmq\conf\2m-2s-sync
45
46
47 Mode                LastWriteTime         Length Name
48 ----                -
```

```

49 -a----      2019/3/28    17:08          922 broker-a-s.properties
50 -a----      2019/3/28    17:08          928 broker-a.properties
51 -a----      2019/3/28    17:08          922 broker-b-s.properties
52 -a----      2019/3/28    17:08          928 broker-b.properties
53
54
55 PS H:\opensoft\rocketmq\conf\2m-2s-sync> cat .\broker-a.properties
56 # Licensed to the Apache Software Foundation (ASF) under one or more
57 # contributor license agreements.  See the NOTICE file distributed with
58 # this work for additional information regarding copyright ownership.
59 # The ASF licenses this file to You under the Apache License, Version 2.0
60 # (the "License"); you may not use this file except in compliance with
61 # the License.  You may obtain a copy of the License at
62 #
63 #     http://www.apache.org/licenses/LICENSE-2.0
64 #
65 # Unless required by applicable law or agreed to in writing, software
66 # distributed under the License is distributed on an "AS IS" BASIS,
67 # WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
68 # See the License for the specific language governing permissions and
69 # limitations under the License.
70 brokerClusterName=DefaultCluster
71 brokerName=broker-a
72 brokerId=0
73 deleteWhen=04
74 fileReservedTime=48
75 brokerRole=SYNC_MASTER
76 flushDiskType=ASYNC_FLUSH
77 PS H:\opensoft\rocketmq\conf\2m-2s-sync>

```

master1

配置文件名称: broker-a.properties

修改配置如下:

```

1  #所属集群名字
2  brokerClusterName=rocketmq-cluster
3  #broker名字, 注意此处不同的配置文件填写的不一样
4  brokerName=broker-a
5  #0 表示 Master, >0 表示 Slave
6  brokerId=0
7  #nameServer地址, 分号分割
8  namesrvAddr=127.0.0.1:9876
9  #在发送消息时, 自动创建服务器不存在的topic, 默认创建的队列数
10 defaultTopicQueueNums=4
11 #是否允许 Broker 自动创建Topic, 建议线下开启, 线上关闭
12 autoCreateTopicEnable=true
13 #是否允许 Broker 自动创建订阅组, 建议线下开启, 线上关闭
14 autoCreateSubscriptionGroup=true
15 #Broker 对外服务的监听端口
16 listenPort=10911
17 #删除文件时间点, 默认凌晨 4点
18 deleteWhen=04

```

```

19 #文件保留时间，默认 48 小时
20 fileReservedTime=120
21 #commitLog每个文件的大小默认1G
22 mappedFileSizeCommitLog=1073741824
23 #ConsumeQueue每个文件默认存30w条，根据业务情况调整
24 mappedFileSizeConsumeQueue=300000
25 #destroyMappedFileIntervalForcibly=120000
26 #redeleteHangedFileInterval=120000
27 #检测物理文件磁盘空间
28 diskMaxUsedSpaceRatio=88
29 #存储路径
30 storePathRootDir=../../data/master1
31 #commitLog 存储路径
32 storePathCommitLog=../../data/master1/commitlog
33 #消费队列存储路径存储路径
34 storePathConsumeQueue=../../data/master1/consumequeue
35 #消息索引存储路径
36 storePathIndex=../../data/master1/index
37 #checkpoint 文件存储路径
38 storeCheckpoint=../../data/master1/checkpoint
39 #abort 文件存储路径
40 abortFile=../../data/master1/abort
41 #限制的消息大小
42 maxMessageSize=65536
43 #flushCommitLogLeastPages=4
44 #flushConsumeQueueLeastPages=2
45 #flushCommitLogThoroughInterval=10000
46 #flushConsumeQueueThoroughInterval=60000
47 #Broker 的角色
48 #- ASYNC_MASTER 异步复制Master
49 #- SYNC_MASTER 同步双写Master
50 #- SLAVE
51 brokerRole=SYNC_MASTER
52 #刷盘方式
53 #- ASYNC_FLUSH 异步刷盘
54 #- SYNC_FLUSH 同步刷盘
55 flushDiskType=SYNC_FLUSH
56 #checkTransactionMessageEnable=false
57 #发消息线程池数量
58 #sendMessageThreadPoolNums=128
59 #拉消息线程池数量
60 #pullMessageThreadPoolNums=128

```

slave2

配置文件名称：broker-b-s.properties

```

1 #所属集群名字
2 brokerClusterName=rocketmq-cluster
3 #broker名字，注意此处不同的配置文件填写的不一样
4 brokerName=broker-b
5 #0 表示 Master，>0 表示 slave
6 brokerId=1

```



```

7  #nameServer地址，分号分割
8  namesrvAddr=127.0.0.1:9876
9  #在发送消息时，自动创建服务器不存在的topic，默认创建的队列数
10 defaultTopicQueueNums=4
11 #是否允许 Broker 自动创建Topic，建议线下开启，线上关闭
12 autoCreateTopicEnable=true
13 #是否允许 Broker 自动创建订阅组，建议线下开启，线上关闭
14 autoCreateSubscriptionGroup=true
15 #Broker 对外服务的监听端口
16 listenPort=10941
17 #删除文件时间点，默认凌晨 4点
18 deleteWhen=04
19 #文件保留时间，默认 48 小时
20 fileReservedTime=120
21 #commitLog每个文件的大小默认1G
22 mappedFileSizeCommitLog=1073741824
23 #ConsumeQueue每个文件默认存30w条，根据业务情况调整
24 mappedFileSizeConsumeQueue=300000
25 #destroyMappedFileIntervalForcibly=120000
26 #redeleteHangedFileInterval=120000
27 #检测物理文件磁盘空间
28 diskMaxUsedSpaceRatio=88
29 #存储路径
30 storePathRootDir=../../data/slave2
31 #commitLog 存储路径
32 storePathCommitLog=../../data/slave2/commitlog
33 #消费队列存储路径存储路径
34 storePathConsumeQueue=../../data/slave2/consumequeue
35 #消息索引存储路径
36 storePathIndex=../../data/slave2/index
37 #checkpoint 文件存储路径
38 storeCheckpoint=../../data/slave2/checkpoint
39 #abort 文件存储路径
40 abortFile=../../data/slave2/abort
41 #限制的消息大小
42 maxMessageSize=65536
43 #flushCommitLogLeastPages=4
44 #flushConsumeQueueLeastPages=2
45 #flushCommitLogThoroughInterval=10000
46 #flushConsumeQueueThoroughInterval=60000
47 #Broker 的角色
48 #- ASYNC_MASTER 异步复制Master
49 #- SYNC_MASTER 同步双写Master
50 #- SLAVE
51 brokerRole=SLAVE
52 #刷盘方式
53 #- ASYNC_FLUSH 异步刷盘
54 #- SYNC_FLUSH 同步刷盘
55 flushDiskType=ASYNC_FLUSH
56 #checkTransactionMessageEnable=false
57 #发消息线程池数量
58 #sendMessageThreadPoolNums=128
59 #拉消息线程池数量
60 #pullMessageThreadPoolNums=128

```

master2

配置文件名称: broker-b.properties

```
1  #所属集群名字
2  brokerClusterName=rocketmq-cluster
3  #broker名字, 注意此处不同的配置文件填写的不一样
4  brokerName=broker-b
5  #0 表示 Master, >0 表示 Slave
6  brokerId=0
7  #nameServer地址, 分号分割
8  namesrvAddr=127.0.0.1:9876
9  #在发送消息时, 自动创建服务器不存在的topic, 默认创建的队列数
10 defaultTopicQueueNums=4
11 #是否允许 Broker 自动创建Topic, 建议线下开启, 线上关闭
12 autoCreateTopicEnable=true
13 #是否允许 Broker 自动创建订阅组, 建议线下开启, 线上关闭
14 autoCreateSubscriptionGroup=true
15 #Broker 对外服务的监听端口
16 listenPort=10921
17 #删除文件时间点, 默认凌晨 4点
18 deleteWhen=04
19 #文件保留时间, 默认 48 小时
20 fileReservedTime=120
21 #commitLog每个文件的大小默认1G
22 mappedFileSizeCommitLog=1073741824
23 #ConsumeQueue每个文件默认存30w条, 根据业务情况调整
24 mappedFileSizeConsumeQueue=300000
25 #destroyMappedFileIntervalForcibly=120000
26 #redeleteHangedFileInterval=120000
27 #检测物理文件磁盘空间
28 diskMaxUsedSpaceRatio=88
29 #存储路径
30 storePathRootDir=../../data/master2
31 #commitLog 存储路径
32 storePathCommitLog=../../data/master2/commitlog
33 #消费队列存储路径存储路径
34 storePathConsumeQueue=../../data/master2/consumequeue
35 #消息索引存储路径
36 storePathIndex=../../data/master2/index
37 #checkpoint 文件存储路径
38 storeCheckpoint=../../data/master2/checkpoint
39 #abort 文件存储路径
40 abortFile=../../data/master2/abort
41 #限制的消息大小
42 maxMessageSize=65536
43 #flushCommitLogLeastPages=4
44 #flushConsumeQueueLeastPages=2
45 #flushCommitLogThoroughInterval=10000
46 #flushConsumeQueueThoroughInterval=60000
47 #Broker 的角色
48 #- ASYNC_MASTER 异步复制Master
49 #- SYNC_MASTER 同步双写Master
50 #- SLAVE
51 brokerRole=SYNC_MASTER
52 #刷盘方式
53 #- ASYNC_FLUSH 异步刷盘
```

```

54 #- SYNC_FLUSH 同步刷盘
55 flushDiskType=SYNC_FLUSH
56 #checkTransactionMessageEnable=false
57 #发消息线程池数量
58 #sendMessageThreadPoolNums=128
59 #拉消息线程池数量
60 #pullMessageThreadPoolNums=128

```

slave1

配置文件名称: broker-a-s.properties

```

1  #所属集群名字
2  brokerClusterName=rocketmq-cluster
3  #broker名字, 注意此处不同的配置文件填写的不一样
4  brokerName=broker-a
5  #0 表示 Master, >0 表示 Slave
6  brokerId=1
7  #nameServer地址, 分号分割
8  namesrvAddr=127.0.0.1:9876
9  #在发送消息时, 自动创建服务器不存在的topic, 默认创建的队列数
10 defaultTopicQueueNums=4
11 #是否允许 Broker 自动创建Topic, 建议线下开启, 线上关闭
12 autoCreateTopicEnable=true
13 #是否允许 Broker 自动创建订阅组, 建议线下开启, 线上关闭
14 autoCreateSubscriptionGroup=true
15 #Broker 对外服务的监听端口
16 listenPort=10931
17 #删除文件时间点, 默认凌晨 4点
18 deleteWhen=04
19 #文件保留时间, 默认 48 小时
20 fileReservedTime=120
21 #commitLog每个文件的大小默认1G
22 mappedFileSizeCommitLog=1073741824
23 #ConsumeQueue每个文件默认存30w条, 根据业务情况调整
24 mappedFileSizeConsumeQueue=300000
25 #destroyMappedFileIntervalForcibly=120000
26 #redeleteHangedFileInterval=120000
27 #检测物理文件磁盘空间
28 diskMaxUsedSpaceRatio=88
29 #存储路径
30 storePathRootDir=../../data/slave1
31 #commitLog 存储路径
32 storePathCommitLog=../../data/slave1/commitlog
33 #消费队列存储路径存储路径
34 storePathConsumeQueue=../../data/slave1/consumequeue
35 #消息索引存储路径
36 storePathIndex=../../data/slave1/index
37 #checkpoint 文件存储路径
38 storeCheckpoint=../../data/slave1/checkpoint
39 #abort 文件存储路径
40 abortFile=../../data/slave1/abort
41 #限制的消息大小

```

```

42 maxMessageSize=65536
43 #flushCommitLogLeastPages=4
44 #flushConsumeQueueLeastPages=2
45 #flushCommitLogThoroughInterval=10000
46 #flushConsumeQueueThoroughInterval=60000
47 #Broker 的角色
48 #- ASYNC_MASTER 异步复制Master
49 #- SYNC_MASTER 同步双写Master
50 #- SLAVE
51 brokerRole=SLAVE
52 #刷盘方式
53 #- ASYNC_FLUSH 异步刷盘
54 #- SYNC_FLUSH 同步刷盘
55 flushDiskType=ASYNC_FLUSH
56 #checkTransactionMessageEnable=false
57 #发消息线程池数量
58 #sendMessageThreadPoolNums=128
59 #拉消息线程池数量
60 #pullMessageThreadPoolNums=128

```

服务启动

启动NameServer

```
1 | .\mqnamesrv
```

```

1 | PS H:\opensoft\rocketmq\bin> .\mqnamesrv
2 | OpenJDK 64-Bit Server VM warning: Using the DefNew young collector with the
   | CMS collector is deprecated and will likely be removed in a future release
3 | OpenJDK 64-Bit Server VM warning: UseCMSCompactAtFullCollection is deprecated
   | and will likely be removed in a future release.
4 | The Name Server boot success. serializeType=JSON

```

启动Broker集群

master1:

```
1 | .\mqbroker -c ../../conf/2m-2s-sync/broker-a.properties
```

slave1:

```
1 | .\mqbroker -c ../../conf/2m-2s-sync/broker-a-s.properties
```

master2:

```
1 | .\mqbroker -c ../../conf/2m-2s-sync/broker-b.properties
```

slave2:

```
1 | .\mqbroker -c ../../conf/2m-2s-sync/broker-b-s.properties
```

```
1 | PS H:\opensoft\rocketmq\bin> .\mqbroker -c ../../conf/2m-2s-sync/broker-  
a.properties  
2 | The broker[broker-a, 172.27.80.1:10911] boot success. serializeType=JSON and  
name server is 127.0.0.1:9876
```

```
1 | PS H:\opensoft\rocketmq\bin> .\mqbroker -c ../../conf/2m-2s-sync/broker-a-  
s.properties  
2 | The broker[broker-a, 172.27.80.1:10931] boot success. serializeType=JSON and  
name server is 127.0.0.1:9876
```

```
1 | PS H:\opensoft\rocketmq\bin> .\mqbroker -c ../../conf/2m-2s-sync/broker-  
b.properties  
2 | The broker[broker-b, 172.27.80.1:10921] boot success. serializeType=JSON and  
name server is 127.0.0.1:9876
```

```
1 | PS H:\opensoft\rocketmq\bin> .\mqbroker -c ../../conf/2m-2s-sync/broker-b-  
s.properties  
2 | The broker[broker-b, 172.27.80.1:10941] boot success. serializeType=JSON and  
name server is 127.0.0.1:9876
```

启动脚本

多窗口模式

```

1 cd bin
2 start "RocketMQ-nameServer-9876" mqnamesrv
3 start "RocketMQ-broker-master1-10911" mqbroker -c ../../conf/2m-2s-
sync/broker-a.properties
4 start "RocketMQ-broker-slave1-10931" mqbroker -c ../../conf/2m-2s-sync/broker-
a-s.properties
5 start "RocketMQ-broker-master2-10921" mqbroker -c ../../conf/2m-2s-
sync/broker-b.properties
6 start "RocketMQ-broker-slave2-10941" mqbroker -c ../../conf/2m-2s-sync/broker-
b-s.properties

```

单窗口模式

```

1 cd bin
2 start /b "RocketMQ-nameServer-9876" mqnamesrv
3 start /b "RocketMQ-broker-master1-10911" mqbroker -c ../../conf/2m-2s-
sync/broker-a.properties
4 start /b "RocketMQ-broker-slave1-10931" mqbroker -c ../../conf/2m-2s-
sync/broker-a-s.properties
5 start /b "RocketMQ-broker-master2-10921" mqbroker -c ../../conf/2m-2s-
sync/broker-b.properties
6 start /b "RocketMQ-broker-slave2-10941" mqbroker -c ../../conf/2m-2s-
sync/broker-b-s.properties

```

```

1 PS H:\opensoft\rocketmq> ls
2
3
4 目录: H:\opensoft\rocketmq
5
6
7 Mode                LastWriteTime         Length Name
8 ----                -
9 d-----            2022/11/29    21:21      benchmark
10 d-----            2022/11/29    21:21      bin
11 d-----            2022/11/29    21:21      conf
12 d-----            2022/12/1     20:42      data
13 d-----            2022/11/29    21:21      lib
14 -a-----            2019/3/28     17:08    17336 LICENSE
15 -a-----            2019/5/21     10:44     1337 NOTICE
16 -a-----            2022/11/29    21:19     2523 README.md
17 -a-----            2022/12/2     10:57     424 集群启动-2m-2s-sync.bat
18 -a-----            2022/12/2     10:59     439 集群启动-单窗口-2m-2s-
sync.bat
19
20
21 PS H:\opensoft\rocketmq> cat .\集群启动-2m-2s-sync.bat
22 cd bin
23 start "RocketMQ-nameServer-9876" mqnamesrv
24 start "RocketMQ-broker-master1-10911" mqbroker -c ../../conf/2m-2s-
sync/broker-a.properties

```

```

25 start "RocketMQ-broker-slave1-10931" mqbroker -c ../../conf/2m-2s-
    sync/broker-a-s.properties
26 start "RocketMQ-broker-master2-10921" mqbroker -c ../../conf/2m-2s-
    sync/broker-b.properties
27 start "RocketMQ-broker-slave2-10941" mqbroker -c ../../conf/2m-2s-
    sync/broker-b-s.properties
28
29 PS H:\opensoft\rocketmq> cat .\集群启动-单窗口-2m-2s-sync.bat
30 cd bin
31 start /b "RocketMQ-nameServer-9876" mqnamesrv
32 start /b "RocketMQ-broker-master1-10911" mqbroker -c ../../conf/2m-2s-
    sync/broker-a.properties
33 start /b "RocketMQ-broker-slave1-10931" mqbroker -c ../../conf/2m-2s-
    sync/broker-a-s.properties
34 start /b "RocketMQ-broker-master2-10921" mqbroker -c ../../conf/2m-2s-
    sync/broker-b.properties
35 start /b "RocketMQ-broker-slave2-10941" mqbroker -c ../../conf/2m-2s-
    sync/broker-b-s.properties
36
37 PS H:\opensoft\rocketmq>

```

查看进程状态

```

1 PS H:\opensoft\rocketmq> jps
2 10848 NamesrvStartup
3 11856 BrokerStartup
4 23268 BrokerStartup
5 14568 BrokerStartup
6 11788 Jps
7 14300 BrokerStartup
8 PS H:\opensoft\rocketmq>

```

mqadmin管理工具

集群监控平台搭建

克隆开源项目

项目地址: <https://github.com/apache/rocketmq-externals>

```
1 | git clone https://github.com/apache/rocketmq-externals
```

```
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
remote: Enumerating objects: 22470, done.
remote: Counting objects: 100% (2725/2725), done.
remote: Compressing objects: 100% (1255/1255), done.
Receiving objects: 22% (5109/22470), 15.18 MiB | 39.00 KiB/s
```

```
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
remote: Enumerating objects: 22470, done.
remote: Counting objects: 100% (2725/2725), done.
remote: Compressing objects: 100% (1255/1255), done.
fatal: fetch-pack: invalid index-pack outputMiB | 24.00 KiB/s
fetch-pack: unexpected disconnect while reading sideband packet
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
fatal: unable to access 'https://github.com/apache/rocketmq-externals/': Failed to connect to github.com port 443: Timed out
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
fatal: unable to access 'https://github.com/apache/rocketmq-externals/': Failed to connect to github.com port 443: Timed out
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
fatal: unable to access 'https://github.com/apache/rocketmq-externals/': Failed to connect to github.com port 443: Timed out
PS C:\Users\mao\Desktop> git clone https://github.com/apache/rocketmq-externals
Cloning into 'rocketmq-externals'...
remote: Enumerating objects: 22470, done.
remote: Counting objects: 100% (2725/2725), done.
remote: Compressing objects: 100% (1255/1255), done.
remote: Total 22470 (delta 1042), reused 2393 (delta 871), pack-reused 19745
Receiving objects: 100% (22470/22470), 35.51 MiB | 40.00 KiB/s, done.
Resolving deltas: 100% (8816/8816), done.
Updating files: 100% (4817/4817), done.
PS C:\Users\mao\Desktop>
```

如果没有rocketmq-console

可以使用以下地址下载

<https://gitcode.net/mirrors/apache/rocketmq-externals/-/tree/rocketmq-console-1.0.0>

配置集群地址

打包前在 `rocketmq-console` 中配置 `namesrv` 集群地址

```
1 | rocketmq.config.namesrvAddr=127.0.0.1:9876
```

```
1 | server.contextPath=
2 | server.port=8080
3 | #spring.application.index=true
4 | spring.application.name=rocketmq-console
5 | spring.http.encoding.charset=UTF-8
6 | spring.http.encoding.enabled=true
7 | spring.http.encoding.force=true
8 | logging.config=classpath:logback.xml
9 | #if this value is empty,use env value rocketmq.config.namesrvAddr
   NAMESRV_ADDR | now, you can set it in ops page.default localhost:9876
10 | rocketmq.config.namesrvAddr=127.0.0.1:9876
11 | #if you use rocketmq version < 3.5.8, rocketmq.config.isVIPChannel should be
   false.default true
12 | rocketmq.config.isVIPChannel=
13 | #rocketmq-console's data path:dashboard/monitor
14 | rocketmq.config.dataPath=/tmp/rocketmq-console/data
15 | #set it false if you don't want use dashboard.default true
16 | rocketmq.config.enableDashBoardCollect=true
```

```
server.contextPath=
server.port=8080
#spring.application.index=true
spring.application.name=rocketmq-console
spring.http.encoding.charset=UTF-8
spring.http.encoding.enabled=true
spring.http.encoding.force=true
logging.config=classpath:logback.xml
#if this value is empty,use env value rocketmq.config.namesrvAddr NAMESRV_ADDR | now, you can set it in ops page.default localhost:9876
rocketmq.config.namesrvAddr=127.0.0.1:9876
#if you use rocketmq version < 3.5.8, rocketmq.config.isVIPChannel should be false.default true
rocketmq.config.isVIPChannel=
#rocketmq-console's data path:dashboard/monitor
rocketmq.config.dataPath=/tmp/rocketmq-console/data
#set it false if you don't want use dashboard.default true
rocketmq.config.enableDashBoardCollect=true
```

打包

```
1 | mvn clean package -DskipTests
```

```
1 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
  console> ls
2
3
4  目录: C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-
  1.0.0\rocketmq-console
5
6
7  Mode                LastWriteTime         Length Name
8  ----                -
9  d-----          2017/6/15      10:47          doc
10 d-----          2017/6/15      10:47          src
11 d-----          2017/6/15      10:47          style
12 d-----          2022/12/3       16:24          target
13 -----          2017/6/15      10:47          23 .gitignore
14 -----          2017/6/15      10:47          322 .travis.yml
15 -----          2017/6/15      10:47          29843 LICENSE
16 -----          2017/6/15      10:47          176 NOTICE
17 -----          2017/6/15      10:47          11424 pom.xml
18 -----          2017/6/15      10:47          2169 README.md
19
20
21 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
  console> mvn clean package -DskipTests
22 [INFO] Scanning for projects...
23 [INFO]
24 [INFO] -----< org.apache:rocketmq-console-ng >-----
  ----
25 [INFO] Building rocketmq-console-ng 1.0.0
26 [INFO] -----[ jar ]-----
  ----
27 [INFO]
28 [INFO] --- maven-clean-plugin:2.6.1:clean (default-clean) @ rocketmq-
  console-ng ---
29 [INFO] Deleting C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-
  1.0.0\rocketmq-console\target
30 [INFO]
31 [INFO] --- maven-checkstyle-plugin:2.17:check (validate) @ rocketmq-
  console-ng ---
32 [INFO] Starting audit...
33 Audit done.
34 [INFO]
35 [INFO] --- jacoco-maven-plugin:0.7.9:prepare-agent (default-prepare-agent)
  @ rocketmq-console-ng ---
36 [INFO] argLine set to -
  javaagent:C:\\Users\\mao\\.m2\\repository\\org\\jacoco\\org.jacoco.agent\\0
  .7.9\\org.jacoco.agent-0.7.9-
  runtime.jar=destfile=C:\\Users\\mao\\Desktop\\rocketmq-externals-rocketmq-
  console-1.0.0\\rocketmq-console\\target\\jacoco.exec
37 [INFO]
```

```

38 [INFO] --- maven-resources-plugin:2.6:resources (default-resources) @
rocketmq-console-ng ---
39 [INFO] Using 'UTF-8' encoding to copy filtered resources.
40 [INFO] Copying 1 resource
41 [INFO] Copying 960 resources
42 [INFO]
43 [INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ rocketmq-
console-ng ---
44 [INFO] Changes detected - recompiling the module!
45 [INFO] Compiling 59 source files to C:\Users\mao\Desktop\rocketmq-
externals-rocketmq-console-1.0.0\rocketmq-console\target\classes
46 [WARNING] /C:/Users/mao/Desktop/rocketmq-externals-rocketmq-console-
1.0.0/rocketmq-
console/src/main/java/org/apache/rocketmq/console/task/DashboardCollectTask
.java: 某些输入文件使用或覆盖了已过时的 API。
47 [WARNING] /C:/Users/mao/Desktop/rocketmq-externals-rocketmq-console-
1.0.0/rocketmq-
console/src/main/java/org/apache/rocketmq/console/task/DashboardCollectTask
.java: 有关详细信息, 请使用 -Xlint:deprecation 重新编译。
48 [WARNING] /C:/Users/mao/Desktop/rocketmq-externals-rocketmq-console-
1.0.0/rocketmq-
console/src/main/java/org/apache/rocketmq/console/support/GlobalRestfulResp
onseBodyAdvice.java: C:\Users\mao\Desktop\rocketmq-externals-rocketmq-
console-1.0.0\rocketmq-
console\src\main\java\org\apache\rocketmq\console\support\GlobalRestfulResp
onseBodyAdvice.java使用了未经检查或不安全的操作。
49 [WARNING] /C:/Users/mao/Desktop/rocketmq-externals-rocketmq-console-
1.0.0/rocketmq-
console/src/main/java/org/apache/rocketmq/console/support/GlobalRestfulResp
onseBodyAdvice.java: 有关详细信息, 请使用 -Xlint:unchecked 重新编译。
50 [INFO]
51 [INFO] --- maven-resources-plugin:2.6:testResources (default-testResources)
@ rocketmq-console-ng ---
52 [INFO] Using 'UTF-8' encoding to copy filtered resources.
53 [INFO] Copying 3 resources
54 [INFO]
55 [INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @
rocketmq-console-ng ---
56 [INFO] Changes detected - recompiling the module!
57 [INFO] Compiling 15 source files to C:\Users\mao\Desktop\rocketmq-
externals-rocketmq-console-1.0.0\rocketmq-console\target\test-classes
58 [INFO]
59 [INFO] --- maven-surefire-plugin:2.19.1:test (default-test) @ rocketmq-
console-ng ---
60 [INFO] Tests are skipped.
61 [INFO]
62 [INFO] --- maven-jar-plugin:2.6:jar (default-jar) @ rocketmq-console-ng ---
63 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/ple
xus/plexus-archiver/2.9/plexus-archiver-2.9.pom
64 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/ple
xus/plexus-archiver/2.9/plexus-archiver-2.9.pom (4.4 kB at 5.4 kB/s)
65 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/ple
xus/plexus-io/2.4/plexus-io-2.4.pom

```

66 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/plexus/plexus-io/2.4/plexus-io-2.4.pom (3.7 kB at 9.8 kB/s)

67 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/plexus/plexus-archiver/2.9/plexus-archiver-2.9.jar

68 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/commons/commons-compress/1.9/commons-compress-1.9.jar

69 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/plexus/plexus-io/2.4/plexus-io-2.4.jar

70 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/plexus/plexus-archiver/2.9/plexus-archiver-2.9.jar (145 kB at 276 kB/s)

71 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/codehaus/plexus/plexus-io/2.4/plexus-io-2.4.jar (81 kB at 146 kB/s)

72 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/commons/commons-compress/1.9/commons-compress-1.9.jar (378 kB at 491 kB/s)

73 [INFO] Building jar: C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-console\target\rocketmq-console-ng-1.0.0.jar

74 [INFO]

75 [INFO] --- spring-boot-maven-plugin:1.4.3.RELEASE:repackage (default) @ rocketmq-console-ng ---

76 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/springframework/boot/spring-boot-loader-tools/1.4.3.RELEASE/spring-boot-loader-tools-1.4.3.RELEASE.pom

77 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/springframework/boot/spring-boot-loader-tools/1.4.3.RELEASE/spring-boot-loader-tools-1.4.3.RELEASE.pom (3.8 kB at 9.9 kB/s)

78 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-aether-provider/3.2.1/maven-aether-provider-3.2.1.pom

79 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-aether-provider/3.2.1/maven-aether-provider-3.2.1.pom (4.1 kB at 9.8 kB/s)

80 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-model-builder/3.2.1/maven-model-builder-3.2.1.pom

81 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-model-builder/3.2.1/maven-model-builder-3.2.1.pom (2.8 kB at 9.4 kB/s)

82 Downloading from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-repository-metadata/3.2.1/maven-repository-metadata-3.2.1.pom

83 Downloaded from alimaven:
http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-repository-metadata/3.2.1/maven-repository-metadata-3.2.1.pom (2.2 kB at 5.4 kB/s)

```

84 Downloading from alimaven:
   http://maven.aliyun.com/nexus/content/repositories/central/org/springframework/boot/spring-boot-loader-tools/1.4.3.RELEASE/spring-boot-loader-tools-1.4.3.RELEASE.jar
85 Downloading from alimaven:
   http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-aether-provider/3.2.1/maven-aether-provider-3.2.1.jar
86 Downloaded from alimaven:
   http://maven.aliyun.com/nexus/content/repositories/central/org/apache/maven/maven-aether-provider/3.2.1/maven-aether-provider-3.2.1.jar (61 kB at 161 kB/s)
87 Downloaded from alimaven:
   http://maven.aliyun.com/nexus/content/repositories/central/org/springframework/boot/spring-boot-loader-tools/1.4.3.RELEASE/spring-boot-loader-tools-1.4.3.RELEASE.jar (145 kB at 237 kB/s)
88 [INFO]
89 [INFO] >>> maven-source-plugin:3.0.1:jar (attach-sources) > generate-sources @ rocketmq-console-ng >>>
90 [INFO]
91 [INFO] --- maven-checkstyle-plugin:2.17:check (validate) @ rocketmq-console-ng ---
92 [INFO] Starting audit...
93 Audit done.
94 [INFO]
95 [INFO] --- jacoco-maven-plugin:0.7.9:prepare-agent (default-prepare-agent) @ rocketmq-console-ng ---
96 [INFO] argLine set to -
   javaagent:C:\\Users\\mao\\.m2\\repository\\org\\jacoco\\org.jacoco.agent\\0.7.9\\org.jacoco.agent-0.7.9-
   runtime.jar=destfile=C:\\Users\\mao\\Desktop\\rocketmq-externals-rocketmq-console-1.0.0\\rocketmq-console\\target\\jacoco.exec
97 [INFO]
98 [INFO] <<< maven-source-plugin:3.0.1:jar (attach-sources) < generate-sources @ rocketmq-console-ng <<<
99 [INFO]
100 [INFO]
101 [INFO] --- maven-source-plugin:3.0.1:jar (attach-sources) @ rocketmq-console-ng ---
102 [INFO] Building jar: C:\\Users\\mao\\Desktop\\rocketmq-externals-rocketmq-console-1.0.0\\rocketmq-console\\target\\rocketmq-console-ng-1.0.0-sources.jar
103 [INFO] -----
   ----
104 [INFO] BUILD SUCCESS
105 [INFO] -----
   ----
106 [INFO] Total time: 21.584 s
107 [INFO] Finished at: 2022-12-03T16:26:22+08:00
108 [INFO] -----
   ----
109 PS C:\\Users\\mao\\Desktop\\rocketmq-externals-rocketmq-console-1.0.0\\rocketmq-console>

```

启动

```
1 | java -jar rocketmq-console-ng-1.0.0.jar
```

```
1 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
  console> ls
2
3
4 目录: C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-
  1.0.0\rocketmq-console
5
6
7 Mode                LastWriteTime         Length Name
8 ----                -
9 d-----            2017/6/15      10:47      doc
10 d-----            2017/6/15      10:47      src
11 d-----            2017/6/15      10:47      style
12 d-----            2022/12/3      16:26      target
13 -----            2017/6/15      10:47         23 .gitignore
14 -----            2017/6/15      10:47        322 .travis.yml
15 -----            2017/6/15      10:47       29843 LICENSE
16 -----            2017/6/15      10:47        176 NOTICE
17 -----            2017/6/15      10:47       11424 pom.xml
18 -----            2017/6/15      10:47        2169 README.md
19
20
21 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
  console> cd .\target\
22 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
  console\target> ls
23
24
25 目录: C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-
  1.0.0\rocketmq-console\target
26
27
28 Mode                LastWriteTime         Length Name
29 ----                -
30 d-----            2022/12/3      16:26      classes
31 d-----            2022/12/3      16:26      generated-sources
32 d-----            2022/12/3      16:26      generated-test-sources
33 d-----            2022/12/3      16:26      maven-archiver
34 d-----            2022/12/3      16:26      maven-status
35 d-----            2022/12/3      16:26      test-classes
36 -a----            2022/12/3      16:26       11081 checkstyle-cachefile
37 -a----            2022/12/3      16:26        5852 checkstyle-checker.xml
38 -a----            2022/12/3      16:26       10990 checkstyle-result.xml
39 -a----            2022/12/3      16:26      3740409 rocketmq-console-ng-
  1.0.0-sources.jar
40 -a----            2022/12/3      16:26      29611766 rocketmq-console-ng-
  1.0.0.jar
41 -a----            2022/12/3      16:26      3765861 rocketmq-console-ng-
  1.0.0.jar.original
42
```

```

43
44 PS C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
console\target> java8 -jar .\rocketmq-console-ng-1.0.0.jar
45 16:29:39,433 |-INFO in ch.qos.logback.classic.LoggerContext[default] -
Could NOT find resource [logback.groovy]
46 16:29:39,433 |-INFO in ch.qos.logback.classic.LoggerContext[default] -
Could NOT find resource [logback-test.xml]
47 16:29:39,433 |-INFO in ch.qos.logback.classic.LoggerContext[default] -
Found resource [logback.xml] at [jar:file:/C:/Users/mao/Desktop/rocketmq-
externals-rocketmq-console-1.0.0/rocketmq-console/target/rocketmq-console-
ng-1.0.0.jar!/BOOT-INF/classes!/logback.xml]
48 16:29:39,458 |-INFO in
ch.qos.logback.core.joran.spi.ConfigurationWatchList@b1bc7ed - URL
[jar:file:/C:/Users/mao/Desktop/rocketmq-externals-rocketmq-console-
1.0.0/rocketmq-console/target/rocketmq-console-ng-1.0.0.jar!/BOOT-
INF/classes!/logback.xml] is not of type file
49 16:29:39,513 |-INFO in
ch.qos.logback.classic.joran.action.ConfigurationAction - debug attribute
not set
50 16:29:39,524 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
About to instantiate appender of type [ch.qos.logback.core.ConsoleAppender]
51 16:29:39,533 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
Naming appender as [STDOUT]
52 16:29:39,541 |-INFO in
ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default
type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] for [encoder]
property
53 16:29:39,592 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
About to instantiate appender of type
[ch.qos.logback.core.rolling.RollingFileAppender]
54 16:29:39,595 |-INFO in ch.qos.logback.core.joran.action.AppenderAction -
Naming appender as [FILE]
55 16:29:39,616 |-INFO in c.q.l.core.rolling.TimeBasedRollingPolicy@2094548358
- No compression will be used
56 16:29:39,618 |-INFO in c.q.l.core.rolling.TimeBasedRollingPolicy@2094548358
- Will use the pattern C:/Users/mao/logs/consolelogs/rocketmq-console-
%d{yyyy-MM-dd}%.i.log for the active file
57 16:29:39,621 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@30dae81 - The date
pattern is 'yyyy-MM-dd' from file name pattern
'C:/Users/mao/logs/consolelogs/rocketmq-console-%d{yyyy-MM-dd}%.i.log'.
58 16:29:39,621 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@30dae81 - Roll-over at
midnight.
59 16:29:39,624 |-INFO in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@30dae81 - Setting initial
period to Sat Dec 03 16:29:03 CST 2022
60 16:29:39,625 |-WARN in
ch.qos.logback.core.rolling.SizeAndTimeBasedFNATP@30dae81 -
SizeAndTimeBasedFNATP is deprecated. Use SizeAndTimeBasedRollingPolicy
instead
61 16:29:39,628 |-INFO in
ch.qos.logback.core.joran.action.NestedComplexPropertyIA - Assuming default
type [ch.qos.logback.classic.encoder.PatternLayoutEncoder] for [encoder]
property
62 16:29:39,630 |-INFO in
ch.qos.logback.core.rolling.RollingFileAppender[FILE] - Active log file
name: C:\Users\mao\logs\consolelogs\rocketmq-console.log

```

```

63 16:29:39,630 |-INFO in ch.qos.logback.core.rolling.RollingFileAppender[FILE] - File property is
ch.qos.logback.core.rolling.RollingFileAppender[FILE] - File property is
set to [C:\Users\mao\logs\consolelogs\rocketmq-console.log]
64 16:29:39,631 |-INFO in ch.qos.logback.classic.joran.action.RootLoggerAction
- Setting level of ROOT logger to INFO
65 16:29:39,631 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction -
Attaching appender named [STDOUT] to Logger[ROOT]
66 16:29:39,632 |-INFO in ch.qos.logback.core.joran.action.AppenderRefAction -
Attaching appender named [FILE] to Logger[ROOT]
67 16:29:39,632 |-INFO in
ch.qos.logback.classic.joran.action.ConfigurationAction - End of
configuration.
68 16:29:39,633 |-INFO in
ch.qos.logback.classic.joran.JoranConfigurator@1b2c6ec2 - Registering
current configuration as safe fallback point
69
70
71 . ____ _ _ _
72 /\ / _ ' _ _ ( ) _ _ _ _ \ \ \ \
73 ( ( ) _ _ | ' _ | ' _ | ' _ \ / _ | \ \ \ \
74 \ \ _ _ | | _ | | | | | | ( _ | ) ) ) )
75 ' | _ _ | . _ | | _ | | _ | | _ , | / / / /
76 =====|_|=====|__=/_/_/_/_/
77 :: Spring Boot :: (v1.4.3.RELEASE)
78
79 [2022-12-03 16:29:40.271] INFO Starting App v1.0.0 on mao with PID 10420
(C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
console\target\rocketmq-console-ng-1.0.0.jar started by mao in
C:\Users\mao\Desktop\rocketmq-externals-rocketmq-console-1.0.0\rocketmq-
console\target)
80 [2022-12-03 16:29:40.275] INFO No active profile set, falling back to
default profiles: default
81 [2022-12-03 16:29:40.357] INFO Refreshing
org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplic
ationContext@6477463f: startup date [Sat Dec 03 16:29:40 CST 2022]; root of
context hierarchy
82 [2022-12-03 16:29:40.408] INFO HV000001: Hibernate validator 5.2.4.Final
83 [2022-12-03 16:29:42.758] INFO Tomcat initialized with port(s): 8080
(http)
84 [2022-12-03 16:29:42.773] INFO Starting service Tomcat
85 [2022-12-03 16:29:42.775] INFO Starting Servlet Engine: Apache
Tomcat/8.5.6
86 [2022-12-03 16:29:42.855] INFO Initializing Spring embedded
WebApplicationContext
87 [2022-12-03 16:29:42.855] INFO Root WebApplicationContext: initialization
completed in 2498 ms
88 [2022-12-03 16:29:43.224] INFO Mapping servlet: 'dispatcherServlet' to [/]
89 [2022-12-03 16:29:43.230] INFO Mapping filter: 'metricsFilter' to: [/]
90 [2022-12-03 16:29:43.230] INFO Mapping filter: 'characterEncodingFilter'
to: [/]
91 [2022-12-03 16:29:43.230] INFO Mapping filter: 'hiddenHttpMethodFilter'
to: [/]
92 [2022-12-03 16:29:43.231] INFO Mapping filter: 'httpPutFormContentFilter'
to: [/]
93 [2022-12-03 16:29:43.231] INFO Mapping filter: 'requestContextFilter' to:
[/]
94 [2022-12-03 16:29:43.232] INFO Mapping filter: 'webRequestLoggingFilter'
to: [/]

```



```

95 [2022-12-03 16:29:43.232] INFO Mapping filter:
   'applicationContextIdFilter' to: [//*]
96 [2022-12-03 16:29:43.291] INFO setNameSrvAddrByProperty
   nameSrvAddr=127.0.0.1:9876
97 [2022-12-03 16:29:44.285] INFO Looking for @ControllerAdvice:
   org.springframework.boot.context.embedded.AnnotationConfigEmbeddedWebApplic
   ationContext@6477463f: startup date [Sat Dec 03 16:29:40 CST 2022]; root of
   context hierarchy
98 [2022-12-03 16:29:44.292] INFO Detected ResponseBodyAdvice bean in
   globalRestfulResponseBodyAdvice
99 [2022-12-03 16:29:44.402] INFO Mapped "
   {[/cluster/brokerConfig.query],methods=[GET]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ClusterController.brokerConfig(java.
   lang.String)
100 [2022-12-03 16:29:44.404] INFO Mapped "{[/cluster/list.query],methods=
   [GET]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ClusterController.list()
101 [2022-12-03 16:29:44.408] INFO Mapped "
   {[/consumer/examineSubscriptionGroupConfig.query]}" onto public
   java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.examineSubscripti
   onGroupConfig(java.lang.String)
102 [2022-12-03 16:29:44.409] INFO Mapped "
   {[/consumer/createOrUpdate.do],methods=[POST]}" onto public
   java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.consumerCreateOrU
   pdateRequest(org.apache.rocketmq.console.model.request.ConsumerConfigInfo)
103 [2022-12-03 16:29:44.410] INFO Mapped "
   {[/consumer/queryTopicByConsumer.query]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.queryConsumerByTo
   pic(java.lang.String)
104 [2022-12-03 16:29:44.410] INFO Mapped "
   {[/consumer/consumerRunningInfo.query]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.getConsumerRunnin
   gInfo(java.lang.String,java.lang.String,boolean)
105 [2022-12-03 16:29:44.411] INFO Mapped "
   {[/consumer/fetchBrokerNameList.query],methods=[GET]}" onto public
   java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.fetchBrokerNameLi
   st(java.lang.String)
106 [2022-12-03 16:29:44.411] INFO Mapped "
   {[/consumer/consumerConnection.query]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.consumerConnectio
   n(java.lang.String)
107 [2022-12-03 16:29:44.412] INFO Mapped "
   {[/consumer/resetOffset.do],methods=[POST]}" onto public java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.resetOffset(org.a
   pache.rocketmq.console.model.request.ResetOffsetRequest)
108 [2022-12-03 16:29:44.413] INFO Mapped "
   {[/consumer/deleteSubGroup.do],methods=[POST]}" onto public
   java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.deleteSubGroup(or
   g.apache.rocketmq.console.model.request.DeleteSubGroupRequest)
109 [2022-12-03 16:29:44.413] INFO Mapped "{[/consumer/group.query]}" onto
   public java.lang.Object
   org.apache.rocketmq.console.controller.ConsumerController.groupQuery(java.l
   ang.String)

```

```

110 [2022-12-03 16:29:44.414] INFO Mapped "{[/consumer/groupList.query]}" onto
    public java.lang.Object
    org.apache.rocketmq.console.controller.ConsumerController.list()
111 [2022-12-03 16:29:44.415] INFO Mapped "{[/dashboard/topic.query],methods=
    [GET]}" onto public java.lang.Object
    org.apache.rocketmq.console.controller.DashboardController.topic(java.lang.
    String,java.lang.String)
112 [2022-12-03 16:29:44.416] INFO Mapped "{[/dashboard/topicCurrent],methods=
    [GET]}" onto public java.lang.Object
    org.apache.rocketmq.console.controller.DashboardController.topicCurrent()
113 [2022-12-03 16:29:44.416] INFO Mapped "{[/dashboard/broker.query],methods=
    [GET]}" onto public java.lang.Object
    org.apache.rocketmq.console.controller.DashboardController.broker(java.lang
    .String)
114 [2022-12-03 16:29:44.420] INFO Mapped "
    {[/message/consumeMessageDirectly.do],methods=[POST]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MessageController.consumeMessageDire
    ctly(java.lang.String,java.lang.String,java.lang.String,java.lang.String)
115 [2022-12-03 16:29:44.420] INFO Mapped "
    {[/message/queryMessageByTopicAndKey.query],methods=[GET]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MessageController.queryMessageByTopi
    cAndKey(java.lang.String,java.lang.String)
116 [2022-12-03 16:29:44.421] INFO Mapped "
    {[/message/queryMessageByTopic.query],methods=[GET]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MessageController.queryMessageByTopi
    c(java.lang.String,long,long)
117 [2022-12-03 16:29:44.421] INFO Mapped "
    {[/message/viewMessage.query],methods=[GET]}" onto public java.lang.Object
    org.apache.rocketmq.console.controller.MessageController.viewMessage(java.l
    ang.String,java.lang.String)
118 [2022-12-03 16:29:44.422] INFO Mapped "
    {[/monitor/deleteConsumerMonitor.do],methods=[POST]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MonitorController.deleteConsumerMoni
    tor(java.lang.String)
119 [2022-12-03 16:29:44.423] INFO Mapped "
    {[/monitor/consumerMonitorConfig.query],methods=[GET]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MonitorController.consumerMonitorCon
    fig()
120 [2022-12-03 16:29:44.423] INFO Mapped "
    {[/monitor/createOrUpdateConsumerMonitor.do],methods=[POST]}" onto public
    java.lang.Object
    org.apache.rocketmq.console.controller.MonitorController.createOrUpdateCons
    umerMonitor(java.lang.String,int,int)
121 [2022-12-03 16:29:44.423] INFO Mapped "
    {[/monitor/consumerMonitorConfigByGroupName.query],methods=[GET]}" onto
    public java.lang.Object
    org.apache.rocketmq.console.controller.MonitorController.consumerMonitorCon
    figByGroupName(java.lang.String)
122 [2022-12-03 16:29:44.424] INFO Mapped "{[/rocketmq/nsaddr],methods=[GET]}"
    onto public java.lang.Object
    org.apache.rocketmq.console.controller.NamesvrController.nsaddr()

```

123 [2022-12-03 16:29:44.425] INFO Mapped "
 {[/ops/updateNameSvrAddr.do],methods=[POST]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.OpsController.updateNameSvrAddr(java
 .lang.String)

124 [2022-12-03 16:29:44.425] INFO Mapped "
 {[/ops/updateIsVIPChannel.do],methods=[POST]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.OpsController.updateIsVIPChannel(jav
 a.lang.String)

125 [2022-12-03 16:29:44.425] INFO Mapped "{[/ops/homePage.query],methods=
 [GET]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.OpsController.homePage()

126 [2022-12-03 16:29:44.426] INFO Mapped "
 {[/ops/rocketMqStatus.query],methods=[GET]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.OpsController.clusterStatus()

127 [2022-12-03 16:29:44.427] INFO Mapped "
 {[/producer/producerConnection.query],methods=[GET]}" onto public
 java.lang.Object
 org.apache.rocketmq.console.controller.ProducerController.producerConnectio
 n(java.lang.String,java.lang.String)

128 [2022-12-03 16:29:44.427] INFO Mapped "{[/test/runTask.do],methods=[GET]}"
 onto public java.lang.Object
 org.apache.rocketmq.console.controller.TestController.list() throws
 org.apache.rocketmq.client.exception.MQClientException,org.apache.rocketmq.
 remoting.exception.RemotingException,java.lang.InterruptedException

129 [2022-12-03 16:29:44.429] INFO Mapped "
 {[/topic/queryConsumerByTopic.query]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.queryConsumerByTopic
 (java.lang.String)

130 [2022-12-03 16:29:44.429] INFO Mapped "
 {[/topic/queryTopicConsumerInfo.query]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.queryTopicConsumerIn
 fo(java.lang.String)

131 [2022-12-03 16:29:44.430] INFO Mapped "
 {[/topic/deleteTopicByBroker.do],methods=[POST]}" onto public
 java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.deleteTopicByBroker(
 java.lang.String,java.lang.String)

132 [2022-12-03 16:29:44.430] INFO Mapped "
 {[/topic/examineTopicConfig.query]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.examineTopicConfig(j
 ava.lang.String,java.lang.String) throws
 org.apache.rocketmq.remoting.exception.RemotingException,org.apache.rocketm
 q.client.exception.MQClientException,java.lang.InterruptedException

133 [2022-12-03 16:29:44.430] INFO Mapped "
 {[/topic/createOrUpdate.do],methods=[POST]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.topicCreateOrUpdateR
 equest(org.apache.rocketmq.console.model.request.TopicConfigInfo)

134 [2022-12-03 16:29:44.431] INFO Mapped "{[/topic/stats.query],methods=
 [GET]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.stats(java.lang.Stri
 ng)

135 [2022-12-03 16:29:44.431] INFO Mapped "{[/topic/route.query],methods=
 [GET]}" onto public java.lang.Object
 org.apache.rocketmq.console.controller.TopicController.route(java.lang.Stri
 ng)

```

136 [2022-12-03 16:29:44.432] INFO Mapped "{[/topic/sendTopicMessage.do],methods=[POST]}" onto public java.lang.Object
org.apache.rocketmq.console.controller.TopicController.sendMessage(org
.apache.rocketmq.console.model.request.SendMessageRequest) throws
org.apache.rocketmq.remoting.exception.RemotingException,org.apache.rocketm
q.client.exception.MQClientException,java.lang.InterruptedException
137 [2022-12-03 16:29:44.433] INFO Mapped "{[/topic/deleteTopic.do],methods=
[POST]}" onto public java.lang.Object
org.apache.rocketmq.console.controller.TopicController.delete(java.lang.Str
ing,java.lang.String)
138 [2022-12-03 16:29:44.437] INFO Mapped "{[/topic/list.query],methods=
[GET]}" onto public java.lang.Object
org.apache.rocketmq.console.controller.TopicController.list() throws
org.apache.rocketmq.client.exception.MQClientException,org.apache.rocketmq.
remoting.exception.RemotingException,java.lang.InterruptedException
139 [2022-12-03 16:29:44.438] INFO Mapped "{[/error],produces=[text/html]}"
onto public org.springframework.web.servlet.ModelAndView
org.springframework.boot.autoconfigure.web.BasicErrorController.errorHtml(j
avax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletResponse
)
140 [2022-12-03 16:29:44.439] INFO Mapped "{[/error]}" onto public
org.springframework.http.ResponseEntity<java.util.Map<java.lang.String,
java.lang.Object>>
org.springframework.boot.autoconfigure.web.BasicErrorController.error(javax
.servlet.http.HttpServletRequest)
141 [2022-12-03 16:29:44.524] INFO Mapped URL path [/webjars/**] onto handler
of type [class
org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
142 [2022-12-03 16:29:44.525] INFO Mapped URL path [/**] onto handler of type
[class org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
143 [2022-12-03 16:29:44.550] INFO Detected @ExceptionHandler methods in
globalExceptionHandler
144 [2022-12-03 16:29:44.550] INFO Detected.ResponseBodyAdvice implementation
in globalRestfulResponseBodyAdvice
145 [2022-12-03 16:29:44.599] INFO Mapped URL path [/**/favicon.ico] onto
handler of type [class
org.springframework.web.servlet.resource.ResourceHttpRequestHandler]
146 [2022-12-03 16:29:44.647] INFO Adding welcome page: class path resource
[static/index.html]
147 [2022-12-03 16:29:45.174] INFO Mapped "{[/beans || /beans.json],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
148 [2022-12-03 16:29:45.176] INFO Mapped "{[/heapdump ||
/heapdump.json],methods=[GET],produces=[application/octet-stream]}" onto
public void
org.springframework.boot.actuate.endpoint.mvc.HeapdumpMvcEndpoint.invoke(bo
olean,javax.servlet.http.HttpServletRequest,javax.servlet.http.HttpServletR
esponse) throws java.io.IOException,javax.servlet.ServletException
149 [2022-12-03 16:29:45.177] INFO Mapped "{[/mappings ||
/mappings.json],methods=[GET],produces=[application/json]}" onto public
java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
150 [2022-12-03 16:29:45.177] INFO Mapped "{[/dump || /dump.json],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
151 [2022-12-03 16:29:45.178] INFO Mapped "{[/info || /info.json],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()

```

```

152 [2022-12-03 16:29:45.179] INFO Mapped "{[/trace || /trace.json],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
153 [2022-12-03 16:29:45.180] INFO Mapped "{[/metrics/{name:.*}],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.MetricsMvcEndpoint.value(java
.lang.String)
154 [2022-12-03 16:29:45.181] INFO Mapped "{[/metrics ||
/metrics.json],methods=[GET],produces=[application/json]}" onto public
java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
155 [2022-12-03 16:29:45.182] INFO Mapped "{[/autoconfig ||
/autoconfig.json],methods=[GET],produces=[application/json]}" onto public
java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
156 [2022-12-03 16:29:45.184] INFO Mapped "{[/env/{name:.*}],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EnvironmentMvcEndpoint.value(
java.lang.String)
157 [2022-12-03 16:29:45.185] INFO Mapped "{[/env || /env.json],methods=
[GET],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
158 [2022-12-03 16:29:45.185] INFO Mapped "{[/health ||
/health.json],produces=[application/json]}" onto public java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.HealthMvcEndpoint.invoke(java
.security.Principal)
159 [2022-12-03 16:29:45.186] INFO Mapped "{[/configprops ||
/configprops.json],methods=[GET],produces=[application/json]}" onto public
java.lang.Object
org.springframework.boot.actuate.endpoint.mvc.EndpointMvcAdapter.invoke()
160 [2022-12-03 16:29:45.325] INFO Registering beans for JMX exposure on
startup
161 [2022-12-03 16:29:45.333] INFO Registering beans for JMX exposure on
startup
162 [2022-12-03 16:29:45.340] INFO Starting beans in phase 0
163 [2022-12-03 16:29:45.350] INFO Located managed bean
'requestMappingEndpoint': registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=requestMappingEndpoint]
164 [2022-12-03 16:29:45.385] INFO Located managed bean 'environmentEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=environmentEndpoint]
165 [2022-12-03 16:29:45.391] INFO Located managed bean 'healthEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=healthEndpoint]
166 [2022-12-03 16:29:45.394] INFO Located managed bean 'beansEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=beansEndpoint]
167 [2022-12-03 16:29:45.400] INFO Located managed bean 'infoEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=infoEndpoint]
168 [2022-12-03 16:29:45.403] INFO Located managed bean 'metricsEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=metricsEndpoint]
169 [2022-12-03 16:29:45.406] INFO Located managed bean 'traceEndpoint':
registering with JMX server as MBean
[org.springframework.boot:type=Endpoint,name=traceEndpoint]

```

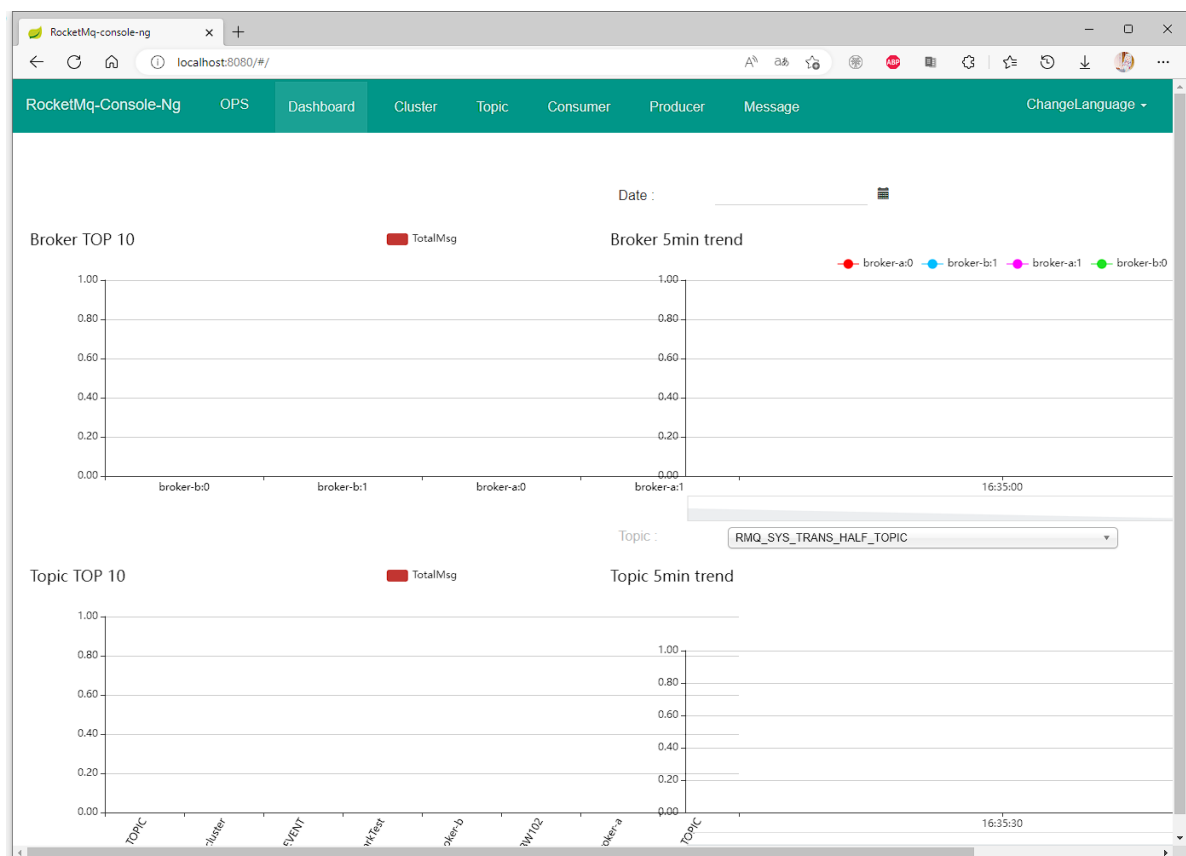
```

170 [2022-12-03 16:29:45.408] INFO Located managed bean 'dumpEndpoint':
    registering with JMX server as MBean
    [org.springframework.boot:type=Endpoint,name=dumpEndpoint]
171 [2022-12-03 16:29:45.414] INFO Located managed bean
    'autoConfigurationReportEndpoint': registering with JMX server as MBean
    [org.springframework.boot:type=Endpoint,name=autoConfigurationReportEndpoi
    nt]
172 [2022-12-03 16:29:45.417] INFO Located managed bean
    'configurationPropertiesReportEndpoint': registering with JMX server as
    MBean
    [org.springframework.boot:type=Endpoint,name=configurationPropertiesReportE
    ndpoint]
173 [2022-12-03 16:29:45.430] INFO No TaskScheduler/ScheduledExecutorService
    bean found for scheduled processing
174 [2022-12-03 16:29:45.447] INFO Initializing ProtocolHandler ["http-nio-
    8080"]
175 [2022-12-03 16:29:45.462] INFO Starting ProtocolHandler [http-nio-8080]
176 [2022-12-03 16:29:45.476] INFO Using a shared selector for servlet
    write/read
177 [2022-12-03 16:29:45.497] INFO Tomcat started on port(s): 8080 (http)
178 [2022-12-03 16:29:45.502] INFO Started App in 5.727 seconds (JVM running
    for 6.463)

```

访问

<http://localhost:8080>

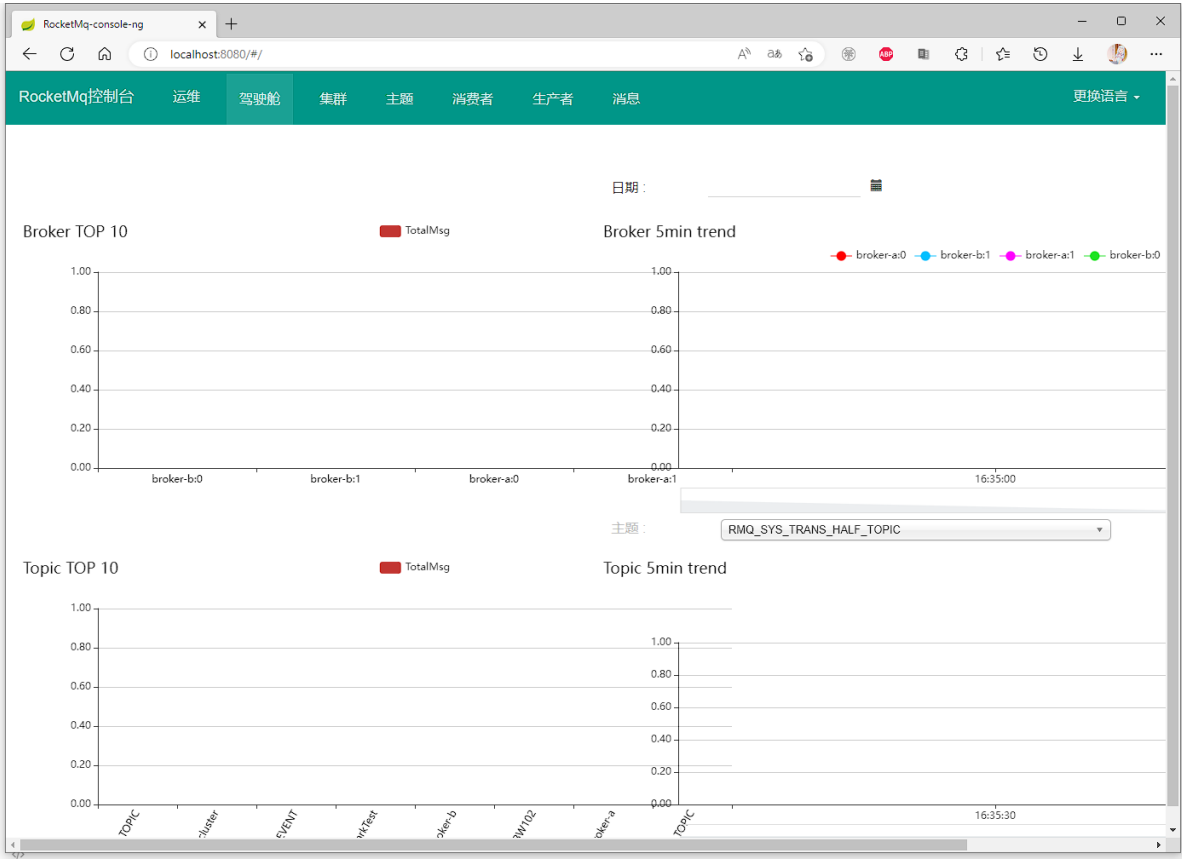


ChangeLanguage ▾

English

Simplified Chinese

broker-b:1 broker-a:1 broker-b:0



RocketMQ控制台

[运维](#)
[驾驶舱](#)
[集群](#)
[主题](#)
[消费者](#)
[生产者](#)
[消息](#)

[更换语言](#)

集群: rocketmq-cluster

分片	编号	地址	版本	生产消息TPS	消费消息TPS	昨日生产总数	昨日消费总数	今天生产总数	今天消费总数	操作
broker-b	0(master)	223.156.159.141:10921	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-b	1(slave)	223.156.159.141:10941	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-a	0(master)	223.156.159.141:10911	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-a	1(slave)	223.156.159.141:10931	V4_5_1	0.00	0.00	0	0	0	0	状态 配置

RocketMQ控制台

[运维](#)
[驾驶舱](#)
[集群](#)
[主题](#)
[消费者](#)
[生产者](#)
[消息](#)

[更换语言](#)

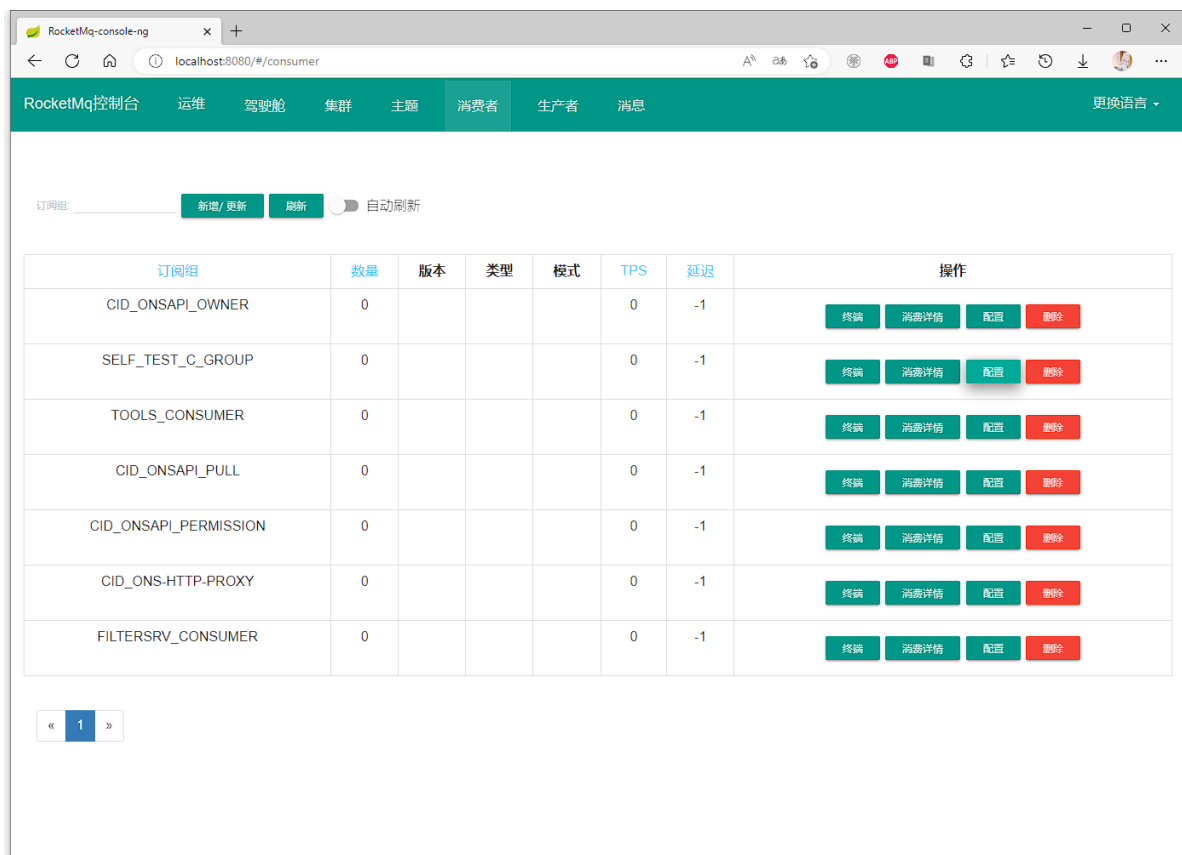
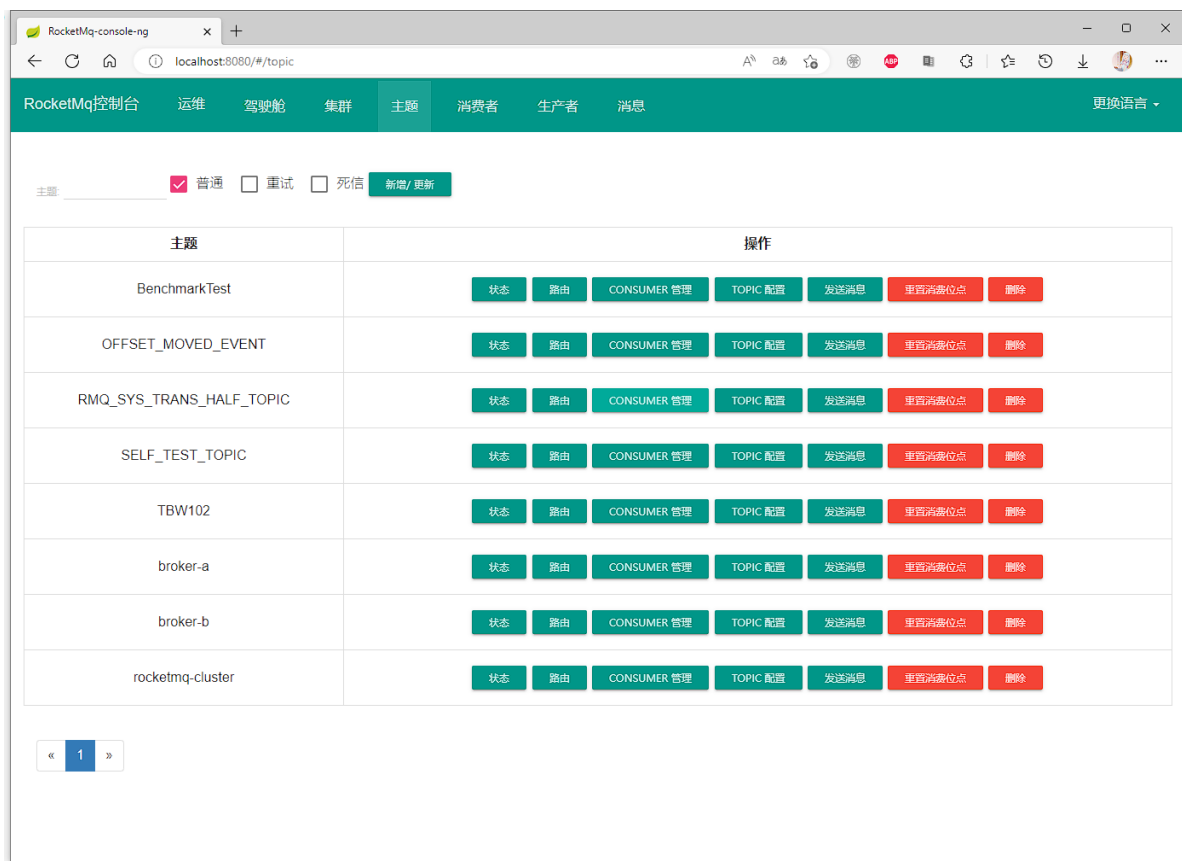
集群: rocketmq-cluster

分片	编号	地址	版本	生产消息TPS	消费消息TPS	昨日生产总数	昨日消费总数	今天生产总数	今天消费总数	操作
broker-b	0(master)	223.156.159.141:10921	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-b	1(slave)	223.156.159.141:10941	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-a	0(master)	223.156.159.141:10911	V4_5_1	0.00	0.00	0	0	0	0	状态 配置
broker-a	1(slave)	223.156.159.141:10931	V4_5_1	0.00	0.00	0	0	0	0	状态 配置

[broker-b][0]

msgPutTotalTodayNow	0
commitLogDiskRatio	0.6582130011369727
getFoundTps	0 0 0 0
EndTransactionThreadPoolQueueCapacity	100000
sendThreadPoolQueueHeadWaitTimeMills	0
putMessageDistributeTime	[<=0ms] 0 [0~10ms] 0 [10~50ms] 0 [50~100ms] 0 [100~200ms] 0 [200~500ms] 0 [500ms~1s] 0 [1~2s] 0 [2~3s] 0 [3~4s] 0 [4~5s] 0 [5~10s] 0 [10s~] 0
queryThreadPoolQueueHeadWaitTimeMills	0
remainHowManyDataToFlush	0 B
msgGetTotalTodayNow	0
commitLogMaxOffset	0
queryThreadPoolQueueSize	0
getMessageEntireTimeMax	1
msgPutTotalTodayMorning	0

关闭



命令介绍

Topic相关

名称	含义	命令选项	说明
updateTopic	创建更新Topic配置	-b	Broker 地址，表示 topic 所在 Broker，只支持单台Broker，地址为ip:port
		-c	cluster 名称，表示 topic 所在集群（集群可通过 clusterList 查询）
		-h-	打印帮助
		-n	NameServer服务地址，格式 ip:port
		-p	指定新topic的读写权限（W=2 R=4 WR=6）
		-r	可读队列数（默认为8）
		-w	可写队列数（默认为8）
		-t	topic 名称（名称只能使用字符 ^[a-zA-Z0-9_-]+\$ ）
deleteTopic	删除Topic	-c	cluster 名称，表示删除某集群下的某个 topic（集群 可通过 clusterList 查询）
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic 名称（名称只能使用字符 ^[a-zA-Z0-9_-]+\$ ）
topicList	查看 Topic 列表信息	-h	打印帮助

		-c	不配置-c只返回topic列表，增加-c返回clusterName, topic, consumerGroup信息，即topic的所属集群和订阅关系，没有参数
		-n	NameServer 服务地址，格式 ip:port
topicRoute	查看 Topic 路由信息	-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
topicStatus	查看 Topic 消息队列 offset	-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
topicClusterList	查看 Topic 所在集群列表	-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
updateTopicPerm	更新 Topic 读写权限	-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-b	Broker 地址，表示 topic 所在 Broker，只支持单台Broker，地址为ip:port
		-p	指定新 topic 的读写权限(W=2 R=4 WR=6)

		-c	cluster 名称，表示 topic 所在集群（集群可通过 clusterList 查询），-b 优先，如果没有 -b，则对集群中所有 Broker 执行命令
updateOrderConf	从 NameServer 上创建、删除、获取特定命名空间的 kv 配置，目前还未启用	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic，键
		-v	orderConf，值
		-m	method，可选 get、put、delete
allocateMQ	以平均负载算法计算消费者列表负载消息队列的负载结果	-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-i	ipList，用逗号分隔，计算这些 ip 去负载 Topic 的消息队列
statsAll	打印 Topic 订阅关系、TPS、积累量、24h 读写总量等信息	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-a	是否只打印活跃 topic
		-t	指定 topic

集群相关

名称	含义	命令选项	说明
clusterList	查看集群信息，集群、BrokerName、BrokerId、TPS等信息	-m	打印更多信息 (增加打印出如下信息 #InTotalYest, #OutTotalYest, #InTotalToday ,#OutTotalToday)
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-i	打印间隔，单位秒
clusterRT	发送消息检测集群各Broker RT。消息发往\${BrokerName} Topic。	-a	amount，每次探测的总数，RT = 总时间 / amount
		-s	消息大小，单位B
		-c	探测哪个集群
		-p	是否打印格式化日志，以 分割，默认不打印
		-h	打印帮助
		-m	所属机房，打印使用
		-i	发送间隔，单位秒
		-n	NameServer 服务地址，格式 ip:port

Broker相关

名称	含义	命令选项	说明
updateBrokerConfig	更新 Broker 配置文件，会修改 Broker.conf	-b	Broker 地址，格式为 ip:port
		-c	cluster 名称
		-k	key 值
		-v	value 值
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
brokerStatus	查看 Broker 统计信息、运行状态（你想要的信息几乎都在里面）	-b	Broker 地址，地址为 ip:port
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
brokerConsumeStats	Broker中各个消费者的消费情况，按 Message Queue维度返回Consume Offset, Broker Offset, Diff, Timestamp等信息	-b	Broker 地址，地址为 ip:port
		-t	请求超时时间
		-l	diff阈值，超过阈值才打印
		-o	是否为顺序topic，一般为false
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
getBrokerConfig	获取Broker配置	-b	Broker 地址，地址为 ip:port
		-n	NameServer 服务地址，格式 ip:port
wipeWritePerm	从NameServer上清除 Broker写权限	-b	Broker 地址，地址为 ip:port
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
cleanExpiredCQ	清理Broker上过期的Consume Queue，如果手动	-n	NameServer 服务地址，格式 ip:port

	减少对列数可能产生过期队列	-h	打印帮助
		-b	Broker 地址，地址为 ip:port
		-c	集群名称
cleanUnusedTopic	清理Broker上不使用的Topic，从内存中释放Topic的 Consume Queue，如果手动删除Topic会产生不使用的 Topic	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-b	Broker 地址，地址为 ip:port
		-c	集群名称
sendMsgStatus	向Broker发消息，返回发送状态和RT	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-b	BrokerName，注意不同于Broker地址
		-s	消息大小，单位B
		-c	发送次数

消息相关

名称	含义	命令选项	说明
queryMsgById	根据offsetMsgId查询msg, 如果使用开源控制台, 应使用offsetMsgId, 此命令还有其他参数, 具体作用请阅读QueryMsgByIdSubCommand。	-i	msgId
		-h	打印帮助
		-n	NameServer 服务地址, 格式 ip:port
queryMsgByKey	根据消息 Key 查询消息	-k	msgKey
		-t	Topic 名称
		-h	打印帮助
		-n	NameServer 服务地址, 格式 ip:port
queryMsgByOffset	根据 Offset 查询消息	-b	Broker 名称, (这里需要注意 填写的是 Broker 的名称, 不是 Broker 的地址, Broker 名称可以在 clusterList 查到)
		-i	query 队列 id
		-o	offset 值
		-t	topic 名称
		-h	打印帮助
		-n	NameServer 服务地址, 格式 ip:port
queryMsgByUniqueKey	根据msgId查询, msgId不同于offsetMsgId, 区别详见常见运维问题。-g, -d配合使用, 查到消息后尝试让特定的消费者消费消息并返回消费结果	-h	打印帮助
		-n	NameServer 服务地址, 格式 ip:port
		-i	unique msg id
		-g	consumerGroup

		-d	clientId
		-t	topic名称
checkMsgSendRT	检测向topic发消息的RT，功能类似clusterRT	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic名称
		-a	探测次数
		-s	消息大小
sendMessage	发送一条消息，可以根据配置发往特定Message Queue，或普通发送。	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic名称
		-p	body，消息体
		-k	keys
		-c	tags
		-b	BrokerName
consumeMessage	消费消息。可以根据offset、开始&结束时间戳、消息队列消费消息，配置不同执行不同消费逻辑，详见 ConsumeMessageCommand。	-i	queueId
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic名称
		-b	BrokerName
		-o	从offset开始消费
		-i	queueId
		-g	消费者分组
		-s	开始时间戳，格式详见-h
		-d	结束时间戳
printMsg	从Broker消费消息并打印，可选时间段	-c	消费多少条消息
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port

		-t	topic名称
		-c	字符集，例如UTF-8
		-s	subExpress，过滤表达式
		-b	开始时间戳，格式参见-h
		-e	结束时间戳
		-d	是否打印消息体
printMsgByQueue	类似printMsg，但指定Message Queue	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-t	topic名称
		-i	queueId
		-a	BrokerName
		-c	字符集，例如UTF-8
		-s	subExpress，过滤表达式
		-b	开始时间戳，格式参见-h
		-e	结束时间戳
		-p	是否打印消息
		-d	是否打印消息体
		-f	是否统计tag数量并打印
resetOffsetByTime	按时间戳重置offset，Broker和consumer都会重置	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-g	消费者分组
		-t	topic名称
		-s	重置为此时间戳对应的offset

		-f	是否强制重置，如果false，只支持回溯offset，如果true，不管时间戳对应offset与consumeOffset关系
		-c	是否重置c++客户端offset

消费者、消费组相关

名称	含义	命令选项	说明
consumerProgress	查看订阅组消费状态，可以查看具体的client IP的消息积累量	-g	消费者所属组名
		-s	是否打印client IP
		-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
consumerStatus	查看消费者状态，包括同一个分组中是否都是相同的订阅，分析Process Queue是否堆积，返回消费者jstack结果，内容较多，使用者参见ConsumerStatusSubCommand	-h	打印帮助
		-n	NameServer 服务地址，格式 ip:port
		-g	consumer group
		-i	clientId
		-s	是否执行jstack
getConsumerStatus	获取 Consumer 消费进度	-g	消费者所属组名
		-t	查询主题
		-i	Consumer 客户端 ip
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
updateSubGroup	更新或创建订阅关系	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-b	Broker地址
		-c	集群名称
		-g	消费者分组名称
		-s	分组是否允许消费
		-m	是否从最小offset开始消费
		-d	是否是广播模式

		-q	重试队列数量
		-r	最大重试次数
		-i	当slaveReadEnable开启时有效，且还未达到从slave消费时建议从哪个BrokerId消费，可以配置备机id，主动从备机消费
		-w	如果Broker建议从slave消费，配置决定从哪个slave消费，配置BrokerId，例如1
		-a	当消费者数量变化时是否通知其他消费者负载均衡
deleteSubGroup	从Broker删除订阅关系	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-b	Broker地址
		-c	集群名称
		-g	消费者分组名称
cloneGroupOffset	在目标群组中使用源群组的offset	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-s	源消费者组
		-d	目标消费者组
		-t	topic名称
		-o	暂未使用

连接相关

名称	含义	命令选项	说明
consumerConnec tion	查询 Consumer 的 网络连接	-g	消费者所属组名
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
producerConnec tion	查询 Producer 的网 络连接	-g	生产者所属组名
		-t	主题名称
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助

NameServer相关

名称	含义	命令选项	说明
updateKvConfig	更新NameServer的kv配置，目前还未使用	-s	命名空间
		-k	key
		-v	value
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
deleteKvConfig	删除NameServer的kv配置	-s	命名空间
		-k	key
		-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
getNamesrvConfig	获取NameServer配置	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
updateNamesrvConfig	修改NameServer配置	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助
		-k	key
		-v	value

其他

名称	含义	命令选项	说明
startMonitoring	开启监控进程，监控消息误删、重试队列消息数等	-n	NameServer 服务地址，格式 ip:port
		-h	打印帮助

消息发送与消费

RocketMQ客户端依赖

```
1      <dependency>
2          <groupId>org.apache.rocketmq</groupId>
3          <artifactId>rocketmq-client</artifactId>
4          <version>4.9.4</version>
5      </dependency>
```

基本消息的发送与接收

消息发送

发送同步消息

这种可靠性同步地发送方式使用的比较广泛，比如：重要的消息通知，短信通知。

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.client.producer.SendResult;
7 import org.apache.rocketmq.common.message.Message;
8 import org.apache.rocketmq.remoting.exception.RemotingException;
9
10 import java.nio.charset.StandardCharsets;
11
12 /**
13  * Project name(项目名称): RocketMQ_基本消息的发送与接收
14  * Package(包名): mao.producer
15  * Class(类名): SyncProducer
16  * Author(作者): mao
17  * Author QQ: 1296193245
18  * GitHub: https://github.com/maomao124/
19  * Date(创建日期): 2022/12/3
20  * Time(创建时间): 17:01
21  * Version(版本): 1.0
22  * Description(描述): 生产者-同步消息
23  */
24
25 public class SyncProducer
26 {
27     public static void main(String[] args)
```

```

28         throws MQClientException, MQBrokerException, RemotingException,
InterruptedException
29     {
30         //生产者, 参数为生产者组名
31         DefaultMQProducer defaultMQProducer = new
DefaultMQProducer("mao_group");
32         //nameserver地址
33         defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
34         //启动
35         defaultMQProducer.start();
36         //发送300条消息
37         for (int i = 0; i < 300; i++)
38         {
39             //创建消息对象, 参数分别是话题和消息体
40             Message message = new Message("test_topic", ("hello" +
i).getBytes(StandardCharsets.UTF_8));
41             //发送消息, 获取发送结果
42             SendResult sendResult = defaultMQProducer.send(message);
43             //直接打印发送结果
44             System.out.println(sendResult);
45             System.out.println();
46         }
47         //关闭
48         defaultMQProducer.shutdown();
49     }
50 }
51 }
52

```

启动后打印的日志

```

1  SendResult [sendStatus=SEND_OK, msgId=7F0000015F4C66D3C6170DFB29950000,
offsetMsgId=DF9C9F8D00002AA90000000000000000, messageQueue=MessageQueue
[topic=test_topic, brokerName=broker-b, queueId=1], queueOffset=0]
2
3  SendResult [sendStatus=SEND_OK, msgId=7F0000015F4C66D3C6170DFB2A0E0001,
offsetMsgId=DF9C9F8D00002AA90000000000000009E, messageQueue=MessageQueue
[topic=test_topic, brokerName=broker-b, queueId=2], queueOffset=0]
4
5  SendResult [sendStatus=SEND_OK, msgId=7F0000015F4C66D3C6170DFB2A1C0002,
offsetMsgId=DF9C9F8D00002AA90000000000000013C, messageQueue=MessageQueue
[topic=test_topic, brokerName=broker-b, queueId=3], queueOffset=0]
6
7  SendResult [sendStatus=SEND_OK, msgId=7F0000015F4C66D3C6170DFB2A2D0003,
offsetMsgId=DF9C9F8D00002A9F0000000000000000, messageQueue=MessageQueue
[topic=test_topic, brokerName=broker-a, queueId=0], queueOffset=0]
8
9  SendResult [sendStatus=SEND_OK, msgId=7F0000015F4C66D3C6170DFB2A780004,
offsetMsgId=DF9C9F8D00002A9F0000000000000009E, messageQueue=MessageQueue
[topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=0]
10 .....

```

RocketMq-console-ng

localhost:8080/#/cluster

RocketMq控制台

运维

驾驶舱

集群

主题

消费者

生产者

消息

更换语言

集群: rocketmq-cluster

分片	编号	地址	版本	生产消息TPS	消费消息TPS	昨日生产总数	昨日消费总数	今天生产总数	今天消费总数	操作
broker-b	0(master)	223.156.159.141:10921	V4_5_1	0.00	0.00	0	0	151	0	<div>状态配置</div>
broker-b	1(slave)	223.156.159.141:10941	V4_5_1	0.00	0.00	0	0	151	0	<div>状态配置</div>
broker-a	0(master)	223.156.159.141:10911	V4_5_1	0.00	0.00	0	0	149	0	<div>状态配置</div>
broker-a	1(slave)	223.156.159.141:10931	V4_5_1	0.00	0.00	0	0	149	0	<div>状态配置</div>

RocketMq-console-ng

localhost:8080/#/topic

RocketMq控制台

运维

驾驶舱

集群

主题

消费者

生产者

消息

更换语言

主题: ☒ 普通 ☐ 重试 ☐ 死信

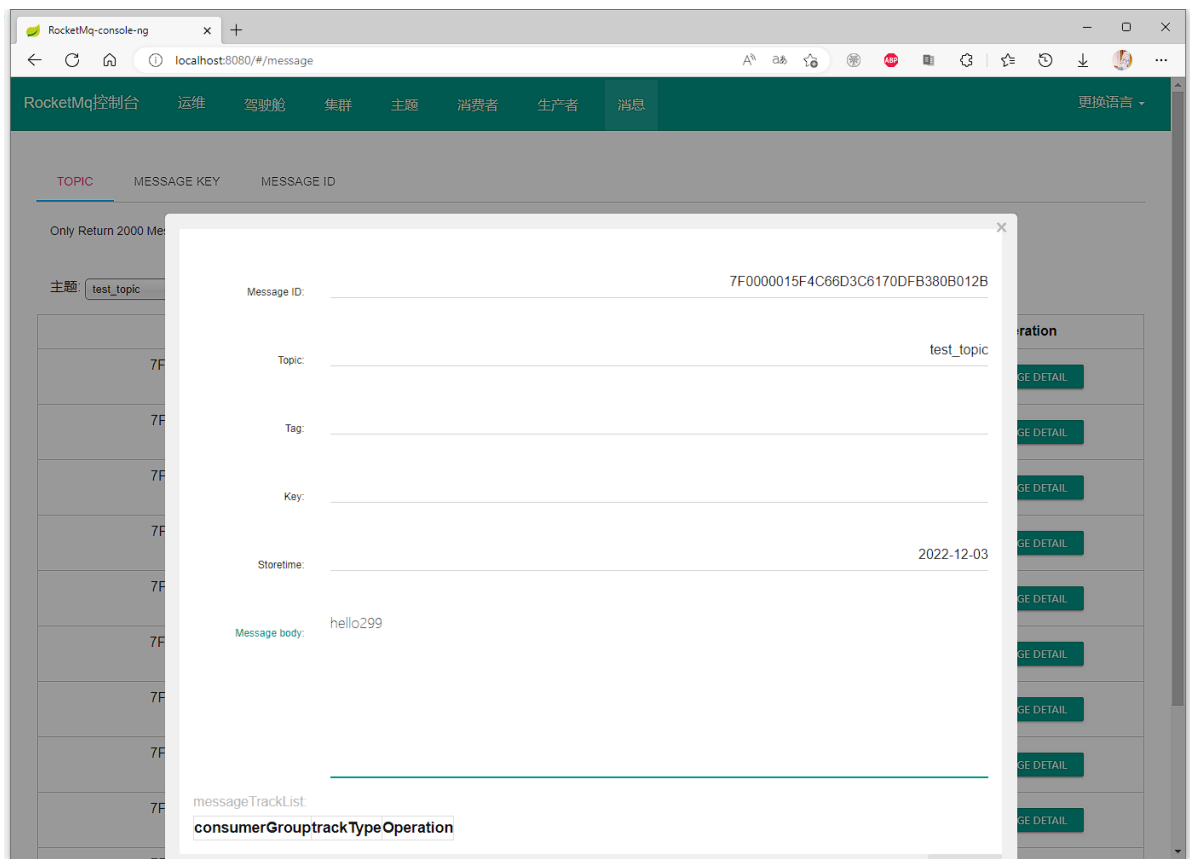
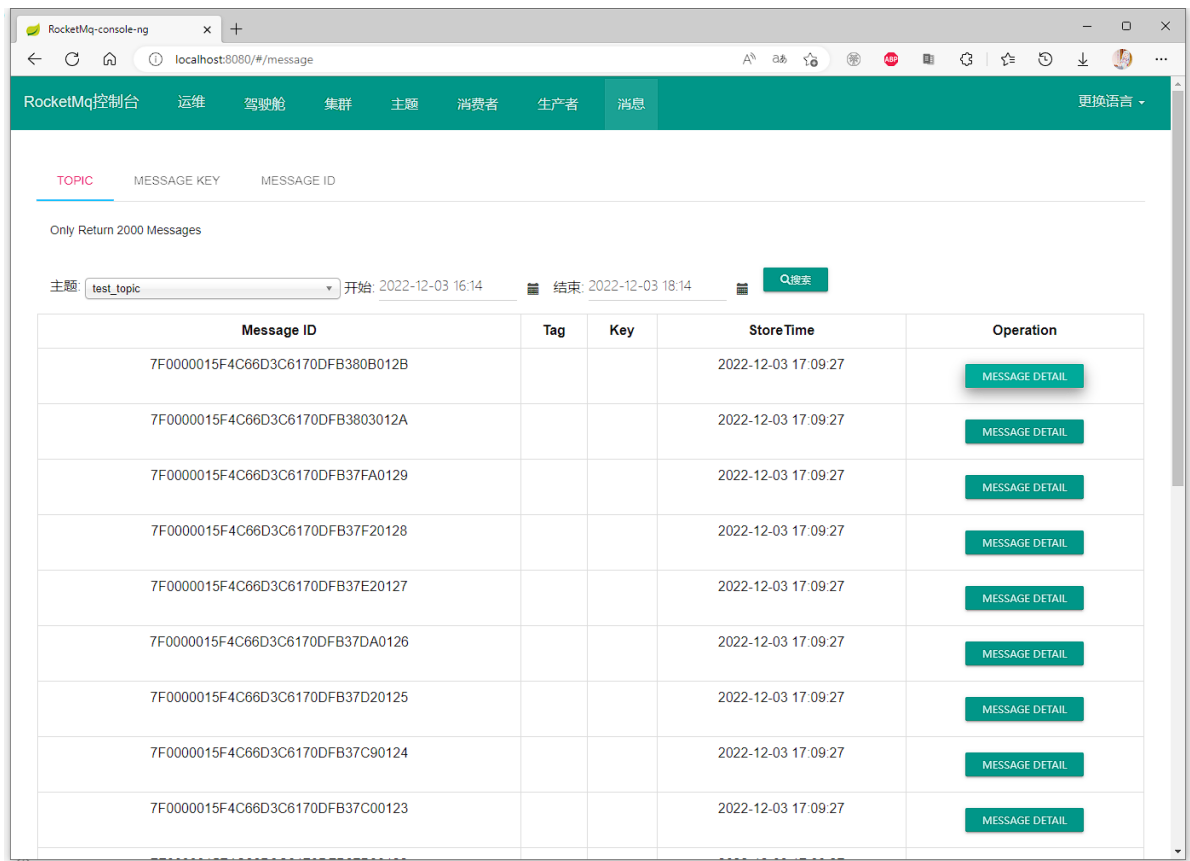
新增/更新

主题	操作
BenchmarkTest	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
OFFSET_MOVED_EVENT	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
RMQ_SYS_TRANS_HALF_TOPIC	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
SELF_TEST_TOPIC	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
TBW102	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
broker-a	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
broker-b	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
rocketmq-cluster	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>
test_topic	<div>状态路由CONSUMER 管理TOPIC 配置发送消息重置消费位点删除</div>

«

1

»



发送异步消息

异步消息通常用在响应时间敏感的业务场景，即发送端不能容忍长时间地等待Broker的响应

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.client.producer.SendCallback;
7 import org.apache.rocketmq.client.producer.SendResult;
8 import org.apache.rocketmq.common.message.Message;
9 import org.apache.rocketmq.remoting.exception.RemotingException;
10
11 import java.nio.charset.StandardCharsets;
12
13 /**
14  * Project name(项目名称): RocketMQ_基本消息的发送与接收
15  * Package(包名): mao.producer
16  * Class(类名): AsyncProducer
17  * Author(作者): mao
18  * Author QQ: 1296193245
19  * GitHub: https://github.com/maomao124/
20  * Date(创建日期): 2022/12/3
21  * Time(创建时间): 17:21
22  * Version(版本): 1.0
23  * Description(描述): 生产者-异步消息
24  */
25
26 public class AsyncProducer
27 {
28     public static void main(String[] args)
29         throws MQClientException, MQBrokerException, RemotingException,
30         InterruptedException
31     {
32         //生产者，参数为生产者组名
33         DefaultMQProducer defaultMQProducer = new
34         DefaultMQProducer("mao_group");
35         //nameserver地址
36         defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
37         //启动
38         defaultMQProducer.start();
39         //设置当异步消息发送失败时，消息重试的时间
40         defaultMQProducer.setRetryTimesWhenSendAsyncFailed(0);
41         //发送300条异步消息
42         for (int i = 300; i < 600; i++)
43         {
44             //创建消息对象，参数分别是主题和消息体
45             Message message = new Message("test_topic", ("hello" +
46             i).getBytes(StandardCharsets.UTF_8));
47             //发送消息，获取发送结果
48             defaultMQProducer.send(message, new SendCallback()
49             {
50                 /**
51                  * 成功
52                  */
53             })
54         }
55     }
56 }
```

```

50         * @param sendResult 发送结果
51         */
52         @Override
53         public void onSuccess(SendResult sendResult)
54         {
55             //直接打印发送结果
56             System.out.println(sendResult);
57             System.out.println();
58         }
59
60         /**
61         * 异常
62         *
63         * @param throwable throwable
64         */
65         @Override
66         public void onException(Throwable throwable)
67         {
68             //打印异常
69             throwable.printStackTrace();
70         }
71     });
72
73 }
74 }
75 }
76

```

可能会出现异常

MQBrokerException: CODE: 2 DESC: [TIMEOUT_CLEAN_QUEUE]broker busy, start flow control for a while, period in queue: 208ms, size of queue: 130

单向发送消息

这种方式主要用在不特别关心发送结果的场景，例如日志发送

```

1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.client.producer.SendResult;
7 import org.apache.rocketmq.common.message.Message;
8 import org.apache.rocketmq.remoting.exception.RemotingException;
9

```

```

10 import java.nio.charset.StandardCharsets;
11
12 /**
13  * Project name(项目名称): RocketMQ_基本消息的发送与接收
14  * Package(包名): mao.producer
15  * Class(类名): OnewayProducer
16  * Author(作者): mao
17  * Author QQ: 1296193245
18  * GitHub: https://github.com/maomao124/
19  * Date(创建日期): 2022/12/3
20  * Time(创建时间): 17:33
21  * Version(版本): 1.0
22  * Description(描述): 生产者-单向消息
23  */
24
25 public class OnewayProducer
26 {
27     public static void main(String[] args)
28         throws MQClientException, MQBrokerException, RemotingException,
29         InterruptedException
30     {
31         //生产者, 参数为生产者组名
32         DefaultMQProducer defaultMQProducer = new
33         DefaultMQProducer("mao_group");
34         //nameserver地址
35         defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
36         //启动
37         defaultMQProducer.start();
38         //发送300条消息
39         for (int i = 600; i < 900; i++)
40         {
41             //创建消息对象, 参数分别是话题和消息体
42             Message message = new Message("test_topic", ("hello" +
43             i).getBytes(StandardCharsets.UTF_8));
44             //发送单向消息
45             defaultMQProducer.sendOneway(message);
46             System.out.println(i);
47         }
48         //关闭
49         defaultMQProducer.shutdown();
50     }
51 }

```

消费消息

负载均衡模式

消费者采用负载均衡方式消费消息，多个消费者共同消费队列消息，每个消费者处理的消息不同

```
1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import
5 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6 import
7 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
8 import
9 org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
10
11 import org.apache.rocketmq.client.exception.MQClientException;
12 import org.apache.rocketmq.common.message.MessageExt;
13 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
14
15 import java.util.List;
16
17 /**
18  * Project name(项目名称): RocketMQ_基本消息的发送与接收
19  * Package(包名): mao.consumer
20  * Class(类名): LoadBalanceConsumer
21  * Author(作者): mao
22  * Author QQ: 1296193245
23  * Github: https://github.com/maomao124/
24  * Date(创建日期): 2022/12/3
25  * Time(创建时间): 19:10
26  * Version(版本): 1.0
27  * Description(描述): 消费者-负载均衡模式
28  */
29
30 public class LoadBalanceConsumer
31 {
32     public static void main(String[] args) throws MQClientException
33     {
34         //创建消费者
35         DefaultMQPushConsumer defaultMQPushConsumer = new
36         DefaultMQPushConsumer("mao_group");
37         //设置nameserver地址
38         defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
39         //订阅topic
40         defaultMQPushConsumer.subscribe("test_topic", "*");
41         //设置消费模式-负载均衡
42         defaultMQPushConsumer.setMessageModel(MessageModel.CLUSTERING);
43         //注册监听器，处理消息
44         defaultMQPushConsumer.registerMessageListener(new
45         MessageListenerConcurrently()
46         {
47             @Override
48             public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
49             list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
50             {
51                 //...
52             }
53         })
54     }
55 }
```



```
44         //打印
45         System.out.println("线程" + Thread.currentThread().getName()
+ "消费消息: " + list);
46         //返回成功的状态
47         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
48     }
49 });
50 //启动消费者
51 defaultMQPushConsumer.start();
52 }
53 }
```

启动后的日志打印

- 1 线程ConsumeMessageThread_mao_group_5消费消息: [MessageExt [brokerName=broker-b, queueId=1, storeSize=159, queueOffset=2, sysFlag=0, bornTimestamp=1670058564408, bornHost=/223.156.159.141:62262, storeTimestamp=1670058564409, storeHost=/223.156.159.141:10921, msgId=DF9C9F8D00002AA9000000000000004F2, commitLogOffset=1266, bodyCRC=441618388, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0, MAX_OFFSET=54, CONSUME_START_TIME=1670066327850, UNIQ_KEY=7F0000015F4C66D3C6170DFB2B380010, WAIT=true}, body=[104, 101, 108, 108, 111, 49, 54], transactionId='null'}]]]
- 2 线程ConsumeMessageThread_mao_group_10消费消息: [MessageExt [brokerName=broker-a, queueId=0, storeSize=159, queueOffset=4, sysFlag=0, bornTimestamp=1670058564583, bornHost=/223.156.159.141:62263, storeTimestamp=1670058564583, storeHost=/223.156.159.141:10911, msgId=DF9C9F8D00002A9F000000000000009EC, commitLogOffset=2540, bodyCRC=829275372, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0, MAX_OFFSET=53, CONSUME_START_TIME=1670066327849, UNIQ_KEY=7F0000015F4C66D3C6170DFB2B8E70023, WAIT=true}, body=[104, 101, 108, 108, 111, 51, 53], transactionId='null'}]]]
- 3 线程ConsumeMessageThread_mao_group_7消费消息: [MessageExt [brokerName=broker-b, queueId=1, storeSize=159, queueOffset=3, sysFlag=0, bornTimestamp=1670058564483, bornHost=/223.156.159.141:62262, storeTimestamp=1670058564483, storeHost=/223.156.159.141:10921, msgId=DF9C9F8D00002AA90000000000000076E, commitLogOffset=1902, bodyCRC=1601289531, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0, MAX_OFFSET=54, CONSUME_START_TIME=1670066327849, UNIQ_KEY=7F0000015F4C66D3C6170DFB2B830018, WAIT=true}, body=[104, 101, 108, 108, 111, 50, 52], transactionId='null'}]]]
- 4 线程ConsumeMessageThread_mao_group_3消费消息: [MessageExt [brokerName=broker-b, queueId=1, storeSize=158, queueOffset=1, sysFlag=0, bornTimestamp=1670058564316, bornHost=/223.156.159.141:62262, storeTimestamp=1670058564317, storeHost=/223.156.159.141:10921, msgId=DF9C9F8D00002AA900000000000000278, commitLogOffset=632, bodyCRC=2129486240, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0, MAX_OFFSET=54, CONSUME_START_TIME=1670066327849, UNIQ_KEY=7F0000015F4C66D3C6170DFB2ADC0008, WAIT=true}, body=[104, 101, 108, 108, 111, 56], transactionId='null'}]]]
- 5 线程ConsumeMessageThread_mao_group_9消费消息: [MessageExt [brokerName=broker-b, queueId=1, storeSize=159, queueOffset=4, sysFlag=0, bornTimestamp=1670058564558, bornHost=/223.156.159.141:62262, storeTimestamp=1670058564558, storeHost=/223.156.159.141:10921, msgId=DF9C9F8D00002AA9000000000000009EA, commitLogOffset=2538, bodyCRC=789129551, reconsumeTimes=0, preparedTransactionOffset=0, toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0, MAX_OFFSET=54, CONSUME_START_TIME=1670066327849, UNIQ_KEY=7F0000015F4C66D3C6170DFB2BCE0020, WAIT=true}, body=[104, 101, 108, 108, 111, 51, 50], transactionId='null'}]]]

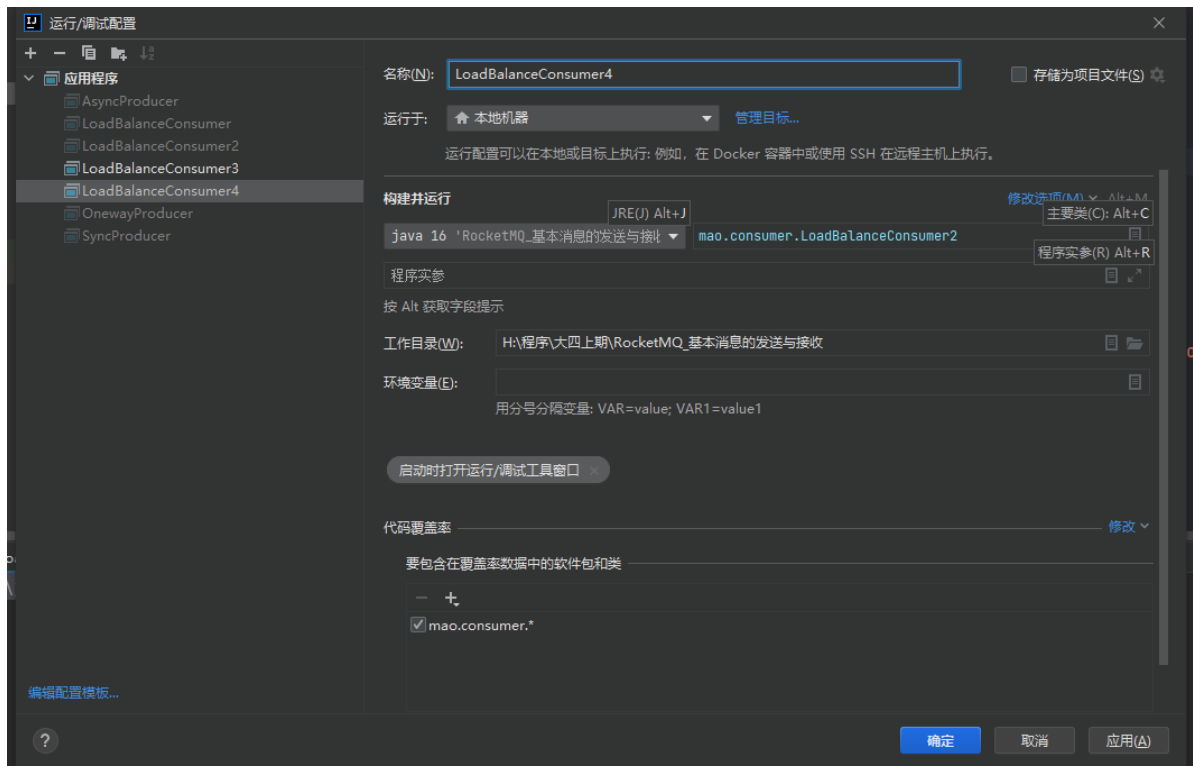
```
6 线程ConsumeMessageThread_mao_group_20消费消息: [MessageExt [brokerName=broker-
b, queueId=1, storeSize=159, queueOffset=9, sysFlag=0,
bornTimestamp=1670058565099, bornHost=/223.156.159.141:62262,
storeTimestamp=1670058565100, storeHost=/223.156.159.141:10921,
msgId=DF9C9F8D00002AA90000000000001656, commitLogOffset=5718,
bodyCRC=1264970827, reconsumeTimes=0, preparedTransactionOffset=0,
toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0,
MAX_OFFSET=54, CONSUME_START_TIME=1670066327853,
UNIQ_KEY=7F0000015F4C66D3C6170DFB2DEB0048, WAIT=true}, body=[104, 101, 108,
108, 111, 55, 50], transactionId='null'}]]

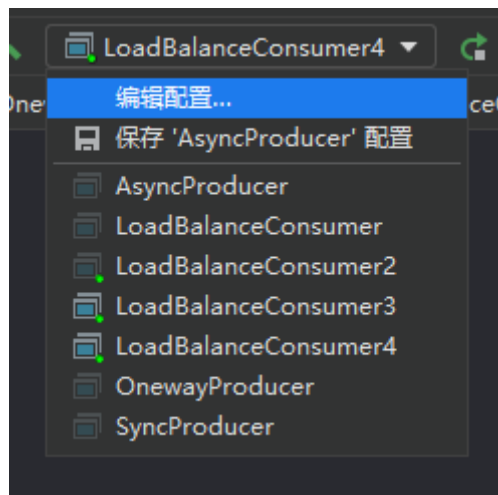
7 线程ConsumeMessageThread_mao_group_17消费消息: [MessageExt [brokerName=broker-
b, queueId=1, storeSize=159, queueOffset=8, sysFlag=0,
bornTimestamp=1670058565008, bornHost=/223.156.159.141:62262,
storeTimestamp=1670058565009, storeHost=/223.156.159.141:10921,
msgId=DF9C9F8D00002AA900000000000013DA, commitLogOffset=5082,
bodyCRC=991787071, reconsumeTimes=0, preparedTransactionOffset=0,
toString()=Message{topic='test_topic', flag=0, properties={MIN_OFFSET=0,
MAX_OFFSET=54, CONSUME_START_TIME=1670066327852,
UNIQ_KEY=7F0000015F4C66D3C6170DFB2D900040, WAIT=true}, body=[104, 101, 108,
108, 111, 54, 52], transactionId='null'}]]

8 .....
```

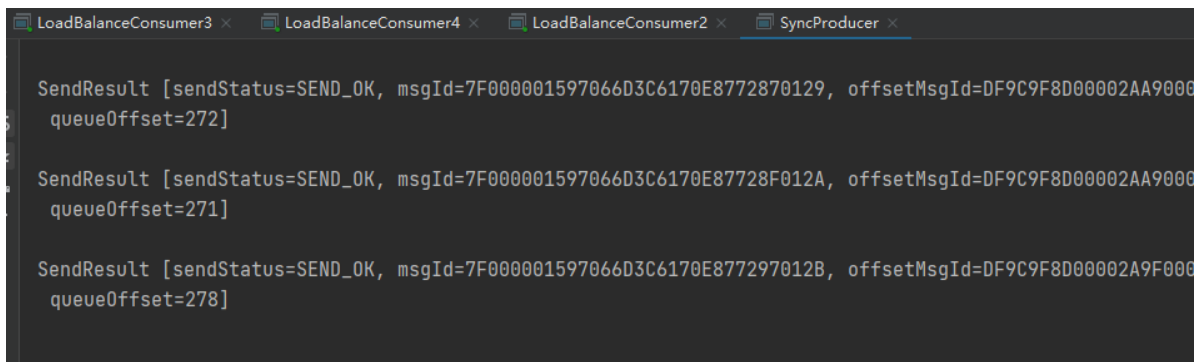
验证负载均衡模式

启动3个消费者





启动生产者

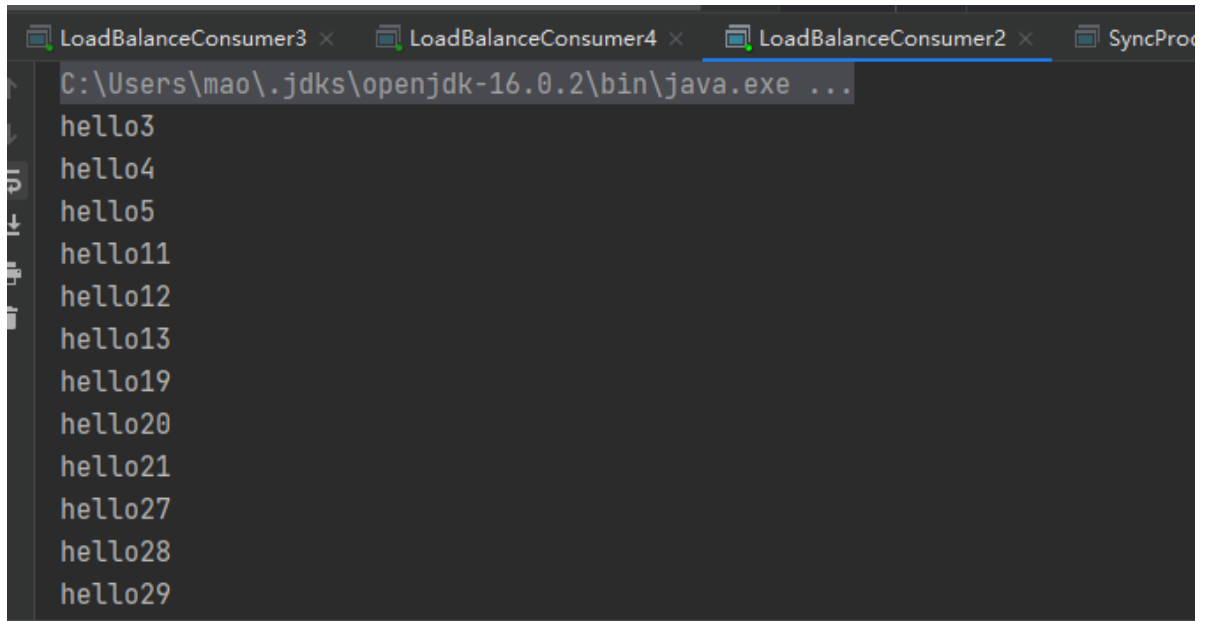


观察3个消费者消费的消息

第一个消费者:

1	hello3
2	hello4
3	hello5
4	hello11
5	hello12
6	hello13
7	hello19
8	hello20
9	hello21
10	hello27
11	hello28
12	hello29
13	hello35
14	hello36
15	hello37
16	hello43
17	hello44
18	hello45
19	hello51

```
20 | hello52
21 | hello53
22 | hello59
23 | .....
```



```
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe ...
hello3
hello4
hello5
hello11
hello12
hello13
hello19
hello20
hello21
hello27
hello28
hello29
```

第二个消费者:

```
1 | hello0
2 | hello6
3 | hello7
4 | hello8
5 | hello14
6 | hello15
7 | hello16
8 | hello22
9 | hello23
10 | hello24
11 | hello30
12 | hello31
13 | hello32
14 | hello38
15 | hello39
16 | hello40
17 | hello46
18 | hello47
19 | hello48
20 | hello54
21 | .....
```

```
LoadBalanceConsumer3 x LoadBalanceConsumer4 x LoadBalanceConsumer2 x SyncF
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe ...
hello0
hello6
hello7
hello8
hello14
hello15
hello16
hello22
hello23
hello24
hello30
hello31
```

第三个消费者:

```
1 hello1
2 hello2
3 hello9
4 hello10
5 hello17
6 hello18
7 hello25
8 hello26
9 hello33
10 hello34
11 hello41
12 hello42
13 hello49
14 hello50
15 hello57
16 hello58
17 hello65
18 hello66
19 hello73
20 hello74
21 hello81
22 hello82
23 .....
```

```
LoadBalanceConsumer3 x LoadBalanceConsumer4 x LoadBalanceConsumer2 x SyncProduce
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe ...
hello1
hello2
hello9
hello10
hello17
hello18
hello25
hello26
hello33
hello34
hello41
hello42
```

负载均衡模式下，消息被消费一次

广播模式

消费者采用广播的方式消费消息，每个消费者消费的消息都是相同的

```
1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import
5 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6 import
7 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
8 import
9 org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
10
11 import org.apache.rocketmq.client.exception.MQClientException;
12 import org.apache.rocketmq.common.message.MessageExt;
13 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
14
15 import java.util.List;
16
17 /**
18  * Project name(项目名称): RocketMQ_基本消息的发送与接收
19  * Package(包名): mao.consumer
20  * Class(类名): BroadcastConsumer
21  * Author(作者): mao
22  * Author QQ: 1296193245
23  * Github: https://github.com/maomao124/
24  * Date(创建日期): 2022/12/4
25  * Time(创建时间): 15:25
26  * Version(版本): 1.0
27  */
```

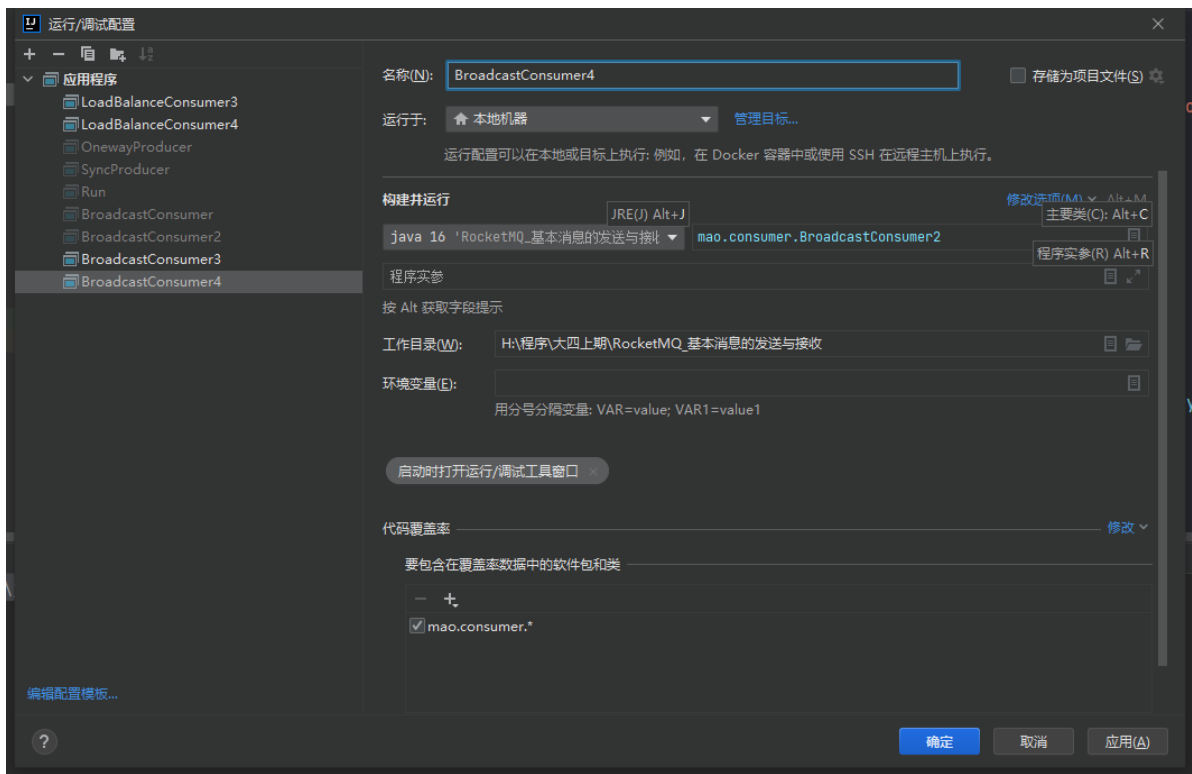
```

23  * Description(描述): 消费者-广播模式
24  */
25
26  public class BroadcastConsumer
27  {
28      public static void main(String[] args) throws MQClientException
29      {
30          //创建消费者
31          DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
32          //设置nameserver地址
33          defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
34          //订阅topic
35          defaultMQPushConsumer.subscribe("test_topic", "*");
36          //设置消费模式-负载均衡
37          defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
38          //注册监听器, 处理消息
39          defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
40          {
41              @Override
42              public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
43              {
44                  //打印
45                  System.out.println("线程" + Thread.currentThread().getName()
+ "消费消息: " + list);
46                  //返回成功的状态
47                  return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
48              }
49          });
50          //启动消费者
51          defaultMQPushConsumer.start();
52      }
53  }

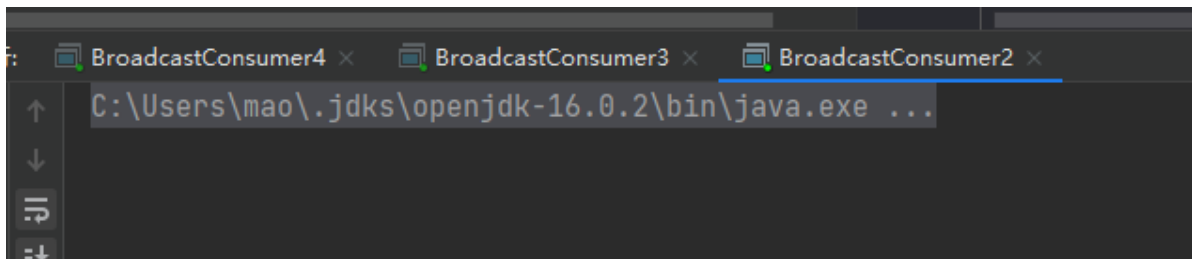
```

测试广播模式

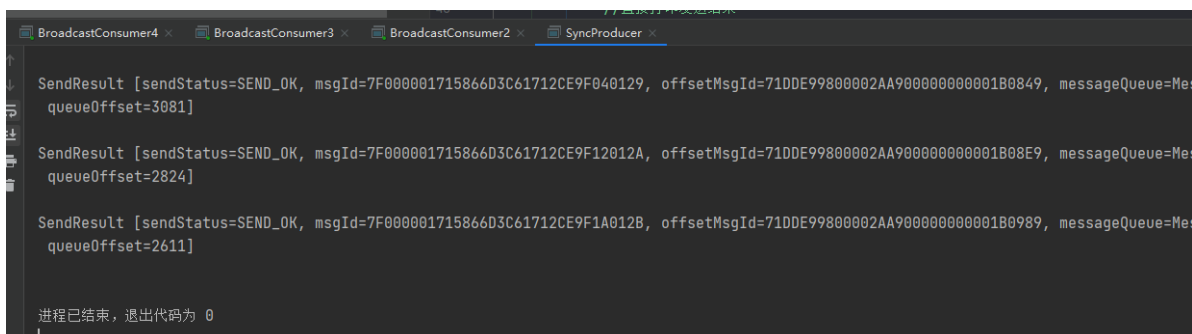
复制配置



启动消费者



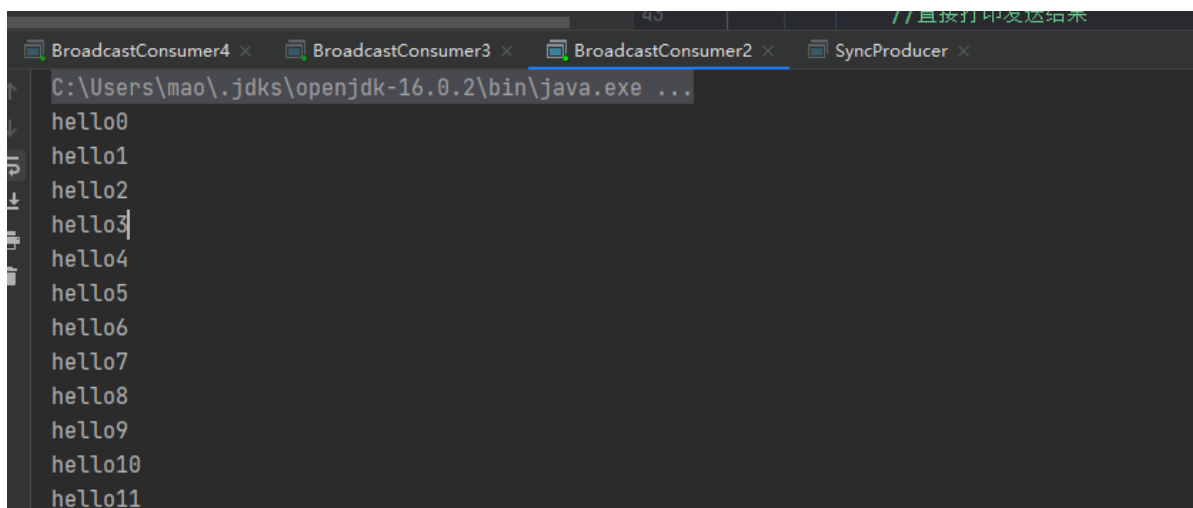
启动生产者



第一个消费者

1	hello0
2	hello1

```
3 | hello2
4 | hello3
5 | hello4
6 | hello5
7 | hello6
8 | hello7
9 | hello8
10 | hello9
11 | hello10
12 | hello11
13 | hello12
14 | hello13
15 | hello14
16 | hello15
17 | hello16
18 | hello17
19 | hello18
20 | hello19
21 | hello20
22 | hello21
23 | hello22
24 | hello23
25 | hello24
26 | hello25
27 | hello26
28 | hello27
29 | hello28
30 | .....
```

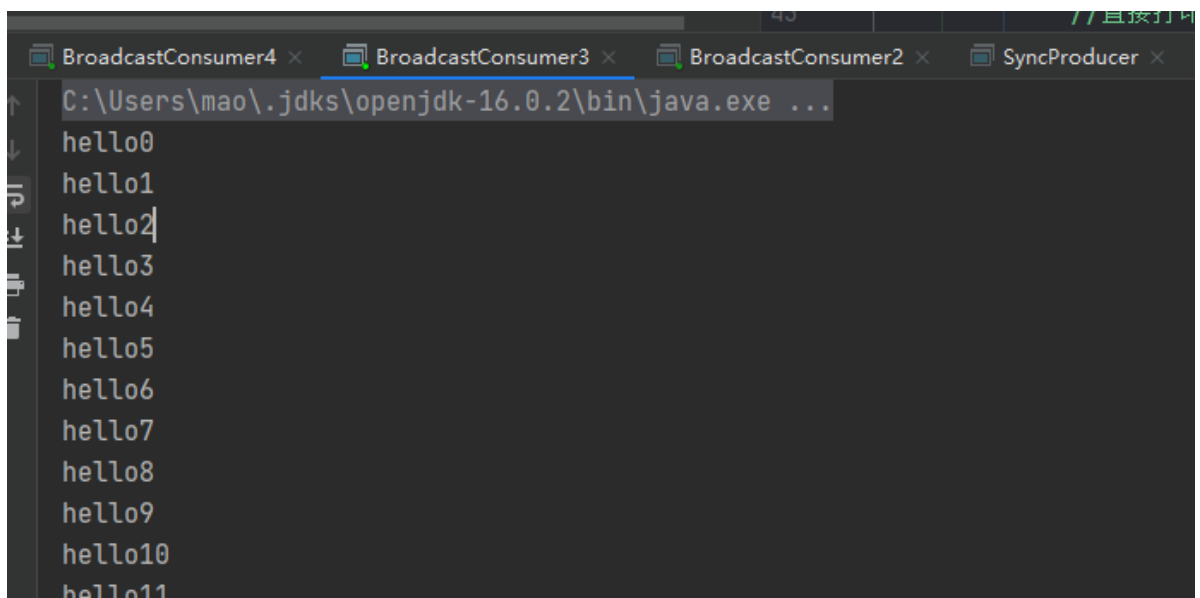


```
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe ...
hello0
hello1
hello2
hello3
hello4
hello5
hello6
hello7
hello8
hello9
hello10
hello11
```

第二个消费者

```
1 | hello0
2 | hello1
3 | hello2
4 | hello3
5 | hello4
6 | hello5
```

```
7 | hello6
8 | hello7
9 | hello8
10 | hello9
11 | hello10
12 | hello11
13 | hello12
14 | hello13
15 | hello14
16 | hello15
17 | hello16
18 | hello17
19 | hello18
20 | hello19
21 | hello20
22 | hello21
23 | hello22
24 | hello23
25 | hello24
26 | .....
```

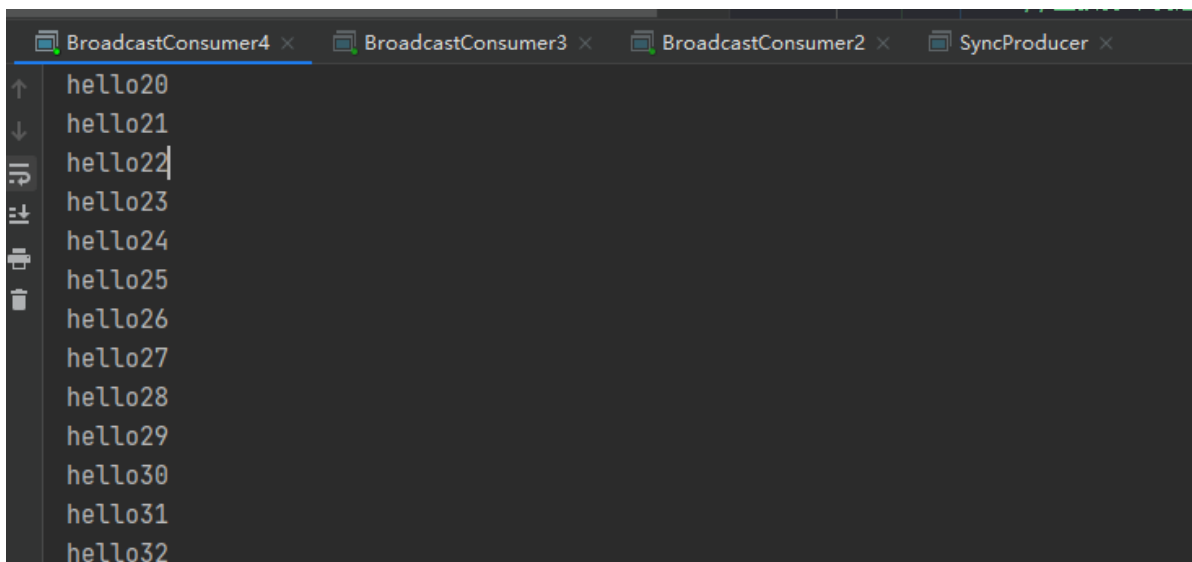


```
C:\Users\mao\.jdk\openjdk-16.0.2\bin\java.exe ...
hello0
hello1
hello2
hello3
hello4
hello5
hello6
hello7
hello8
hello9
hello10
hello11
```

第三个消费者

```
1 | hello0
2 | hello1
3 | hello2
4 | hello3
5 | hello4
6 | hello5
7 | hello6
8 | hello7
9 | hello8
10 | hello9
11 | hello10
12 | hello11
```

```
13 hello12
14 hello13
15 hello14
16 hello15
17 hello16
18 hello17
19 hello18
20 hello19
21 hello20
22 hello21
23 hello22
24 hello23
25 hello24
26 hello25
27 hello26
28 hello27
29 hello28
30 hello29
31 .....
```



消息会被每一个消费者消费

顺序消息

消息有序指的是可以按照消息的发送顺序来消费(FIFO)。RocketMQ可以严格的保证消息有序，可以分为分区有序或者全局有序。

顺序消费的原理解析，在默认的情况下消息发送会采取Round Robin轮询方式把消息发送到不同的queue(分区队列)；而消费消息的时候从多个queue上拉取消息，这种情况发送和消费是不能保证顺序。但是如果控制发送的顺序消息只依次发送到同一个queue中，消费的时候只从这个queue上依次拉取，则就保证了顺序。当发送和消费参与的queue只有一个，则是全局有序；如果多个queue参与，则为分区有序，即相对每个queue，消息都是有序的。

生产

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.client.producer.MessageQueueSelector;
7 import org.apache.rocketmq.common.message.Message;
8 import org.apache.rocketmq.common.message.MessageQueue;
9 import org.apache.rocketmq.remoting.exception.RemotingException;
10
11 import java.nio.charset.StandardCharsets;
12 import java.text.SimpleDateFormat;
13 import java.util.Date;
14 import java.util.List;
15
16 /**
17  * Project name(项目名称): RocketMQ_顺序消息的发送与接收
18  * Package(包名): mao.producer
19  * Class(类名): SequentialProducer
20  * Author(作者): mao
21  * Author QQ: 1296193245
22  * GitHub: https://github.com/maomao124/
23  * Date(创建日期): 2022/12/5
24  * Time(创建时间): 13:34
25  * Version(版本): 1.0
26  * Description(描述): 顺序消息-生产者
27  */
28
29 public class SequentialProducer
30 {
31     /**
32      * 简单日期格式
33      */
34     static SimpleDateFormat simpleDateFormat = new
35         SimpleDateFormat("HH:mm:ss");
36
37     public static void main(String[] args)
38         throws MQClientException, MQBrokerException, RemotingException,
39         InterruptedException
40     {
41         //生产者
42         DefaultMQProducer defaultMQProducer = new
43             DefaultMQProducer("mao_group");
```

```

41 //设置nameserver地址
42 defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
43 //启动生产者
44 defaultMQProducer.start();
45 //发送300条顺序消息
46 for (int i = 0; i < 300; i++)
47 {
48     //消息体
49     String messageBody = SimpleDateFormat.format(new Date()) + " -->
消息" + i;
50     //消息对象
51     Message message = new Message("test_topic",
messageBody.getBytes(StandardCharsets.UTF_8));
52     defaultMQProducer.send(message, new MessageQueueSelector()
53     {
54         /**
55          * 选择发送的队列
56          *
57          * @param list    列表
58          * @param message 消息
59          * @param o        o
60          * @return {@link MessageQueue}
61          */
62         @Override
63         public MessageQueue select(List<MessageQueue> list, Message
message, Object o)
64         {
65             System.out.println(messageBody + " 发送到0号队列，队列总
数: " + list.size());
66             return list.get(0);
67         }
68     }, i);
69 }
70 //关闭
71 defaultMQProducer.shutdown();
72 }
73 }

```

```

1 13:54:23 --> 消息0 发送到0号队列，队列总数: 8
2 13:54:23 --> 消息1 发送到0号队列，队列总数: 8
3 13:54:23 --> 消息2 发送到0号队列，队列总数: 8
4 13:54:23 --> 消息3 发送到0号队列，队列总数: 8
5 13:54:23 --> 消息4 发送到0号队列，队列总数: 8
6 13:54:23 --> 消息5 发送到0号队列，队列总数: 8
7 13:54:23 --> 消息6 发送到0号队列，队列总数: 8
8 13:54:23 --> 消息7 发送到0号队列，队列总数: 8
9 13:54:23 --> 消息8 发送到0号队列，队列总数: 8
10 13:54:23 --> 消息9 发送到0号队列，队列总数: 8
11 13:54:23 --> 消息10 发送到0号队列，队列总数: 8
12 13:54:23 --> 消息11 发送到0号队列，队列总数: 8
13 13:54:23 --> 消息12 发送到0号队列，队列总数: 8
14 13:54:23 --> 消息13 发送到0号队列，队列总数: 8
15 13:54:23 --> 消息14 发送到0号队列，队列总数: 8

```

```

16 13:54:23 --> 消息15 发送到0号队列，队列总数：8
17 13:54:23 --> 消息16 发送到0号队列，队列总数：8
18 13:54:23 --> 消息17 发送到0号队列，队列总数：8
19 13:54:23 --> 消息18 发送到0号队列，队列总数：8
20 13:54:23 --> 消息19 发送到0号队列，队列总数：8
21 13:54:23 --> 消息20 发送到0号队列，队列总数：8
22 13:54:23 --> 消息21 发送到0号队列，队列总数：8
23 13:54:23 --> 消息22 发送到0号队列，队列总数：8
24 13:54:23 --> 消息23 发送到0号队列，队列总数：8
25 13:54:23 --> 消息24 发送到0号队列，队列总数：8
26 13:54:23 --> 消息25 发送到0号队列，队列总数：8
27 13:54:23 --> 消息26 发送到0号队列，队列总数：8
28 13:54:23 --> 消息27 发送到0号队列，队列总数：8
29 13:54:23 --> 消息28 发送到0号队列，队列总数：8
30 13:54:23 --> 消息29 发送到0号队列，队列总数：8
31 13:54:23 --> 消息30 发送到0号队列，队列总数：8
32 13:54:23 --> 消息31 发送到0号队列，队列总数：8
33 13:54:23 --> 消息32 发送到0号队列，队列总数：8
34 13:54:23 --> 消息33 发送到0号队列，队列总数：8
35 .....

```

消费

```

1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import org.apache.rocketmq.client.consumer.listener.ConsumeOrderlyContext;
5 import org.apache.rocketmq.client.consumer.listener.ConsumeOrderlyStatus;
6 import org.apache.rocketmq.client.consumer.listener.MessageListenerOrderly;
7 import org.apache.rocketmq.client.exception.MQClientException;
8 import org.apache.rocketmq.common.message.MessageExt;
9 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
10
11 import java.nio.charset.StandardCharsets;
12 import java.util.List;
13
14 /**
15  * Project name(项目名称): RocketMQ_顺序消息的发送与接收
16  * Package(包名): mao.consumer
17  * Class(类名): SequentialConsumer
18  * Author(作者): mao
19  * Author QQ: 1296193245
20  * GitHub: https://github.com/maomao124/
21  * Date(创建日期): 2022/12/5
22  * Time(创建时间): 13:57
23  * Version(版本): 1.0
24  * Description(描述): 消费者-顺序消息
25  */
26
27 public class SequentialConsumer

```

```

28 {
29     /**
30      * 得到int随机
31      *
32      * @param min 最小值
33      * @param max 最大值
34      * @return int
35      */
36     public static int getIntRandom(int min, int max)
37     {
38         if (min > max)
39         {
40             min = max;
41         }
42         return min + (int) (Math.random() * (max - min + 1));
43     }
44
45     public static void main(String[] args) throws MQClientException
46     {
47         //消费者
48         DefaultMQPushConsumer defaultMQPushConsumer = new
49         DefaultMQPushConsumer("mao_group");
50         //设置nameserver地址
51         defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
52         //设置消费模式-负载均衡
53         defaultMQPushConsumer.setMessageModel(MessageModel.CLUSTERING);
54         //设置消费的主题
55         defaultMQPushConsumer.subscribe("test_topic", "*");
56         //注册监听器
57         defaultMQPushConsumer.registerMessageListener(new
58         MessageListenerOrderly()
59         {
60             @Override
61             public ConsumeOrderlyStatus consumeMessage(List<MessageExt>
62             list, ConsumeOrderlyContext consumeOrderlyContext)
63             {
64                 //线程
65                 Thread currentThread = Thread.currentThread();
66                 //打印，每个queue有唯一的consume线程来消费，订单对每个queue(分区)有
67                 序
68                 for (MessageExt messageExt : list)
69                 {
70                     System.out.println("当前线程: " + currentThread.getName()
71                     + " " + "队列id: " + messageExt.getQueueId()
72                     + " " + "消息: " + new
73                     String(messageExt.getBody(),
74                     StandardCharsets.UTF_8));
75                     try
76                     {
77                         {
78                             Thread.sleep(getIntRandom(100, 1000));
79                         }
80                     }
81                     catch (InterruptedException e)
82                     {
83                         {
84                             e.printStackTrace();
85                         }
86                     }
87                 }
88                 //返回成功
89                 return ConsumeOrderlyStatus.SUCCESS;
90             }
91         });
92     }
93 }

```



```

80         }
81     });
82     //启动
83     defaultMQPushConsumer.start();
84 }
85 }

```

```

1 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:55 -->
   消息0
2 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息1
3 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息2
4 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息3
5 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息4
6 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息5
7 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息6
8 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息7
9 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息8
10 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息9
11 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息10
12 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息11
13 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息12
14 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息13
15 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息14
16 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息15
17 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息16
18 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息17
19 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息18
20 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息19
21 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息20
22 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息21
23 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息22
24 当前线程: ConsumeMessageThread_mao_group_1  队列id: 0      消息: 14:14:56 -->
   消息23

```

25	当前线程: ConsumeMessageThread_mao_group_1 消息24	队列id: 0	消息: 14:14:56 -->
26	当前线程: ConsumeMessageThread_mao_group_1 消息25	队列id: 0	消息: 14:14:56 -->
27	当前线程: ConsumeMessageThread_mao_group_1 消息26	队列id: 0	消息: 14:14:56 -->
28	当前线程: ConsumeMessageThread_mao_group_1 消息27	队列id: 0	消息: 14:14:56 -->
29	当前线程: ConsumeMessageThread_mao_group_1 消息28	队列id: 0	消息: 14:14:56 -->
30	当前线程: ConsumeMessageThread_mao_group_1 消息29	队列id: 0	消息: 14:14:56 -->
31	当前线程: ConsumeMessageThread_mao_group_1 消息30	队列id: 0	消息: 14:14:56 -->
32	当前线程: ConsumeMessageThread_mao_group_1 消息31	队列id: 0	消息: 14:14:56 -->
33	当前线程: ConsumeMessageThread_mao_group_1 消息32	队列id: 0	消息: 14:14:56 -->
34	当前线程: ConsumeMessageThread_mao_group_1 消息33	队列id: 0	消息: 14:14:56 -->
35	当前线程: ConsumeMessageThread_mao_group_1 消息34	队列id: 0	消息: 14:14:56 -->
36	当前线程: ConsumeMessageThread_mao_group_1 消息35	队列id: 0	消息: 14:14:56 -->
37	当前线程: ConsumeMessageThread_mao_group_1 消息36	队列id: 0	消息: 14:14:56 -->
38	当前线程: ConsumeMessageThread_mao_group_1 消息37	队列id: 0	消息: 14:14:56 -->
39	当前线程: ConsumeMessageThread_mao_group_1 消息38	队列id: 0	消息: 14:14:56 -->
40	当前线程: ConsumeMessageThread_mao_group_1 消息39	队列id: 0	消息: 14:14:56 -->
41	当前线程: ConsumeMessageThread_mao_group_1 消息40	队列id: 0	消息: 14:14:56 -->
42	当前线程: ConsumeMessageThread_mao_group_1 消息41	队列id: 0	消息: 14:14:56 -->
43	当前线程: ConsumeMessageThread_mao_group_1 消息42	队列id: 0	消息: 14:14:56 -->
44	当前线程: ConsumeMessageThread_mao_group_1 消息43	队列id: 0	消息: 14:14:56 -->
45		

延时消息

比如电商里，提交了一个订单就可以发送一个延时消息，1h后去检查这个订单的状态，如果还是未付款就取消订单释放库存。

使用有限制

```
1 // org/apache/rocketmq/store/config/MessageStoreConfig.java
2 private String messageDelayLevel = "1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m
  10m 20m 30m 1h 2h";
```

现在RocketMq并不支持任意时间的延时，需要设置几个固定的延时等级，从1s到2h分别对应着等级1到18

生产者

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.common.message.Message;
7 import org.apache.rocketmq.remoting.exception.RemotingException;
8
9 import java.nio.charset.StandardCharsets;
10 import java.text.SimpleDateFormat;
11 import java.util.Date;
12
13 /**
14  * Project name(项目名称): RocketMQ_延迟消息的发送与接收
15  * Package(包名): mao.producer
16  * Class(类名): DelayProducer
17  * Author(作者): mao
18  * Author QQ: 1296193245
19  * GitHub: https://github.com/maomao124/
20  * Date(创建日期): 2022/12/5
21  * Time(创建时间): 20:52
22  * Version(版本): 1.0
23  * Description(描述): 生产者-延迟消息
24  */
25
26 public class DelayProducer
27 {
28     /**
29      * 简单日期格式
30      */
31     static SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("HH:mm:ss");
32
33     public static void main(String[] args)
34     throws MQClientException, MQBrokerException, RemotingException,
InterruptedException
35     {
```

```

36     //生产者
37     DefaultMQProducer defaultMQProducer = new
DefaultMQProducer("mao_group");
38     //设置nameserver地址
39     defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
40     //启动
41     defaultMQProducer.start();
42     //发送100条延迟消息
43     for (int i = 0; i < 100; i++)
44     {
45         //时间
46         String format = SimpleDateFormat.format(new Date());
47         //消息对象
48         Message message = new Message("test_topic", (format + " --> 消息"
+ i).getBytes(StandardCharsets.UTF_8));
49         //设置延时，水平4为30秒
50         message.setDelayTimeLevel(4);
51         //发送消息
52         defaultMQProducer.send(message);
53     }
54     //关闭
55     defaultMQProducer.shutdown();
56 }
57 }

```

消费者

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
6  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
7  import org.apache.rocketmq.client.exception.MQClientException;
8  import org.apache.rocketmq.common.message.MessageExt;
9
10 import java.nio.charset.StandardCharsets;
11 import java.text.SimpleDateFormat;
12 import java.util.Date;
13 import java.util.List;
14
15 /**
16  * Project name(项目名称): RocketMQ_延迟消息的发送与接收
17  * Package(包名): mao.consumer
18  * Class(类名): DelayConsumer
19  * Author(作者): mao
20  * Author QQ: 1296193245
21  * GitHub: https://github.com/maomao124/

```

```

22  * Date(创建日期): 2022/12/5
23  * Time(创建时间): 21:00
24  * Version(版本): 1.0
25  * Description(描述): 消费者-延时消息
26  */
27
28 public class DelayConsumer
29 {
30     /**
31      * 简单日期格式
32      */
33     static SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("HH:mm:ss");
34
35     public static void main(String[] args) throws MQClientException
36     {
37         //消费者
38         DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
39         //订阅
40         defaultMQPushConsumer.subscribe("test_topic", "*");
41         //注册监听器
42         defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
43         {
44             @Override
45             public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
46             {
47                 //当前时间
48                 String format = simpleDateFormat.format(new Date());
49                 for (MessageExt messageExt : list)
50                 {
51                     //打印消息
52                     System.out.println("当前时间: " + format + " " + "队列
id: " + messageExt.getQueueId()
53                                     + " " + "消息: " + new String(messageExt.getBody(),
StandardCharsets.UTF_8));
54                 }
55                 return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
56             }
57         });
58         //启动
59         defaultMQPushConsumer.start();
60     }
61 }

```

启动

先启动消费者，再启动生产者

1	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息0
2	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息1
3	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息2
4	当前时间: 21:07:35	队列id: 1	消息: 21:07:05 --> 消息3
5	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息4
6	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息5
7	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息6
8	当前时间: 21:07:35	队列id: 1	消息: 21:07:05 --> 消息7
9	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息8
10	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息9
11	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息10
12	当前时间: 21:07:35	队列id: 1	消息: 21:07:05 --> 消息11
13	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息12
14	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息13
15	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息14
16	当前时间: 21:07:35	队列id: 1	消息: 21:07:05 --> 消息15
17	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息16
18	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息17
19	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息18
20	当前时间: 21:07:35	队列id: 1	消息: 21:07:05 --> 消息19
21	当前时间: 21:07:35	队列id: 2	消息: 21:07:05 --> 消息20
22	当前时间: 21:07:35	队列id: 3	消息: 21:07:05 --> 消息21
23	当前时间: 21:07:35	队列id: 0	消息: 21:07:05 --> 消息22
24	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息23
25	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息24
26	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息25
27	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息26
28	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息27
29	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息28
30	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息29
31	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息30
32	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息31
33	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息32
34	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息33
35	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息34
36	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息35
37	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息36
38	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息37
39	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息38
40	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息39
41	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息40
42	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息41
43	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息42
44	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息43
45	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息44
46	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息45
47	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息46
48	当前时间: 21:07:36	队列id: 1	消息: 21:07:06 --> 消息47
49	当前时间: 21:07:36	队列id: 2	消息: 21:07:06 --> 消息48
50	当前时间: 21:07:36	队列id: 3	消息: 21:07:06 --> 消息49
51	当前时间: 21:07:36	队列id: 0	消息: 21:07:06 --> 消息50


```

1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.common.message.Message;
7 import org.apache.rocketmq.remoting.exception.RemotingException;
8
9 import java.nio.charset.StandardCharsets;
10 import java.text.SimpleDateFormat;
11 import java.util.Date;
12
13 /**
14  * Project name(项目名称): RocketMQ_延迟消息的发送与接收
15  * Package(包名): mao.producer
16  * Class(类名): DelayProducer2
17  * Author(作者): mao
18  * Author QQ: 1296193245
19  * GitHub: https://github.com/maomao124/
20  * Date(创建日期): 2022/12/5
21  * Time(创建时间): 21:11
22  * Version(版本): 1.0
23  * Description(描述): 无
24  */
25
26 public class DelayProducer2
27 {
28     /**
29      * 简单日期格式
30      */
31     static SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("HH:mm:ss");
32
33     /**
34      * 得到int随机
35      *
36      * @param min 最小值
37      * @param max 最大值
38      * @return int
39      */
40     public static int getIntRandom(int min, int max)
41     {
42         if (min > max)
43         {
44             min = max;
45         }
46         return min + (int) (Math.random() * (max - min + 1));
47     }
48
49     public static void main(String[] args)
50         throws MQClientException, MQBrokerException, RemotingException,
InterruptedException
51     {
52         //生产者
53         DefaultMQProducer defaultMQProducer = new
DefaultMQProducer("mao_group");
54         //设置nameserver地址
55         defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");

```



```

56         //启动
57         defaultMQProducer.start();
58         //发送100条延迟消息
59         for (int i = 0; i < 100; i++)
60         {
61             //时间
62             String format = SimpleDateFormat.format(new Date());
63             //消息对象
64             Message message = new Message("test_topic", (format + " --> 消息"
+ i).getBytes(StandardCharsets.UTF_8));
65             //设置随机延时
66             //int level = getIntRandom(1, 4);
67             int level = i < 50 ? 4 : 2;
68             message.setDelayTimeLevel(level);
69             //发送消息
70             defaultMQProducer.send(message);
71             //打印
72             System.out.println(format + " --> 消息" + i + "    延迟水平: " +
level);
73         }
74         //关闭
75         defaultMQProducer.shutdown();
76     }
77 }

```

先运行消费者，再运行生产者

生产者打印的消息

```

1  21:16:09 --> 消息0    延迟水平: 4
2  21:16:09 --> 消息1    延迟水平: 4
3  21:16:09 --> 消息2    延迟水平: 4
4  21:16:09 --> 消息3    延迟水平: 4
5  21:16:09 --> 消息4    延迟水平: 4
6  21:16:09 --> 消息5    延迟水平: 4
7  21:16:09 --> 消息6    延迟水平: 4
8  21:16:09 --> 消息7    延迟水平: 4
9  21:16:09 --> 消息8    延迟水平: 4
10 21:16:09 --> 消息9    延迟水平: 4
11 21:16:09 --> 消息10   延迟水平: 4
12 21:16:09 --> 消息11   延迟水平: 4
13 21:16:09 --> 消息12   延迟水平: 4
14 21:16:09 --> 消息13   延迟水平: 4
15 21:16:09 --> 消息14   延迟水平: 4
16 21:16:09 --> 消息15   延迟水平: 4
17 21:16:09 --> 消息16   延迟水平: 4
18 21:16:09 --> 消息17   延迟水平: 4
19 21:16:09 --> 消息18   延迟水平: 4
20 21:16:09 --> 消息19   延迟水平: 4
21 21:16:09 --> 消息20   延迟水平: 4
22 21:16:09 --> 消息21   延迟水平: 4
23 21:16:09 --> 消息22   延迟水平: 4

```

24	21:16:09	-->	消息23	延迟水平: 4
25	21:16:09	-->	消息24	延迟水平: 4
26	21:16:09	-->	消息25	延迟水平: 4
27	21:16:09	-->	消息26	延迟水平: 4
28	21:16:09	-->	消息27	延迟水平: 4
29	21:16:09	-->	消息28	延迟水平: 4
30	21:16:09	-->	消息29	延迟水平: 4
31	21:16:09	-->	消息30	延迟水平: 4
32	21:16:09	-->	消息31	延迟水平: 4
33	21:16:09	-->	消息32	延迟水平: 4
34	21:16:09	-->	消息33	延迟水平: 4
35	21:16:09	-->	消息34	延迟水平: 4
36	21:16:09	-->	消息35	延迟水平: 4
37	21:16:09	-->	消息36	延迟水平: 4
38	21:16:09	-->	消息37	延迟水平: 4
39	21:16:09	-->	消息38	延迟水平: 4
40	21:16:09	-->	消息39	延迟水平: 4
41	21:16:09	-->	消息40	延迟水平: 4
42	21:16:09	-->	消息41	延迟水平: 4
43	21:16:09	-->	消息42	延迟水平: 4
44	21:16:09	-->	消息43	延迟水平: 4
45	21:16:09	-->	消息44	延迟水平: 4
46	21:16:09	-->	消息45	延迟水平: 4
47	21:16:09	-->	消息46	延迟水平: 4
48	21:16:09	-->	消息47	延迟水平: 4
49	21:16:09	-->	消息48	延迟水平: 4
50	21:16:09	-->	消息49	延迟水平: 4
51	21:16:09	-->	消息50	延迟水平: 2
52	21:16:10	-->	消息51	延迟水平: 2
53	21:16:10	-->	消息52	延迟水平: 2
54	21:16:10	-->	消息53	延迟水平: 2
55	21:16:10	-->	消息54	延迟水平: 2
56	21:16:10	-->	消息55	延迟水平: 2
57	21:16:10	-->	消息56	延迟水平: 2
58	21:16:10	-->	消息57	延迟水平: 2
59	21:16:10	-->	消息58	延迟水平: 2
60	21:16:10	-->	消息59	延迟水平: 2
61	21:16:10	-->	消息60	延迟水平: 2
62	21:16:10	-->	消息61	延迟水平: 2
63	21:16:10	-->	消息62	延迟水平: 2
64	21:16:10	-->	消息63	延迟水平: 2
65	21:16:10	-->	消息64	延迟水平: 2
66	21:16:10	-->	消息65	延迟水平: 2
67	21:16:10	-->	消息66	延迟水平: 2
68	21:16:10	-->	消息67	延迟水平: 2
69	21:16:10	-->	消息68	延迟水平: 2
70	21:16:10	-->	消息69	延迟水平: 2
71	21:16:10	-->	消息70	延迟水平: 2
72	21:16:10	-->	消息71	延迟水平: 2
73	21:16:10	-->	消息72	延迟水平: 2
74	21:16:10	-->	消息73	延迟水平: 2
75	21:16:10	-->	消息74	延迟水平: 2
76	21:16:10	-->	消息75	延迟水平: 2
77	21:16:10	-->	消息76	延迟水平: 2
78	21:16:10	-->	消息77	延迟水平: 2
79	21:16:10	-->	消息78	延迟水平: 2
80	21:16:10	-->	消息79	延迟水平: 2
81	21:16:10	-->	消息80	延迟水平: 2

82	21:16:10 -->	消息81	延迟水平: 2
83	21:16:10 -->	消息82	延迟水平: 2
84	21:16:10 -->	消息83	延迟水平: 2
85	21:16:10 -->	消息84	延迟水平: 2
86	21:16:10 -->	消息85	延迟水平: 2
87	21:16:10 -->	消息86	延迟水平: 2
88	21:16:10 -->	消息87	延迟水平: 2
89	21:16:10 -->	消息88	延迟水平: 2
90	21:16:10 -->	消息89	延迟水平: 2
91	21:16:10 -->	消息90	延迟水平: 2
92	21:16:10 -->	消息91	延迟水平: 2
93	21:16:10 -->	消息92	延迟水平: 2
94	21:16:10 -->	消息93	延迟水平: 2
95	21:16:10 -->	消息94	延迟水平: 2
96	21:16:10 -->	消息95	延迟水平: 2
97	21:16:10 -->	消息96	延迟水平: 2
98	21:16:10 -->	消息97	延迟水平: 2
99	21:16:10 -->	消息98	延迟水平: 2
100	21:16:10 -->	消息99	延迟水平: 2

消费者打印的消息

1	当前时间: 21:16:15	队列id: 3	消息: 21:16:09 -->	消息50
2	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息51
3	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息52
4	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息53
5	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息54
6	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息55
7	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息56
8	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息57
9	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息58
10	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息59
11	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息60
12	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息61
13	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息62
14	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息63
15	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息64
16	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息65
17	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息66
18	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息67
19	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息68
20	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息69
21	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息70
22	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息71
23	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息72
24	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息73
25	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息74
26	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息75
27	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息76
28	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息77
29	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息78
30	当前时间: 21:16:15	队列id: 0	消息: 21:16:10 -->	消息79
31	当前时间: 21:16:15	队列id: 1	消息: 21:16:10 -->	消息80
32	当前时间: 21:16:15	队列id: 3	消息: 21:16:10 -->	消息82
33	当前时间: 21:16:15	队列id: 2	消息: 21:16:10 -->	消息81

92	当前时间: 21:16:39	队列id: 2	消息: 21:16:09 --> 消息41
93	当前时间: 21:16:39	队列id: 3	消息: 21:16:09 --> 消息42
94	当前时间: 21:16:39	队列id: 0	消息: 21:16:09 --> 消息43
95	当前时间: 21:16:39	队列id: 1	消息: 21:16:09 --> 消息44
96	当前时间: 21:16:39	队列id: 2	消息: 21:16:09 --> 消息45
97	当前时间: 21:16:39	队列id: 3	消息: 21:16:09 --> 消息46
98	当前时间: 21:16:39	队列id: 0	消息: 21:16:09 --> 消息47
99	当前时间: 21:16:39	队列id: 1	消息: 21:16:09 --> 消息48
100	当前时间: 21:16:39	队列id: 2	消息: 21:16:09 --> 消息49

观察相同队列的消息顺序

比如消息91和消息43，对应生产者而言，43比91更早进入队列，队列先进先出的特性，在消费者里43应该比91更早出队，但是事实上却没有，rocketMQ解决了此问题，更加符合业务要求，rabbitMQ却没有解决，但是可以延迟交换机插件解决此问题，rabbitMQ的延迟交换机插件没有时间限制

批量消息

批量发送消息能显著提高传递小消息的性能。限制是这些批量消息应该有相同的topic，相同的waitStoreMsgOK，而且不能是延时消息。此外，这一批消息的总大小不应超过4MB。

生产

```

1 package mao.producer;
2
3 import org.apache.rocketmq.common.message.Message;
4
5 import java.util.Iterator;
6 import java.util.List;
7 import java.util.Map;
8
9 /**
10  * Project name(项目名称): RocketMQ_批量消息的发送与接收
11  * Package(包名): mao.producer
12  * Class(类名): Listsplitter
13  * Author(作者): mao
14  * Author QQ: 1296193245
15  * Github: https://github.com/maomao124/

```

```

16  * Date(创建日期): 2022/12/6
17  * Time(创建时间): 14:14
18  * Version(版本): 1.0
19  * Description(描述): 将消息拆分, 没次遍历得到的list集合的数据大小不大于4M
20  */
21
22  public class ListSplitter implements Iterator<List<Message>>
23  {
24
25      /**
26       * 大小限制
27       */
28      private static final int SIZE_LIMIT = 1000 * 40;
29
30      /**
31       * 消息列表
32       */
33      private final List<Message> messages;
34
35      /**
36       * 当前索引位置
37       */
38      private int currIndex;
39
40      /**
41       * 构造方法
42       *
43       * @param messages 消息列表
44       */
45      public ListSplitter(List<Message> messages)
46      {
47          this.messages = messages;
48      }
49
50      @Override
51      public boolean hasNext()
52      {
53          return currIndex < messages.size();
54      }
55
56      /**
57       * 下一个
58       *
59       * @return {@link List}<{@link Message}>
60       */
61      @Override
62      public List<Message> next()
63      {
64          //当前的起始索引
65          int nextIndex = currIndex;
66          //当前的总大小
67          int totalSize = 0;
68          //遍历
69          for (; nextIndex < messages.size(); nextIndex++)
70          {
71              //取到消息对象
72              Message message = messages.get(nextIndex);
73              //计算当前消息对象的大小

```

```

74         int tmpSize = message.getTopic().length() +
message.getBody().length;
75         //消息属性
76         Map<String, String> messageProperties =
message.getProperties();
77         //遍历属性
78         for (Map.Entry<String, String> entry :
messageProperties.entrySet())
79         {
80             //计算属性的大小,添加进tmpSize
81             tmpSize += entry.getKey().length() +
entry.getValue().length();
82         }
83         //日志-20字节
84         tmpSize = tmpSize + 20;
85         //判断当前大小(当前一个消息的大小)是否超过了限制的最大大小
86         if (tmpSize > SIZE_LIMIT)
87         {
88             //超过了限制的最大大小
89             if (nextIndex - currIndex == 0)
90             {
91                 //假如下一个子列表没有元素,则添加这个子列表然后退出循环,否则只是
退出循环
92                 nextIndex++;
93             }
94             break;
95         }
96         //当前遍历的总大小加上这次遍历的大小大于限制的大小
97         if (tmpSize + totalSize > SIZE_LIMIT)
98         {
99             //直接退出循环
100             break;
101         }
102         else
103         {
104             //当前遍历的总大小加上这次遍历的大小没有大于限制的大小
105             //继续遍历
106             totalSize += tmpSize;
107         }
108     }
109     //生成列表并返回
110     List<Message> subList = messages.subList(currIndex, nextIndex);
111     currIndex = nextIndex;
112     return subList;
113 }
114 }
115

```

```

1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;

```

```

5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.client.producer.SendResult;
7 import org.apache.rocketmq.common.message.Message;
8 import org.apache.rocketmq.remoting.exception.RemotingException;
9
10 import java.nio.charset.StandardCharsets;
11 import java.util.ArrayList;
12 import java.util.List;
13
14 /**
15  * Project name(项目名称): RocketMQ_批量消息的发送与接收
16  * Package(包名): mao.producer
17  * Class(类名): BulkProducer
18  * Author(作者): mao
19  * Author QQ: 1296193245
20  * Github: https://github.com/maomao124/
21  * Date(创建日期): 2022/12/6
22  * Time(创建时间): 14:34
23  * Version(版本): 1.0
24  * Description(描述): 批量消息-生产者
25  */
26
27 public class BulkProducer
28 {
29     /**
30      * 得到int随机
31      *
32      * @param min 最小值
33      * @param max 最大值
34      * @return int
35      */
36     public static int getIntRandom(int min, int max)
37     {
38         if (min > max)
39         {
40             min = max;
41         }
42         return min + (int) (Math.random() * (max - min + 1));
43     }
44
45     public static String getBody()
46     {
47         StringBuilder stringBuilder = new StringBuilder(2000);
48         for (int i = 0; i < getIntRandom(10, 500); i++)
49         {
50             stringBuilder.append(Integer.toString(getIntRandom(2000,
51 80000)));
52         }
53         return stringBuilder.toString();
54     }
55
56     public static void main(String[] args)
57         throws MQClientException, MQBrokerException, RemotingException,
58         InterruptedException
59     {
60         List<Message> messageList = new ArrayList<>(1000);
61         for (int i = 0; i < 1000; i++)

```



```

61     {
62         Message message = new Message("test_topic",
getBody().getBytes(StandardCharsets.UTF_8));
63         messageList.add(message);
64     }
65     ListSplitter listSplitter = new ListSplitter(messageList);
66     int i = 1;
67     int position = 0;
68     List<List<Message>> bulkMessageList = new ArrayList<>();
69     while (listSplitter.hasNext())
70     {
71         System.out.println("-----");
72         System.out.println("第" + i + "次遍历");
73         List<Message> messages = listSplitter.next();
74         System.out.println("列表大小: " + messages.size());
75         System.out.println("位置: " + position + "-" + (position +
messages.size() - 1));
76         System.out.println("-----");
77         i++;
78         position = position + messages.size();
79         bulkMessageList.add(messages);
80     }
81
82
83     System.out.println();
84     System.out.println("开始发送批量消息");
85
86     //生产者
87     DefaultMQProducer defaultMQProducer = new
DefaultMQProducer("mao_group");
88     //设置nameserver地址
89     defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
90     //启动
91     defaultMQProducer.start();
92     //发送批量消息
93     for (int i1 = 0; i1 < bulkMessageList.size(); i1++)
94     {
95         System.out.println("-----");
96         List<Message> messages = bulkMessageList.get(i1);
97         System.out.println("正在发送第" + (i1 + 1) + "批消息");
98         System.out.println("列表大小: " + messages.size());
99         SendResult sendResult = defaultMQProducer.send(messages);
100         //System.out.println("发送完成, 发送结果: " + sendResult);
101         System.out.println("-----");
102     }
103     System.out.println("完成");
104     defaultMQProducer.shutdown();
105 }
106 }

```

消费

```
1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import
5 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6 import
7 org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
8 import
9 org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
10 import org.apache.rocketmq.client.exception.MQClientException;
11 import org.apache.rocketmq.common.message.MessageExt;
12
13 import java.nio.charset.StandardCharsets;
14 import java.util.List;
15
16 /**
17  * Project name(项目名称): RocketMQ_批量消息的发送与接收
18  * Package(包名): mao.consumer
19  * Class(类名): BulkConsumer
20  * Author(作者): mao
21  * Author QQ: 1296193245
22  * GitHub: https://github.com/maomao124/
23  * Date(创建日期): 2022/12/6
24  * Time(创建时间): 15:26
25  * Version(版本): 1.0
26  * Description(描述): 消费者
27  */
28
29 public class BulkConsumer
30 {
31     public static void main(String[] args) throws MQClientException
32     {
33         //消费者
34         DefaultMQPushConsumer defaultMQPushConsumer = new
35         DefaultMQPushConsumer("mao_group");
36         //设置nameserver地址
37         defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
38         //订阅
39         defaultMQPushConsumer.subscribe("test_topic", "*");
40         //注册监听器
41         defaultMQPushConsumer.registerMessageListener(new
42         MessageListenerConcurrently()
43         {
44             @Override
45             public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
46             list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
47             {
48                 for (MessageExt messageExt : list)
49                 {
50                     System.out.println(new String(messageExt.getBody(),
51                     StandardCharsets.UTF_8));
52                     System.out.println();
53                 }
54             }
55         })
56     }
57 }
```

```

47         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
48     }
49 });
50 //启动
51 defaultMQPushConsumer.start();
52 }
53 }

```

启动

如果设置成4M，将会出现以下异常

```

1 Exception in thread "main"
  org.apache.rocketmq.client.exception.MQBrokerException: CODE: 13 DESC: the
  message is illegal, maybe msg body or properties length not matched. msg body
  length limit 128k, msg properties length limit 32k. BROKER:
  113.221.235.60:10911

```

生产者

```

1  -----
2  第1次遍历
3  列表大小: 72
4  位置: 0-71
5  -----
6  -----
7  第2次遍历
8  列表大小: 70
9  位置: 72-141
10 -----
11 -----
12 第3次遍历
13 列表大小: 62
14 位置: 142-203
15 -----
16 -----
17 第4次遍历
18 列表大小: 66
19 位置: 204-269
20 -----
21 -----
22 第5次遍历
23 列表大小: 64
24 位置: 270-333
25 -----
26 -----
27 第6次遍历

```

28 列表大小: 67
29 位置: 334-400
30 -----
31 -----
32 第7次遍历
33 列表大小: 63
34 位置: 401-463
35 -----
36 -----
37 第8次遍历
38 列表大小: 64
39 位置: 464-527
40 -----
41 -----
42 第9次遍历
43 列表大小: 62
44 位置: 528-589
45 -----
46 -----
47 第10次遍历
48 列表大小: 67
49 位置: 590-656
50 -----
51 -----
52 第11次遍历
53 列表大小: 59
54 位置: 657-715
55 -----
56 -----
57 第12次遍历
58 列表大小: 69
59 位置: 716-784
60 -----
61 -----
62 第13次遍历
63 列表大小: 70
64 位置: 785-854
65 -----
66 -----
67 第14次遍历
68 列表大小: 66
69 位置: 855-920
70 -----
71 -----
72 第15次遍历
73 列表大小: 69
74 位置: 921-989
75 -----
76 -----
77 第16次遍历
78 列表大小: 10
79 位置: 990-999
80 -----
81 -----
82 开始发送批量消息
83 -----
84 正在发送第1批消息
85 列表大小: 72

86 -----
87 -----
88 正在发送第2批消息
89 列表大小: 70
90 -----
91 -----
92 正在发送第3批消息
93 列表大小: 62
94 -----
95 -----
96 正在发送第4批消息
97 列表大小: 66
98 -----
99 -----
100 正在发送第5批消息
101 列表大小: 64
102 -----
103 -----
104 正在发送第6批消息
105 列表大小: 67
106 -----
107 -----
108 正在发送第7批消息
109 列表大小: 63
110 -----
111 -----
112 正在发送第8批消息
113 列表大小: 64
114 -----
115 -----
116 正在发送第9批消息
117 列表大小: 62
118 -----
119 -----
120 正在发送第10批消息
121 列表大小: 67
122 -----
123 -----
124 正在发送第11批消息
125 列表大小: 59
126 -----
127 -----
128 正在发送第12批消息
129 列表大小: 69
130 -----
131 -----
132 正在发送第13批消息
133 列表大小: 70
134 -----
135 -----
136 正在发送第14批消息
137 列表大小: 66
138 -----
139 -----
140 正在发送第15批消息
141 列表大小: 69
142 -----
143 -----

```
144 | 正在发送第16批消息
145 | 列表大小: 10
146 | -----
147 | 完成
```

过滤消息

在大多数情况下，TAG是一个简单而有用的设计，其可以选择您想要的消息

如：

```
1 | consumer.subscribe("TOPIC", "TAGA || TAGB || TAGC");
```

消费者将接收包含TAGA或TAGB或TAGC的消息。但是限制是一个消息只能有一个标签，这对于复杂的场景可能不起作用。在这种情况下，可以使用SQL表达式筛选消息。SQL特性可以通过发送消息时的属性来进行计算。在RocketMQ定义的语法下，可以实现一些简单的逻辑

SQL基本语法

RocketMQ只定义了一些基本语法来支持这个特性。你也可以很容易地扩展它。

- 数值比较，比如：>, >=, <, <=, **BETWEEN**, =;
- 字符比较，比如：=, <>, **IN**;
- **IS NULL** 或者 **IS NOT NULL**;
- 逻辑符号 **AND**, **OR**, **NOT**;

常量支持类型为：

- 数值，比如：123, 3.1415;
- 字符，比如：'abc'，必须用单引号包裹起来；
- **NULL**，特殊的常量
- 布尔值，**TRUE** 或 **FALSE**

只有使用push模式的消费者才能用使用SQL92标准的sql语句，接口如下：

```
1 | public void subscribe(finalString topic, final MessageSelector
   | messageSelector)
```

消息生产者

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.common.message.Message;
7 import org.apache.rocketmq.remoting.exception.RemotingException;
8
9 import java.nio.charset.StandardCharsets;
10
11 /**
12  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
13  * Package(包名): mao.producer
14  * Class(类名): FilterProducer1
15  * Author(作者): mao
16  * Author QQ: 1296193245
17  * GitHub: https://github.com/maomao124/
18  * Date(创建日期): 2022/12/7
19  * Time(创建时间): 14:16
20  * Version(版本): 1.0
21  * Description(描述): 生产者
22  */
23
24 public class FilterProducer1
25 {
26     /**
27      * 标签数组
28      */
29     private static final String[] TAGS = {"tag1", "tag2", "tag3", "tag4",
30 "tag5"};
31
32     public static void main(String[] args)
33         throws MQClientException, MQBrokerException, RemotingException,
34         InterruptedException
35     {
36         //生产者
37         DefaultMQProducer defaultMQProducer = new
38         DefaultMQProducer("mao_group");
39         //nameserver地址
40         defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
41         //启动
42         defaultMQProducer.start();
43         //发送消息
44         for (int i = 0; i < 100; i++)
45         {
46             //标签
47             String tag = TAGS[i % (TAGS.length)];
48             //消息体字符串
49             String msg = "消息" + i + "    消息标签: " + tag;
50             //消息对象
```

```

49         Message message = new Message("test_topic", tag,
msg.getBytes(StandardCharsets.UTF_8));
50         //打印
51         System.out.println(msg);
52         //发送消息
53         defaultMQProducer.send(message);
54     }
55     //关闭
56     defaultMQProducer.shutdown();
57 }
58 }

```

消息消费者

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
6  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
7  import org.apache.rocketmq.client.exception.MQClientException;
8  import org.apache.rocketmq.common.message.MessageExt;
9  import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
10
11 import java.nio.charset.StandardCharsets;
12 import java.util.List;
13
14 /**
15  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
16  * Package(包名): mao.consumer
17  * Class(类名): FilterConsumer1
18  * Author(作者): mao
19  * Author QQ: 1296193245
20  * GitHub: https://github.com/maomao124/
21  * Date(创建日期): 2022/12/7
22  * Time(创建时间): 14:24
23  * Version(版本): 1.0
24  * Description(描述): 消费者, 消费标签为1和3的消息
25  */
26
27 public class FilterConsumer1
28 {
29     public static void main(String[] args) throws MQClientException
30     {
31         //消费者
32         DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");

```



```

33 //设置nameserver地址
34 defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
35 //设置消费模式-广播模式
36 defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
37 //订阅
38 defaultMQPushConsumer.subscribe("test_topic", "tag1 || tag3");
39 //注册监听器
40 defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
41 {
42     @Override
43     public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
44     {
45         for (MessageExt messageExt : list)
46         {
47             System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
48         }
49         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
50     }
51 });
52 //启动
53 defaultMQPushConsumer.start();
54 }
55 }

```

```

1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
5 import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
6 import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
7 import org.apache.rocketmq.client.exception.MQClientException;
8 import org.apache.rocketmq.common.message.MessageExt;
9 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
10
11 import java.nio.charset.StandardCharsets;
12 import java.util.List;
13
14 /**
15  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
16  * Package(包名): mao.consumer
17  * Class(类名): FilterConsumer2
18  * Author(作者): mao
19  * Author QQ: 1296193245
20  * Github: https://github.com/maomao124/
21  * Date(创建日期): 2022/12/7
22  * Time(创建时间): 14:32
23  * Version(版本): 1.0

```

```

24  * Description(描述): 消费者2, 只消费标签为5的消息
25  */
26
27  public class FilterConsumer2
28  {
29      public static void main(String[] args) throws MQClientException
30      {
31          //消费者
32          DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
33          //设置nameserver地址
34          defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
35          //设置消费模式-广播模式
36          defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
37          //订阅
38          defaultMQPushConsumer.subscribe("test_topic", "tag5");
39          //注册监听器
40          defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
41          {
42              @Override
43              public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
44              {
45                  for (MessageExt messageExt : list)
46                  {
47                      System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
48                  }
49                  return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
50              }
51          });
52          //启动
53          defaultMQPushConsumer.start();
54      }
55  }

```

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
6  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
7  import org.apache.rocketmq.client.exception.MQClientException;
8  import org.apache.rocketmq.common.message.MessageExt;
9  import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
10
11  import java.nio.charset.StandardCharsets;
12  import java.util.List;
13

```

```

14  /**
15   * Project name(项目名称): RocketMQ_过滤消息的发送与接收
16   * Package(包名): mao.consumer
17   * Class(类名): FilterConsumer3
18   * Author(作者): mao
19   * Author QQ: 1296193245
20   * GitHub: https://github.com/maomao124/
21   * Date(创建日期): 2022/12/7
22   * Time(创建时间): 14:34
23   * Version(版本): 1.0
24   * Description(描述): 消费者, 消费标签为1,2和4的消息
25   */
26
27  public class FilterConsumer3
28  {
29      public static void main(String[] args) throws MQClientException
30      {
31          //消费者
32          DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
33          //设置nameserver地址
34          defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
35          //设置消费模式-广播模式
36          defaultMQPushConsumer.setMessageMode(MessageMode.BROADCASTING);
37          //订阅
38          defaultMQPushConsumer.subscribe("test_topic", "tag1 || tag2 ||
tag4");
39          //注册监听器
40          defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
41          {
42              @Override
43              public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
44              {
45                  for (MessageExt messageExt : list)
46                  {
47                      System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
48                  }
49                  return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
50              }
51          });
52          //启动
53          defaultMQPushConsumer.start();
54      }
55  }

```

启动测试

启动消费者1，再启动生产者

生产者打印的信息：

1	消息0	消息标签: tag1
2	消息1	消息标签: tag2
3	消息2	消息标签: tag3
4	消息3	消息标签: tag4
5	消息4	消息标签: tag5
6	消息5	消息标签: tag1
7	消息6	消息标签: tag2
8	消息7	消息标签: tag3
9	消息8	消息标签: tag4
10	消息9	消息标签: tag5
11	消息10	消息标签: tag1
12	消息11	消息标签: tag2
13	消息12	消息标签: tag3
14	消息13	消息标签: tag4
15	消息14	消息标签: tag5
16	消息15	消息标签: tag1
17	消息16	消息标签: tag2
18	消息17	消息标签: tag3
19	消息18	消息标签: tag4
20	消息19	消息标签: tag5
21	消息20	消息标签: tag1
22	消息21	消息标签: tag2
23	消息22	消息标签: tag3
24	消息23	消息标签: tag4
25	消息24	消息标签: tag5
26	消息25	消息标签: tag1
27	消息26	消息标签: tag2
28	消息27	消息标签: tag3
29	消息28	消息标签: tag4
30	消息29	消息标签: tag5
31	消息30	消息标签: tag1
32	消息31	消息标签: tag2
33	消息32	消息标签: tag3
34	消息33	消息标签: tag4
35	消息34	消息标签: tag5
36	消息35	消息标签: tag1
37	消息36	消息标签: tag2
38	消息37	消息标签: tag3
39	消息38	消息标签: tag4
40	消息39	消息标签: tag5
41	消息40	消息标签: tag1
42	消息41	消息标签: tag2
43	消息42	消息标签: tag3
44	消息43	消息标签: tag4
45	消息44	消息标签: tag5
46	消息45	消息标签: tag1
47	消息46	消息标签: tag2
48	消息47	消息标签: tag3
49	消息48	消息标签: tag4
50	消息49	消息标签: tag5

51	消息50	消息标签: tag1
52	消息51	消息标签: tag2
53	消息52	消息标签: tag3
54	消息53	消息标签: tag4
55	消息54	消息标签: tag5
56	消息55	消息标签: tag1
57	消息56	消息标签: tag2
58	消息57	消息标签: tag3
59	消息58	消息标签: tag4
60	消息59	消息标签: tag5
61	消息60	消息标签: tag1
62	消息61	消息标签: tag2
63	消息62	消息标签: tag3
64	消息63	消息标签: tag4
65	消息64	消息标签: tag5
66	消息65	消息标签: tag1
67	消息66	消息标签: tag2
68	消息67	消息标签: tag3
69	消息68	消息标签: tag4
70	消息69	消息标签: tag5
71	消息70	消息标签: tag1
72	消息71	消息标签: tag2
73	消息72	消息标签: tag3
74	消息73	消息标签: tag4
75	消息74	消息标签: tag5
76	消息75	消息标签: tag1
77	消息76	消息标签: tag2
78	消息77	消息标签: tag3
79	消息78	消息标签: tag4
80	消息79	消息标签: tag5
81	消息80	消息标签: tag1
82	消息81	消息标签: tag2
83	消息82	消息标签: tag3
84	消息83	消息标签: tag4
85	消息84	消息标签: tag5
86	消息85	消息标签: tag1
87	消息86	消息标签: tag2
88	消息87	消息标签: tag3
89	消息88	消息标签: tag4
90	消息89	消息标签: tag5
91	消息90	消息标签: tag1
92	消息91	消息标签: tag2
93	消息92	消息标签: tag3
94	消息93	消息标签: tag4
95	消息94	消息标签: tag5
96	消息95	消息标签: tag1
97	消息96	消息标签: tag2
98	消息97	消息标签: tag3
99	消息98	消息标签: tag4
100	消息99	消息标签: tag5

消费者1打印的信息

1	消息0	消息标签: tag1
---	-----	------------

2	消息2	消息标签: tag3
3	消息5	消息标签: tag1
4	消息7	消息标签: tag3
5	消息10	消息标签: tag1
6	消息12	消息标签: tag3
7	消息15	消息标签: tag1
8	消息17	消息标签: tag3
9	消息20	消息标签: tag1
10	消息22	消息标签: tag3
11	消息25	消息标签: tag1
12	消息27	消息标签: tag3
13	消息30	消息标签: tag1
14	消息32	消息标签: tag3
15	消息35	消息标签: tag1
16	消息37	消息标签: tag3
17	消息40	消息标签: tag1
18	消息42	消息标签: tag3
19	消息45	消息标签: tag1
20	消息47	消息标签: tag3
21	消息50	消息标签: tag1
22	消息52	消息标签: tag3
23	消息55	消息标签: tag1
24	消息57	消息标签: tag3
25	消息60	消息标签: tag1
26	消息62	消息标签: tag3
27	消息65	消息标签: tag1
28	消息67	消息标签: tag3
29	消息70	消息标签: tag1
30	消息72	消息标签: tag3
31	消息75	消息标签: tag1
32	消息77	消息标签: tag3
33	消息80	消息标签: tag1
34	消息82	消息标签: tag3
35	消息85	消息标签: tag1
36	消息87	消息标签: tag3
37	消息90	消息标签: tag1
38	消息92	消息标签: tag3
39	消息95	消息标签: tag1
40	消息97	消息标签: tag3

关闭消费者1，启动消费者2，再启动生产者

消费者2打印的信息

1	消息4	消息标签: tag5
2	消息9	消息标签: tag5
3	消息14	消息标签: tag5
4	消息19	消息标签: tag5
5	消息24	消息标签: tag5
6	消息29	消息标签: tag5
7	消息34	消息标签: tag5
8	消息39	消息标签: tag5

9	消息44	消息标签: tag5
10	消息49	消息标签: tag5
11	消息54	消息标签: tag5
12	消息59	消息标签: tag5
13	消息64	消息标签: tag5
14	消息69	消息标签: tag5
15	消息74	消息标签: tag5
16	消息79	消息标签: tag5
17	消息84	消息标签: tag5
18	消息89	消息标签: tag5
19	消息94	消息标签: tag5
20	消息99	消息标签: tag5

关闭消费者2，启动消费者3，再启动生产者

消费者3打印的信息

1	消息0	消息标签: tag1
2	消息1	消息标签: tag2
3	消息3	消息标签: tag4
4	消息5	消息标签: tag1
5	消息6	消息标签: tag2
6	消息8	消息标签: tag4
7	消息10	消息标签: tag1
8	消息11	消息标签: tag2
9	消息13	消息标签: tag4
10	消息15	消息标签: tag1
11	消息16	消息标签: tag2
12	消息18	消息标签: tag4
13	消息20	消息标签: tag1
14	消息21	消息标签: tag2
15	消息23	消息标签: tag4
16	消息25	消息标签: tag1
17	消息26	消息标签: tag2
18	消息28	消息标签: tag4
19	消息30	消息标签: tag1
20	消息31	消息标签: tag2
21	消息33	消息标签: tag4
22	消息35	消息标签: tag1
23	消息36	消息标签: tag2
24	消息38	消息标签: tag4
25	消息40	消息标签: tag1
26	消息41	消息标签: tag2
27	消息43	消息标签: tag4
28	消息45	消息标签: tag1
29	消息46	消息标签: tag2
30	消息48	消息标签: tag4
31	消息50	消息标签: tag1
32	消息51	消息标签: tag2
33	消息53	消息标签: tag4
34	消息55	消息标签: tag1

35	消息56	消息标签: tag2
36	消息58	消息标签: tag4
37	消息60	消息标签: tag1
38	消息61	消息标签: tag2
39	消息63	消息标签: tag4
40	消息65	消息标签: tag1
41	消息66	消息标签: tag2
42	消息68	消息标签: tag4
43	消息70	消息标签: tag1
44	消息71	消息标签: tag2
45	消息73	消息标签: tag4
46	消息75	消息标签: tag1
47	消息76	消息标签: tag2
48	消息78	消息标签: tag4
49	消息80	消息标签: tag1
50	消息81	消息标签: tag2
51	消息83	消息标签: tag4
52	消息85	消息标签: tag1
53	消息86	消息标签: tag2
54	消息88	消息标签: tag4
55	消息90	消息标签: tag1
56	消息91	消息标签: tag2
57	消息93	消息标签: tag4
58	消息95	消息标签: tag1
59	消息96	消息标签: tag2
60	消息98	消息标签: tag4

消息生产者

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQBrokerException;
4 import org.apache.rocketmq.client.exception.MQClientException;
5 import org.apache.rocketmq.client.producer.DefaultMQProducer;
6 import org.apache.rocketmq.common.message.Message;
7 import org.apache.rocketmq.remoting.exception.RemotingException;
8
9 import java.nio.charset.StandardCharsets;
10
11 /**
12  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
13  * Package(包名): mao.producer
14  * Class(类名): FilterProducer2
15  * Author(作者): mao
16  * Author QQ: 1296193245
17  * GitHub: https://github.com/maomao124/
18  * Date(创建日期): 2022/12/7
19  * Time(创建时间): 14:51
```



```

20  * Version(版本): 1.0
21  * Description(描述): 无
22  */
23
24  public class FilterProducer2
25  {
26      /**
27       * 得到int随机
28       *
29       * @param min 最小值
30       * @param max 最大值
31       * @return int
32       */
33      public static int getIntRandom(int min, int max)
34      {
35          if (min > max)
36          {
37              min = max;
38          }
39          return min + (int) (Math.random() * (max - min + 1));
40      }
41
42      public static void main(String[] args)
43          throws MQClientException, MQBrokerException, RemotingException,
44      InterruptedException
45      {
46          //生产者
47          DefaultMQProducer defaultMQProducer = new
48      DefaultMQProducer("mao_group");
49          //nameserver地址
50          defaultMQProducer.setNamesrvAddr("127.0.0.1:9876");
51          //启动
52          defaultMQProducer.start();
53          //发送消息
54          for (int i = 0; i < 100; i++)
55          {
56              //属性
57              int a = getIntRandom(0, 10);
58              int b = getIntRandom(0, 20);
59              //消息体字符串
60              String msg = "消息" + i + "      属性a: " + a + "      属性b: " + b;
61              //消息对象
62              Message message = new Message("test_topic",
63      msg.getBytes(StandardCharsets.UTF_8));
64              //填充属性
65              message.putUserProperty("a", String.valueOf(a));
66              message.putUserProperty("b", String.valueOf(b));
67              //打印
68              System.out.println(msg);
69              //发送消息
70              defaultMQProducer.send(message);
71          }
72      }
73      //关闭
74      defaultMQProducer.shutdown();
75  }
76  }

```

消息消费者

```
1 package mao.consumer;
2
3 import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4 import org.apache.rocketmq.client.consumer.MessageSelector;
5 import
    org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6 import
    org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
7 import
    org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
8 import org.apache.rocketmq.client.exception.MQClientException;
9 import org.apache.rocketmq.common.message.MessageExt;
10 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
11
12 import java.nio.charset.StandardCharsets;
13 import java.util.List;
14
15 /**
16  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
17  * Package(包名): mao.consumer
18  * Class(类名): FilterConsumer4
19  * Author(作者): mao
20  * Author QQ: 1296193245
21  * Github: https://github.com/maomao124/
22  * Date(创建日期): 2022/12/7
23  * Time(创建时间): 14:54
24  * Version(版本): 1.0
25  * Description(描述): 消费者, 条件: a>=7 or b<6
26  */
27
28 public class FilterConsumer4
29 {
30     public static void main(String[] args) throws MQClientException
31     {
32         //消费者
33         DefaultMQPushConsumer defaultMQPushConsumer = new
    DefaultMQPushConsumer("mao_group");
34         //设置nameserver地址
35         defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
36         //设置消费模式-广播模式
37         defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
38         //订阅
39         defaultMQPushConsumer.subscribe("test_topic",
    MessageSelector.bySql("a>=7 or b<6"));
40         //注册监听器
41         defaultMQPushConsumer.registerMessageListener(new
    MessageListenerConcurrently()
42         {
43             @Override
```

```

44         public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
45     {
46         for (MessageExt messageExt : list)
47         {
48             System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
49         }
50         return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
51     }
52     });
53     //启动
54     defaultMQPushConsumer.start();
55 }
56 }
57

```

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import org.apache.rocketmq.client.consumer.MessageSelector;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
7  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
8  import org.apache.rocketmq.client.exception.MQClientException;
9  import org.apache.rocketmq.common.message.MessageExt;
10 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
11
12 import java.nio.charset.StandardCharsets;
13 import java.util.List;
14
15 /**
16  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
17  * Package(包名): mao.consumer
18  * Class(类名): FilterConsumer5
19  * Author(作者): mao
20  * Author QQ: 1296193245
21  * GitHub: https://github.com/maomao124/
22  * Date(创建日期): 2022/12/7
23  * Time(创建时间): 14:57
24  * Version(版本): 1.0
25  * Description(描述): 消费者, 条件: a<=6 and b>11
26  */
27
28 public class FilterConsumer5
29 {
30     public static void main(String[] args) throws MQClientException
31     {
32         //消费者

```

```

33     DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
34     //设置nameserver地址
35     defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
36     //设置消费模式-广播模式
37     defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
38     //订阅
39     defaultMQPushConsumer.subscribe("test_topic",
MessageSelector.bySql("a<=6 and b>11"));
40     //注册监听器
41     defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
42     {
43         @Override
44         public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
45         {
46             for (MessageExt messageExt : list)
47             {
48                 System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
49             }
50             return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
51         }
52     });
53     //启动
54     defaultMQPushConsumer.start();
55 }
56 }

```

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import org.apache.rocketmq.client.consumer.MessageSelector;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
6  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
7  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
8  import org.apache.rocketmq.client.exception.MQClientException;
9  import org.apache.rocketmq.common.message.MessageExt;
10 import org.apache.rocketmq.common.protocol.heartbeat.MessageModel;
11
12 import java.nio.charset.StandardCharsets;
13 import java.util.List;
14
15 /**
16  * Project name(项目名称): RocketMQ_过滤消息的发送与接收
17  * Package(包名): mao.consumer
18  * Class(类名): FilterConsumer6
19  * Author(作者): mao
20  * Author QQ: 1296193245

```

```

21  * GitHub: https://github.com/maomao124/
22  * Date(创建日期): 2022/12/7
23  * Time(创建时间): 14:59
24  * Version(版本): 1.0
25  * Description(描述): 消费者, 条件: a=3 or b=10
26  */
27
28 public class FilterConsumer6
29 {
30     public static void main(String[] args) throws MQClientException
31     {
32         //消费者
33         DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
34         //设置nameserver地址
35         defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
36         //设置消费模式-广播模式
37         defaultMQPushConsumer.setMessageModel(MessageModel.BROADCASTING);
38         //订阅
39         defaultMQPushConsumer.subscribe("test_topic",
MessageSelector.bySql("a=3 or b=10"));
40         //注册监听器
41         defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
42         {
43             @Override
44             public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
45             {
46                 for (MessageExt messageExt : list)
47                 {
48                     System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
49                 }
50                 return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
51             }
52         });
53         //启动
54         defaultMQPushConsumer.start();
55     }
56 }

```

启动测试

如果出现以下异常:

```

1 | Exception in thread "main"
   | org.apache.rocketmq.client.exception.MQClientException: CODE: 1  DESC: The
   | broker does not support consumer to filter message by SQL92
2 | For more information, please visit the url,
   | http://rocketmq.apache.org/docs/faq/
3 |     at
   | org.apache.rocketmq.client.impl.MQClientAPIImpl.checkClientInBroker(MQClientA
   | PIImpl.java:2310)
4 |     at
   | org.apache.rocketmq.client.impl.factory.MQClientInstance.checkClientInBroker(
   | MQClientInstance.java:450)
5 |     at
   | org.apache.rocketmq.client.impl.consumer.DefaultMQPushConsumerImpl.start(Defa
   | ultMQPushConsumerImpl.java:656)
6 |     at
   | org.apache.rocketmq.client.consumer.DefaultMQPushConsumer.start(DefaultMQPush
   | Consumer.java:707)
7 |     at mao.consumer.FilterConsumer4.main(FilterConsumer4.java:54)

```

则需要在配置文件中需要添加配置项

```

1 | enablePropertyFilter=true

```

重启服务

启动消费者4，再启动生产者2

生产者2打印的信息

1	消息0	属性a: 8	属性b: 1
2	消息1	属性a: 3	属性b: 5
3	消息2	属性a: 9	属性b: 2
4	消息3	属性a: 6	属性b: 5
5	消息4	属性a: 8	属性b: 1
6	消息5	属性a: 8	属性b: 12
7	消息6	属性a: 1	属性b: 0
8	消息7	属性a: 7	属性b: 14
9	消息8	属性a: 1	属性b: 8
10	消息9	属性a: 8	属性b: 20
11	消息10	属性a: 1	属性b: 10
12	消息11	属性a: 8	属性b: 20
13	消息12	属性a: 1	属性b: 1
14	消息13	属性a: 3	属性b: 12
15	消息14	属性a: 3	属性b: 2
16	消息15	属性a: 6	属性b: 16
17	消息16	属性a: 3	属性b: 3

18	消息17	属性a: 5	属性b: 11
19	消息18	属性a: 6	属性b: 16
20	消息19	属性a: 3	属性b: 11
21	消息20	属性a: 3	属性b: 0
22	消息21	属性a: 9	属性b: 2
23	消息22	属性a: 0	属性b: 15
24	消息23	属性a: 9	属性b: 10
25	消息24	属性a: 8	属性b: 16
26	消息25	属性a: 7	属性b: 13
27	消息26	属性a: 7	属性b: 0
28	消息27	属性a: 7	属性b: 0
29	消息28	属性a: 8	属性b: 20
30	消息29	属性a: 10	属性b: 4
31	消息30	属性a: 2	属性b: 18
32	消息31	属性a: 6	属性b: 1
33	消息32	属性a: 10	属性b: 18
34	消息33	属性a: 3	属性b: 8
35	消息34	属性a: 9	属性b: 20
36	消息35	属性a: 4	属性b: 13
37	消息36	属性a: 1	属性b: 10
38	消息37	属性a: 1	属性b: 0
39	消息38	属性a: 8	属性b: 11
40	消息39	属性a: 7	属性b: 9
41	消息40	属性a: 1	属性b: 19
42	消息41	属性a: 3	属性b: 18
43	消息42	属性a: 5	属性b: 4
44	消息43	属性a: 7	属性b: 19
45	消息44	属性a: 0	属性b: 14
46	消息45	属性a: 6	属性b: 16
47	消息46	属性a: 5	属性b: 4
48	消息47	属性a: 2	属性b: 6
49	消息48	属性a: 3	属性b: 2
50	消息49	属性a: 3	属性b: 4
51	消息50	属性a: 8	属性b: 8
52	消息51	属性a: 0	属性b: 12
53	消息52	属性a: 8	属性b: 20
54	消息53	属性a: 0	属性b: 7
55	消息54	属性a: 1	属性b: 1
56	消息55	属性a: 1	属性b: 0
57	消息56	属性a: 7	属性b: 1
58	消息57	属性a: 3	属性b: 2
59	消息58	属性a: 5	属性b: 11
60	消息59	属性a: 4	属性b: 19
61	消息60	属性a: 6	属性b: 0
62	消息61	属性a: 9	属性b: 18
63	消息62	属性a: 4	属性b: 4
64	消息63	属性a: 4	属性b: 15
65	消息64	属性a: 5	属性b: 19
66	消息65	属性a: 1	属性b: 3
67	消息66	属性a: 8	属性b: 8
68	消息67	属性a: 7	属性b: 1
69	消息68	属性a: 2	属性b: 13
70	消息69	属性a: 0	属性b: 16
71	消息70	属性a: 1	属性b: 15
72	消息71	属性a: 8	属性b: 20
73	消息72	属性a: 7	属性b: 12
74	消息73	属性a: 7	属性b: 7
75	消息74	属性a: 2	属性b: 19

76	消息75	属性a: 10	属性b: 16
77	消息76	属性a: 8	属性b: 3
78	消息77	属性a: 8	属性b: 17
79	消息78	属性a: 7	属性b: 9
80	消息79	属性a: 1	属性b: 9
81	消息80	属性a: 4	属性b: 18
82	消息81	属性a: 9	属性b: 7
83	消息82	属性a: 7	属性b: 1
84	消息83	属性a: 6	属性b: 20
85	消息84	属性a: 7	属性b: 20
86	消息85	属性a: 3	属性b: 3
87	消息86	属性a: 2	属性b: 13
88	消息87	属性a: 6	属性b: 15
89	消息88	属性a: 3	属性b: 18
90	消息89	属性a: 6	属性b: 6
91	消息90	属性a: 2	属性b: 4
92	消息91	属性a: 1	属性b: 17
93	消息92	属性a: 4	属性b: 3
94	消息93	属性a: 3	属性b: 7
95	消息94	属性a: 4	属性b: 16
96	消息95	属性a: 10	属性b: 12
97	消息96	属性a: 4	属性b: 19
98	消息97	属性a: 2	属性b: 19
99	消息98	属性a: 0	属性b: 1
100	消息99	属性a: 0	属性b: 12

消费者4打印的信息

1	消息0	属性a: 8	属性b: 1
2	消息1	属性a: 3	属性b: 5
3	消息2	属性a: 9	属性b: 2
4	消息3	属性a: 6	属性b: 5
5	消息4	属性a: 8	属性b: 1
6	消息5	属性a: 8	属性b: 12
7	消息6	属性a: 1	属性b: 0
8	消息7	属性a: 7	属性b: 14
9	消息9	属性a: 8	属性b: 20
10	消息11	属性a: 8	属性b: 20
11	消息12	属性a: 1	属性b: 1
12	消息14	属性a: 3	属性b: 2
13	消息16	属性a: 3	属性b: 3
14	消息20	属性a: 3	属性b: 0
15	消息21	属性a: 9	属性b: 2
16	消息23	属性a: 9	属性b: 10
17	消息24	属性a: 8	属性b: 16
18	消息25	属性a: 7	属性b: 13
19	消息26	属性a: 7	属性b: 0
20	消息27	属性a: 7	属性b: 0
21	消息28	属性a: 8	属性b: 20
22	消息29	属性a: 10	属性b: 4
23	消息31	属性a: 6	属性b: 1
24	消息32	属性a: 10	属性b: 18
25	消息34	属性a: 9	属性b: 20
26	消息37	属性a: 1	属性b: 0
27	消息38	属性a: 8	属性b: 11

28	消息39	属性a: 7	属性b: 9
29	消息42	属性a: 5	属性b: 4
30	消息43	属性a: 7	属性b: 19
31	消息46	属性a: 5	属性b: 4
32	消息48	属性a: 3	属性b: 2
33	消息49	属性a: 3	属性b: 4
34	消息50	属性a: 8	属性b: 8
35	消息52	属性a: 8	属性b: 20
36	消息54	属性a: 1	属性b: 1
37	消息55	属性a: 1	属性b: 0
38	消息56	属性a: 7	属性b: 1
39	消息57	属性a: 3	属性b: 2
40	消息60	属性a: 6	属性b: 0
41	消息61	属性a: 9	属性b: 18
42	消息62	属性a: 4	属性b: 4
43	消息65	属性a: 1	属性b: 3
44	消息66	属性a: 8	属性b: 8
45	消息67	属性a: 7	属性b: 1
46	消息71	属性a: 8	属性b: 20
47	消息72	属性a: 7	属性b: 12
48	消息73	属性a: 7	属性b: 7
49	消息75	属性a: 10	属性b: 16
50	消息76	属性a: 8	属性b: 3
51	消息77	属性a: 8	属性b: 17
52	消息78	属性a: 7	属性b: 9
53	消息81	属性a: 9	属性b: 7
54	消息82	属性a: 7	属性b: 1
55	消息84	属性a: 7	属性b: 20
56	消息85	属性a: 3	属性b: 3
57	消息90	属性a: 2	属性b: 4
58	消息92	属性a: 4	属性b: 3
59	消息95	属性a: 10	属性b: 12
60	消息98	属性a: 0	属性b: 1

关闭消费者4，启动消费者5，再启动生产者2

生产者2打印的信息

1	消息0	属性a: 8	属性b: 19
2	消息1	属性a: 1	属性b: 4
3	消息2	属性a: 2	属性b: 17
4	消息3	属性a: 0	属性b: 0
5	消息4	属性a: 10	属性b: 14
6	消息5	属性a: 4	属性b: 18
7	消息6	属性a: 2	属性b: 13
8	消息7	属性a: 6	属性b: 20
9	消息8	属性a: 4	属性b: 7
10	消息9	属性a: 1	属性b: 15
11	消息10	属性a: 10	属性b: 12
12	消息11	属性a: 3	属性b: 7
13	消息12	属性a: 8	属性b: 8

14	消息13	属性a: 7	属性b: 5
15	消息14	属性a: 0	属性b: 15
16	消息15	属性a: 10	属性b: 17
17	消息16	属性a: 1	属性b: 16
18	消息17	属性a: 2	属性b: 14
19	消息18	属性a: 0	属性b: 20
20	消息19	属性a: 7	属性b: 3
21	消息20	属性a: 0	属性b: 1
22	消息21	属性a: 8	属性b: 12
23	消息22	属性a: 6	属性b: 12
24	消息23	属性a: 2	属性b: 17
25	消息24	属性a: 5	属性b: 1
26	消息25	属性a: 4	属性b: 17
27	消息26	属性a: 3	属性b: 9
28	消息27	属性a: 5	属性b: 15
29	消息28	属性a: 6	属性b: 5
30	消息29	属性a: 9	属性b: 14
31	消息30	属性a: 5	属性b: 3
32	消息31	属性a: 8	属性b: 13
33	消息32	属性a: 6	属性b: 7
34	消息33	属性a: 2	属性b: 20
35	消息34	属性a: 6	属性b: 12
36	消息35	属性a: 8	属性b: 17
37	消息36	属性a: 6	属性b: 0
38	消息37	属性a: 9	属性b: 19
39	消息38	属性a: 7	属性b: 5
40	消息39	属性a: 4	属性b: 4
41	消息40	属性a: 0	属性b: 3
42	消息41	属性a: 1	属性b: 3
43	消息42	属性a: 0	属性b: 13
44	消息43	属性a: 1	属性b: 11
45	消息44	属性a: 8	属性b: 3
46	消息45	属性a: 7	属性b: 1
47	消息46	属性a: 6	属性b: 5
48	消息47	属性a: 3	属性b: 2
49	消息48	属性a: 1	属性b: 18
50	消息49	属性a: 8	属性b: 3
51	消息50	属性a: 8	属性b: 14
52	消息51	属性a: 0	属性b: 8
53	消息52	属性a: 9	属性b: 5
54	消息53	属性a: 1	属性b: 10
55	消息54	属性a: 7	属性b: 6
56	消息55	属性a: 3	属性b: 15
57	消息56	属性a: 8	属性b: 10
58	消息57	属性a: 4	属性b: 0
59	消息58	属性a: 9	属性b: 13
60	消息59	属性a: 7	属性b: 8
61	消息60	属性a: 4	属性b: 10
62	消息61	属性a: 10	属性b: 14
63	消息62	属性a: 6	属性b: 13
64	消息63	属性a: 1	属性b: 6
65	消息64	属性a: 3	属性b: 6
66	消息65	属性a: 10	属性b: 4
67	消息66	属性a: 3	属性b: 11
68	消息67	属性a: 3	属性b: 14
69	消息68	属性a: 5	属性b: 19
70	消息69	属性a: 2	属性b: 0
71	消息70	属性a: 1	属性b: 7

72	消息71	属性a: 10	属性b: 19
73	消息72	属性a: 3	属性b: 11
74	消息73	属性a: 9	属性b: 8
75	消息74	属性a: 3	属性b: 11
76	消息75	属性a: 7	属性b: 15
77	消息76	属性a: 2	属性b: 15
78	消息77	属性a: 7	属性b: 18
79	消息78	属性a: 8	属性b: 10
80	消息79	属性a: 9	属性b: 11
81	消息80	属性a: 10	属性b: 17
82	消息81	属性a: 5	属性b: 0
83	消息82	属性a: 9	属性b: 12
84	消息83	属性a: 2	属性b: 5
85	消息84	属性a: 6	属性b: 4
86	消息85	属性a: 4	属性b: 3
87	消息86	属性a: 0	属性b: 13
88	消息87	属性a: 1	属性b: 14
89	消息88	属性a: 1	属性b: 19
90	消息89	属性a: 0	属性b: 11
91	消息90	属性a: 3	属性b: 10
92	消息91	属性a: 5	属性b: 19
93	消息92	属性a: 9	属性b: 5
94	消息93	属性a: 9	属性b: 11
95	消息94	属性a: 0	属性b: 14
96	消息95	属性a: 6	属性b: 8
97	消息96	属性a: 10	属性b: 4
98	消息97	属性a: 1	属性b: 7
99	消息98	属性a: 10	属性b: 19
100	消息99	属性a: 1	属性b: 11

消费者5打印的信息

1	消息2	属性a: 2	属性b: 17
2	消息5	属性a: 4	属性b: 18
3	消息6	属性a: 2	属性b: 13
4	消息7	属性a: 6	属性b: 20
5	消息9	属性a: 1	属性b: 15
6	消息14	属性a: 0	属性b: 15
7	消息16	属性a: 1	属性b: 16
8	消息17	属性a: 2	属性b: 14
9	消息18	属性a: 0	属性b: 20
10	消息22	属性a: 6	属性b: 12
11	消息23	属性a: 2	属性b: 17
12	消息25	属性a: 4	属性b: 17
13	消息27	属性a: 5	属性b: 15
14	消息33	属性a: 2	属性b: 20
15	消息34	属性a: 6	属性b: 12
16	消息42	属性a: 0	属性b: 13
17	消息48	属性a: 1	属性b: 18
18	消息55	属性a: 3	属性b: 15
19	消息62	属性a: 6	属性b: 13
20	消息67	属性a: 3	属性b: 14
21	消息68	属性a: 5	属性b: 19
22	消息76	属性a: 2	属性b: 15

23	消息86	属性a: 0	属性b: 13
24	消息87	属性a: 1	属性b: 14
25	消息88	属性a: 1	属性b: 19
26	消息91	属性a: 5	属性b: 19
27	消息94	属性a: 0	属性b: 14

关闭消费者5，启动消费者6，再启动生产者2

生产者2打印的信息

1	消息0	属性a: 0	属性b: 17
2	消息1	属性a: 2	属性b: 13
3	消息2	属性a: 2	属性b: 5
4	消息3	属性a: 10	属性b: 19
5	消息4	属性a: 6	属性b: 11
6	消息5	属性a: 10	属性b: 11
7	消息6	属性a: 5	属性b: 14
8	消息7	属性a: 2	属性b: 15
9	消息8	属性a: 10	属性b: 9
10	消息9	属性a: 1	属性b: 14
11	消息10	属性a: 10	属性b: 12
12	消息11	属性a: 2	属性b: 9
13	消息12	属性a: 1	属性b: 2
14	消息13	属性a: 7	属性b: 6
15	消息14	属性a: 6	属性b: 5
16	消息15	属性a: 6	属性b: 17
17	消息16	属性a: 6	属性b: 9
18	消息17	属性a: 4	属性b: 13
19	消息18	属性a: 7	属性b: 11
20	消息19	属性a: 10	属性b: 18
21	消息20	属性a: 4	属性b: 3
22	消息21	属性a: 5	属性b: 17
23	消息22	属性a: 10	属性b: 2
24	消息23	属性a: 4	属性b: 12
25	消息24	属性a: 8	属性b: 12
26	消息25	属性a: 0	属性b: 18
27	消息26	属性a: 7	属性b: 3
28	消息27	属性a: 3	属性b: 15
29	消息28	属性a: 10	属性b: 19
30	消息29	属性a: 5	属性b: 1
31	消息30	属性a: 10	属性b: 10
32	消息31	属性a: 9	属性b: 2
33	消息32	属性a: 1	属性b: 5
34	消息33	属性a: 10	属性b: 3
35	消息34	属性a: 6	属性b: 5
36	消息35	属性a: 5	属性b: 5
37	消息36	属性a: 0	属性b: 8
38	消息37	属性a: 2	属性b: 4
39	消息38	属性a: 7	属性b: 0
40	消息39	属性a: 8	属性b: 15
41	消息40	属性a: 1	属性b: 5

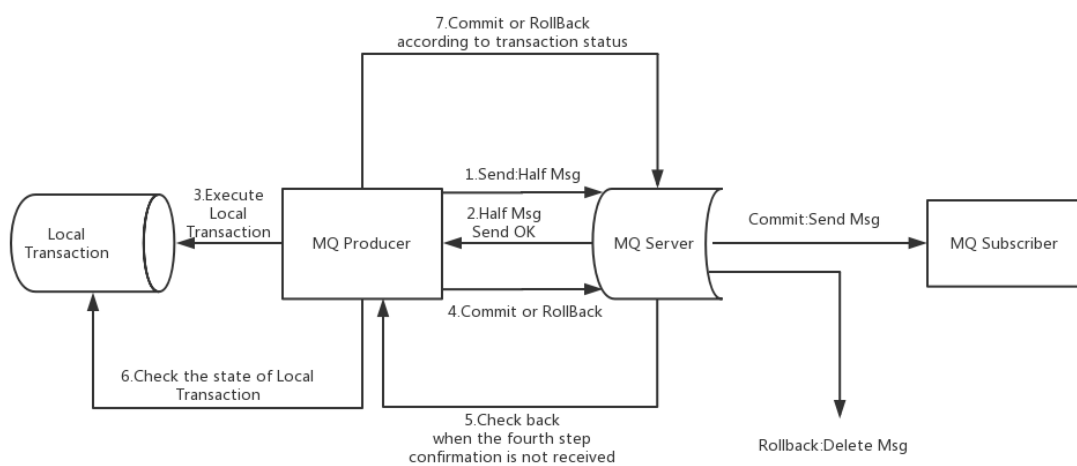
42	消息41	属性a: 9	属性b: 2
43	消息42	属性a: 10	属性b: 2
44	消息43	属性a: 7	属性b: 14
45	消息44	属性a: 9	属性b: 17
46	消息45	属性a: 8	属性b: 1
47	消息46	属性a: 4	属性b: 13
48	消息47	属性a: 9	属性b: 9
49	消息48	属性a: 7	属性b: 5
50	消息49	属性a: 7	属性b: 3
51	消息50	属性a: 8	属性b: 17
52	消息51	属性a: 4	属性b: 10
53	消息52	属性a: 1	属性b: 5
54	消息53	属性a: 3	属性b: 9
55	消息54	属性a: 7	属性b: 9
56	消息55	属性a: 7	属性b: 12
57	消息56	属性a: 2	属性b: 13
58	消息57	属性a: 7	属性b: 18
59	消息58	属性a: 10	属性b: 3
60	消息59	属性a: 9	属性b: 20
61	消息60	属性a: 6	属性b: 9
62	消息61	属性a: 3	属性b: 20
63	消息62	属性a: 2	属性b: 11
64	消息63	属性a: 4	属性b: 16
65	消息64	属性a: 3	属性b: 4
66	消息65	属性a: 7	属性b: 20
67	消息66	属性a: 4	属性b: 0
68	消息67	属性a: 1	属性b: 10
69	消息68	属性a: 2	属性b: 9
70	消息69	属性a: 1	属性b: 19
71	消息70	属性a: 3	属性b: 10
72	消息71	属性a: 1	属性b: 7
73	消息72	属性a: 5	属性b: 5
74	消息73	属性a: 9	属性b: 10
75	消息74	属性a: 10	属性b: 3
76	消息75	属性a: 4	属性b: 11
77	消息76	属性a: 1	属性b: 1
78	消息77	属性a: 1	属性b: 11
79	消息78	属性a: 2	属性b: 5
80	消息79	属性a: 1	属性b: 3
81	消息80	属性a: 10	属性b: 3
82	消息81	属性a: 10	属性b: 19
83	消息82	属性a: 3	属性b: 17
84	消息83	属性a: 7	属性b: 17
85	消息84	属性a: 5	属性b: 15
86	消息85	属性a: 2	属性b: 1
87	消息86	属性a: 0	属性b: 15
88	消息87	属性a: 9	属性b: 10
89	消息88	属性a: 9	属性b: 0
90	消息89	属性a: 7	属性b: 2
91	消息90	属性a: 4	属性b: 15
92	消息91	属性a: 5	属性b: 17
93	消息92	属性a: 8	属性b: 3
94	消息93	属性a: 9	属性b: 17
95	消息94	属性a: 7	属性b: 2
96	消息95	属性a: 10	属性b: 14
97	消息96	属性a: 6	属性b: 1
98	消息97	属性a: 9	属性b: 6
99	消息98	属性a: 8	属性b: 8

消费者6打印的信息

1	消息27	属性a: 3	属性b: 15
2	消息30	属性a: 10	属性b: 10
3	消息51	属性a: 4	属性b: 10
4	消息53	属性a: 3	属性b: 9
5	消息61	属性a: 3	属性b: 20
6	消息64	属性a: 3	属性b: 4
7	消息67	属性a: 1	属性b: 10
8	消息70	属性a: 3	属性b: 10
9	消息73	属性a: 9	属性b: 10
10	消息82	属性a: 3	属性b: 17
11	消息87	属性a: 9	属性b: 10

事务消息

流程分析



两个流程：正常事务消息的发送及提交、事务消息的补偿流程。

事务消息发送及提交

- (1) 发送消息（half消息）。
- (2) 服务端响应消息写入结果。
- (3) 根据发送结果执行本地事务（如果写入失败，此时half消息对业务不可见，本地逻辑不执行）。
- (4) 根据本地事务状态执行Commit或者Rollback（Commit操作生成消息索引，消息对消费者可见）

事务补偿

- (1) 对没有Commit/Rollback的事务消息（pending状态的消息），从服务端发起一次“回查”
- (2) Producer收到回查消息，检查回查消息对应的本地事务的状态
- (3) 根据本地事务状态，重新Commit或者Rollback

其中，补偿阶段用于解决消息Commit或者Rollback发生超时或者失败的情况。

事务消息状态

事务消息共有三种状态，提交状态、回滚状态、中间状态：

- TransactionStatus.CommitTransaction: 提交事务，它允许消费者消费此消息。
- TransactionStatus.RollbackTransaction: 回滚事务，它代表该消息将被删除，不允许被消费。
- TransactionStatus.Unknown: 中间状态，它代表需要检查消息队列来确定状态。

发送事务消息

创建事务性生产者

使用 `TransactionMQProducer` 类创建生产者，并指定唯一的 `ProducerGroup`，就可以设置自定义线程池来处理这些检查请求。执行本地事务后、需要根据执行结果对消息队列进行回复

```
1 package mao.producer;
2
3 import org.apache.rocketmq.client.exception.MQClientException;
4 import org.apache.rocketmq.client.producer.TransactionListener;
5 import org.apache.rocketmq.client.producer.TransactionMQProducer;
6 import org.apache.rocketmq.client.producer.TransactionSendResult;
7 import org.apache.rocketmq.common.message.Message;
8
9 import java.nio.charset.StandardCharsets;
```

```

10
11 /**
12  * Project name(项目名称): RocketMQ_事务消息的发送与接收
13  * Package(包名): mao.producer
14  * Class(类名): TransactionalProducer
15  * Author(作者): mao
16  * Author QQ: 1296193245
17  * GitHub: https://github.com/maomao124/
18  * Date(创建日期): 2022/12/10
19  * Time(创建时间): 15:53
20  * Version(版本): 1.0
21  * Description(描述): 生产者-事务消息
22  */
23
24 public class TransactionalProducer
25 {
26     /**
27      * 标签
28      */
29     private static final String[] tags = new String[]{"TagA", "TagB",
"TagC"};
30
31     public static void main(String[] args) throws MQClientException,
InterruptedException
32     {
33         //生产者
34         TransactionMQProducer transactionMQProducer = new
TransactionalMQProducer("mao_group");
35         //事务监听器
36         TransactionListener transactionListener = new
TransactionalListenerImpl();
37         //设置nameserver地址
38         transactionMQProducer.setNamesrvAddr("127.0.0.1:9876");
39         //设置监听器
40         transactionMQProducer.setTransactionListener(transactionListener);
41         //启动生产者
42         transactionMQProducer.start();
43         //发送30条消息
44         for (int i = 0; i < 30; i++)
45         {
46             try
47             {
48                 //标签
49                 String tag = tags[i % (tags.length)];
50                 //消息体
51                 String body = "标签: " + tag + " 消息: " + i;
52                 //消息对象
53                 Message message = new Message("test_topic", tag,
body.getBytes(StandardCharsets.UTF_8));
54                 //发送事务消息
55                 TransactionSendResult transactionSendResult =
transactionMQProducer.sendMessageInTransaction(message, null);
56                 Thread.sleep(500);
57                 //打印
58                 System.out.println(body);
59             }
60             catch (Exception e)
61             {

```



```

62         e.printStackTrace();
63     }
64 }
65 }
66 }

```

实现事务的监听接口

```

1  package mao.producer;
2
3  import org.apache.commons.lang3.StringUtils;
4  import org.apache.rocketmq.client.producer.LocalTransactionState;
5  import org.apache.rocketmq.client.producer.TransactionListener;
6  import org.apache.rocketmq.common.message.Message;
7  import org.apache.rocketmq.common.message.MessageExt;
8
9  /**
10   * Project name(项目名称): RocketMQ_事务消息的发送与接收
11   * Package(包名): mao.producer
12   * Class(类名): TransactionListenerImpl
13   * Author(作者): mao
14   * Author QQ: 1296193245
15   * Github: https://github.com/maomao124/
16   * Date(创建日期): 2022/12/10
17   * Time(创建时间): 16:01
18   * Version(版本): 1.0
19   * Description(描述): 事务监听器实现
20   */
21
22  public class TransactionListenerImpl implements TransactionListener
23  {
24      /**
25       * 执行本地事务
26       *
27       * @param msg 消息
28       * @param arg 参数
29       * @return {@link LocalTransactionState}
30       */
31      @Override
32      public LocalTransactionState executeLocalTransaction(Message msg, Object
arg)
33      {
34          System.out.println("执行本地事务");
35          if (StringUtils.equals("TagA", msg.getTags()))
36          {
37              //如果是标签a的消息, 直接提交
38              return LocalTransactionState.COMMIT_MESSAGE;
39          }
40          else if (StringUtils.equals("TagB", msg.getTags()))
41          {
42              //如果是标签B的消息, 回滚
43              return LocalTransactionState.ROLLBACK_MESSAGE;

```

```

44         }
45         else
46         {
47             //否则，不知道，中间状态
48             return LocalTransactionState.UNKNOW;
49         }
50
51     }
52
53     /**
54      * 检查本地事务
55      *
56      * @param msg 消息
57      * @return {@link LocalTransactionState}
58      */
59     @Override
60     public LocalTransactionState checkLocalTransaction(MessageExt msg)
61     {
62         System.out.println("MQ检查消息Tag【" + msg.getTags() + "】的本地事务执行
结果");
63         return LocalTransactionState.COMMIT_MESSAGE;
64     }
65 }

```

接收事务消息

```

1  package mao.consumer;
2
3  import org.apache.rocketmq.client.consumer.DefaultMQPushConsumer;
4  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyContext;
5  import
org.apache.rocketmq.client.consumer.listener.ConsumeConcurrentlyStatus;
6  import
org.apache.rocketmq.client.consumer.listener.MessageListenerConcurrently;
7  import org.apache.rocketmq.client.exception.MQClientException;
8  import org.apache.rocketmq.common.message.MessageExt;
9
10 import java.nio.charset.StandardCharsets;
11 import java.util.List;
12
13 /**
14  * Project name(项目名称): RocketMQ_事务消息的发送与接收
15  * Package(包名): mao.consumer
16  * Class(类名): Consumer
17  * Author(作者): mao
18  * Author QQ: 1296193245
19  * GitHub: https://github.com/maomao124/
20  * Date(创建日期): 2022/12/10
21  * Time(创建时间): 16:11
22  * Version(版本): 1.0

```

```

23  * Description(描述): 消息消费者
24  */
25
26  public class Consumer
27  {
28      public static void main(String[] args) throws MQClientException
29      {
30          //消费者
31          DefaultMQPushConsumer defaultMQPushConsumer = new
DefaultMQPushConsumer("mao_group");
32          //nameserver
33          defaultMQPushConsumer.setNamesrvAddr("127.0.0.1:9876");
34          //订阅
35          defaultMQPushConsumer.subscribe("test_topic", "*");
36          //监听器
37          defaultMQPushConsumer.registerMessageListener(new
MessageListenerConcurrently()
38          {
39              @Override
40              public ConsumeConcurrentlyStatus consumeMessage(List<MessageExt>
list, ConsumeConcurrentlyContext consumeConcurrentlyContext)
41              {
42                  for (MessageExt messageExt : list)
43                  {
44                      System.out.println(new String(messageExt.getBody(),
StandardCharsets.UTF_8));
45                  }
46                  //返回成功
47                  return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
48              }
49          });
50          //启动
51          defaultMQPushConsumer.start();
52      }
53  }

```

启动测试

先启动消费者，再启动生产者

等待一段时间

生产者打印的信息

```

1  执行本地事务
2  标签: TagA   消息: 0
3  执行本地事务
4  标签: TagB   消息: 1
5  执行本地事务
6  标签: TagC   消息: 2

```

7	执行本地事务	
8	标签: TagA	消息: 3
9	执行本地事务	
10	标签: TagB	消息: 4
11	执行本地事务	
12	标签: TagC	消息: 5
13	执行本地事务	
14	标签: TagA	消息: 6
15	执行本地事务	
16	标签: TagB	消息: 7
17	执行本地事务	
18	标签: TagC	消息: 8
19	执行本地事务	
20	标签: TagA	消息: 9
21	执行本地事务	
22	标签: TagB	消息: 10
23	执行本地事务	
24	标签: TagC	消息: 11
25	执行本地事务	
26	标签: TagA	消息: 12
27	执行本地事务	
28	标签: TagB	消息: 13
29	执行本地事务	
30	标签: TagC	消息: 14
31	执行本地事务	
32	标签: TagA	消息: 15
33	执行本地事务	
34	标签: TagB	消息: 16
35	执行本地事务	
36	标签: TagC	消息: 17
37	执行本地事务	
38	标签: TagA	消息: 18
39	执行本地事务	
40	标签: TagB	消息: 19
41	执行本地事务	
42	标签: TagC	消息: 20
43	执行本地事务	
44	标签: TagA	消息: 21
45	执行本地事务	
46	标签: TagB	消息: 22
47	执行本地事务	
48	标签: TagC	消息: 23
49	执行本地事务	
50	标签: TagA	消息: 24
51	执行本地事务	
52	标签: TagB	消息: 25
53	执行本地事务	
54	标签: TagC	消息: 26
55	执行本地事务	
56	标签: TagA	消息: 27
57	执行本地事务	
58	标签: TagB	消息: 28
59	执行本地事务	
60	标签: TagC	消息: 29
61	MQ检查消息Tag【TagC】的本地事务执行结果	
62	MQ检查消息Tag【TagC】的本地事务执行结果	
63	MQ检查消息Tag【TagC】的本地事务执行结果	
64	MQ检查消息Tag【TagC】的本地事务执行结果	

```
65 MQ检查消息Tag【TagC】的本地事务执行结果
66 MQ检查消息Tag【TagC】的本地事务执行结果
67 MQ检查消息Tag【TagC】的本地事务执行结果
68 MQ检查消息Tag【TagC】的本地事务执行结果
69 MQ检查消息Tag【TagC】的本地事务执行结果
70 MQ检查消息Tag【TagC】的本地事务执行结果
```

消费者打印的信息

```
1  标签: TagA    消息: 0
2  标签: TagA    消息: 3
3  标签: TagA    消息: 6
4  标签: TagA    消息: 9
5  标签: TagA    消息: 12
6  标签: TagA    消息: 15
7  标签: TagA    消息: 18
8  标签: TagA    消息: 21
9  标签: TagA    消息: 24
10 标签: TagA    消息: 27
11 标签: TagC    消息: 11
12 标签: TagC    消息: 8
13 标签: TagC    消息: 5
14 标签: TagC    消息: 2
15 标签: TagC    消息: 14
16 标签: TagC    消息: 23
17 标签: TagC    消息: 17
18 标签: TagC    消息: 29
19 标签: TagC    消息: 20
20 标签: TagC    消息: 26
```

使用限制

1. 事务消息不支持延时消息和批量消息。
2. 为了避免单个消息被检查太多次而导致半队列消息累积，我们默认将单个消息的检查次数限制为 15 次，但是用户可以通过 Broker 配置文件的 `transactionCheckMax` 参数来修改此限制。如果已经检查某条消息超过 N 次的话（`N = transactionCheckMax`）则 Broker 将丢弃此消息，并在默认情况下同时打印错误日志。用户可以通过重写 `AbstractTransactionCheckListener` 类来修改这个行为。
3. 事务消息将在 Broker 配置文件中的参数 `transactionMsgTimeout` 这样的特定时间长度之后被检查。当发送事务消息时，用户还可以通过设置用户属性 `CHECK_IMMUNITY_TIME_IN_SECONDS` 来改变这个限制，该参数优先于 `transactionMsgTimeout` 参数。
4. 事务性消息可能不止一次被检查或消费。

5. 提交给用户的目标主题消息可能会失败，目前这依日志的记录而定。它的高可用性通过 RocketMQ 本身的高可用性机制来保证，如果希望确保事务消息不丢失、并且事务完整性得到保证，建议使用同步的双重写入机制。
6. 事务消息的生产者 ID 不能与其他类型消息的生产者 ID 共享。与其他类型的消息不同，事务消息允许反向查询、MQ 服务器能通过它们的生产者 ID 查询到消费者。

整合spring boot

依赖

```
1      <!--RocketMQ spring boot 依赖-->
2      <dependency>
3          <groupId>org.apache.rocketmq</groupId>
4          <artifactId>rocketmq-spring-boot-starter</artifactId>
5          <version>2.2.2</version>
6      </dependency>
```

Yml文件配置

```
1  rocketmq:
2    name-server: 127.0.0.1:9876
3    producer:
4      # 生产者分组
5      group: mao_group
6      # 发送消息超时时间，单位：毫秒。默认为 3000
7      send-message-timeout: 3000
8      # 消息压缩阈值，当消息体的大小超过该阈值后，进行消息压缩。默认为 4 * 1024B
9      compress-message-body-threshold: 4096
10     # 消息体的最大允许大小。。默认为 4 * 1024 * 1024B
11     max-message-size: 4194304
12     # 同步发送消息时，失败重试次数。默认为 2 次。
13     retry-times-when-send-failed: 2
14     # 异步发送消息时，失败重试次数。默认为 2 次。
```

```

15     retry-times-when-send-async-failed: 2
16     # 发送消息给 Broker 时，如果发送失败，是否重试另外一台 Broker 。默认为 false
17     retry-next-server: false
18     # Access Key ，可阅读
    https://github.com/apache/rocketmq/blob/master/docs/cn/acl/user_guide.md 文档
19     access-key:
20     # Secret Key
21     secret-key:
22     # 是否开启消息轨迹功能。默认为 true 开启。可阅读
    https://github.com/apache/rocketmq/blob/master/docs/cn/msg_trace/user_guide.
    md 文档
23     enable-msg-trace: true
24     # 自定义消息轨迹的 Topic 。默认为 RMQ_SYS_TRACE_TOPIC
25     customized-trace-topic: RMQ_SYS_TRACE_TOPIC
26     #consumer:
27     # 配置某个消费分组，是否监听指定 Topic 。结构为 Map<消费者分组, <Topic,
    Boolean>> 。默认情况下，不配置表示监听。
28     #listeners:
29     #erbadagang-consumer-group:
30     # 关闭 test-consumer-group 对 topic1 的监听消费
31     #topic1: false

```

消息发送

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.spring.core.RocketMQTemplate;
4 import org.slf4j.Logger;
5 import org.slf4j.LoggerFactory;
6 import org.springframework.web.bind.annotation.GetMapping;
7 import org.springframework.web.bind.annotation.PathVariable;
8 import org.springframework.web.bind.annotation.RestController;
9
10 import javax.annotation.Resource;
11
12 /**
13  * Project name(项目名称): rocketMQ_spring_boot
14  * Package(包名): mao.rocketmq_spring_boot.controller
15  * Class(类名): RocketController
16  * Author(作者): mao
17  * Author QQ: 1296193245
18  * GitHub: https://github.com/maomao124/
19  * Date(创建日期): 2022/12/11
20  * Time(创建时间): 18:29
21  * Version(版本): 1.0
22  * Description(描述): 无
23  */
24
25 @RestController

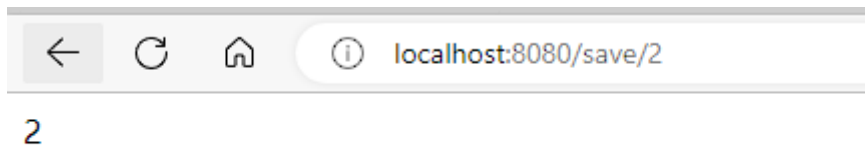
```



```

15 2022-12-11 18:40:56.989 INFO 13960 --- [          main]
    w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext:
    initialization completed in 841 ms
16 2022-12-11 18:40:57.477 INFO 13960 --- [          main]
    o.s.b.w.embedded.tomcat.TomcatWebServer  : Tomcat started on port(s): 8080
    (http) with context path ''
17 2022-12-11 18:40:57.489 INFO 13960 --- [          main]
    m.r.RocketMqSpringBootApplication       : Started
    RocketMqSpringBootApplication in 1.696 seconds (JVM running for 2.326)
18 2022-12-11 18:41:01.210 INFO 13960 --- [nio-8080-exec-1] o.a.c.c.C.
    [Tomcat].[localhost].[/]               : Initializing Spring DispatcherServlet
    'dispatcherServlet'
19 2022-12-11 18:41:01.210 INFO 13960 --- [nio-8080-exec-1]
    o.s.web.servlet.DispatcherServlet       : Initializing Servlet
    'dispatcherServlet'
20 2022-12-11 18:41:01.211 INFO 13960 --- [nio-8080-exec-1]
    o.s.web.servlet.DispatcherServlet       : Completed initialization in 1 ms
21 2022-12-11 18:41:01.232 INFO 13960 --- [nio-8080-exec-1]
    m.r.controller.RocketController         : 开始发送消息
22

```



发送同步消息

API

```

1  //发送普通同步消息-Object
2  syncSend(String destination, Object payload);
3
4  //发送普通同步消息-Message
5  syncSend(String destination, Message<?> message);
6
7  //发送批量普通同步消息
8  syncSend(String destination, Collection<T> messages);
9
10 //发送普通同步消息-Object，并设置发送超时时间
11 syncSend(String destination, Object payload, long timeout);
12
13 //发送普通同步消息-Message，并设置发送超时时间
14 syncSend(String destination, Message<?> message, long timeout);
15

```

```

16 //发送批量普通同步消息，并设置发送超时时间
17 syncSend(String destination, Collection<T> messages, long timeout);
18
19 //发送普通同步延迟消息，并设置超时
20 syncSend(String destination, Message<?> message, long timeout, int
    delayLevel);

```

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.client.producer.SendResult;
4 import org.apache.rocketmq.common.message.Message;
5 import org.apache.rocketmq.spring.core.RocketMQTemplate;
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8 import org.springframework.web.bind.annotation.GetMapping;
9 import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import javax.annotation.Resource;
13 import java.nio.charset.StandardCharsets;
14
15 /**
16  * Project name(项目名称): rocketMQ_spring_boot
17  * Package(包名): mao.rocketmq_spring_boot.controller
18  * Class(类名): RocketController
19  * Author(作者): mao
20  * Author QQ: 1296193245
21  * GitHub: https://github.com/maomao124/
22  * Date(创建日期): 2022/12/11
23  * Time(创建时间): 18:29
24  * Version(版本): 1.0
25  * Description(描述): 无
26  */
27
28 @RestController
29 public class RocketController
30 {
31     /**
32      * 日志
33      */
34     private static final Logger log =
35         LoggerFactory.getLogger(RocketController.class);
36
37     @Resource
38     private RocketMQTemplate rocketMQTemplate;
39
40     /**
41      * 发送消息
42      *
43      * @param data 数据
44      * @return {@link String}
45      */
46     @GetMapping("/save/{data}")

```

```

46     public String save(@PathVariable String data)
47     {
48         log.info("开始发送消息");
49         rocketMQTemplate.convertAndSend("test_topic", data);
50         return data;
51     }
52
53     /**
54      * 发送同步消息
55      *
56      * @param data 数据
57      * @return {@link SendResult}
58      */
59     @GetMapping("/save2/{data}")
60     public SendResult save2(@PathVariable String data)
61     {
62         log.info("开始发送同步消息");
63         Message message = new Message();
64         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
65         return rocketMQTemplate.syncSend("test_topic", message);
66     }
67
68 }

```

```

1 2022-12-11 18:47:31.019 INFO 18040 --- [nio-8080-exec-1]
   m.r.controller.RocketController      : 开始发送同步消息

```

```

1 {"sendStatus":"SEND_OK","msgId":"7F000001467866D3C6173787DDFC0000","messageQu
   eue":{"topic":"test_topic","brokerName":"broker-
   a","queueId":1},"queueOffset":61957,"transactionId":null,"offsetMsgId":"AC17A
   00100002A9F0000000009A79F24","regionId":"DefaultRegion","traceOn":true}

```

```

1 {"sendStatus":"SEND_OK","msgId":"7F000001467866D3C617378862F40003","messageQu
   eue":{"topic":"test_topic","brokerName":"broker-
   a","queueId":2},"queueOffset":61869,"transactionId":null,"offsetMsgId":"AC17A
   00100002A9F0000000009A7A22E","regionId":"DefaultRegion","traceOn":true}

```

发送批量消息

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.client.producer.SendResult;
4 import org.apache.rocketmq.common.message.Message;

```

```

5  import org.apache.rocketmq.spring.core.RocketMQTemplate;
6  import org.slf4j.Logger;
7  import org.slf4j.LoggerFactory;
8  import org.springframework.web.bind.annotation.GetMapping;
9  import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import javax.annotation.Resource;
13 import java.nio.charset.StandardCharsets;
14 import java.util.ArrayList;
15 import java.util.List;
16
17 /**
18  * Project name(项目名称): rocketMQ_spring_boot
19  * Package(包名): mao.rocketmq_spring_boot.controller
20  * Class(类名): RocketController
21  * Author(作者): mao
22  * Author QQ: 1296193245
23  * GitHub: https://github.com/maomao124/
24  * Date(创建日期): 2022/12/11
25  * Time(创建时间): 18:29
26  * Version(版本): 1.0
27  * Description(描述): 无
28  */
29
30 @RestController
31 public class RocketController
32 {
33     /**
34      * 日志
35      */
36     private static final Logger log =
37         LoggerFactory.getLogger(RocketController.class);
38
39     @Resource
40     private RocketMQTemplate rocketMQTemplate;
41
42     /**
43      * 发送消息
44      *
45      * @param data 数据
46      * @return {@link String}
47      */
48     @GetMapping("/save/{data}")
49     public String save(@PathVariable String data)
50     {
51         log.info("开始发送消息");
52         rocketMQTemplate.convertAndSend("test_topic", data);
53         return data;
54     }
55
56     /**
57      * 发送同步消息
58      *
59      * @param data 数据
60      * @return {@link SendResult}
61      */
62     @GetMapping("/save2/{data}")

```

```

62     public SendResult save2(@PathVariable String data)
63     {
64         log.info("开始发送同步消息");
65         Message message = new Message();
66         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
67         return rocketMQTemplate.syncSend("test_topic", message);
68     }
69
70     /**
71      * 发送批量消息
72      *
73      * @param data 数据
74      * @return {@link SendResult}
75      */
76     @GetMapping("/save3/{data}")
77     public SendResult save3(@PathVariable String data)
78     {
79         log.info("开始发送批量消息");
80         List<Message> messageList = new ArrayList<>(10);
81         for (int i = 0; i < 10; i++)
82         {
83             Message message = new Message();
84             message.setBody(("同步消息:" + data + " -" +
i).getBytes(StandardCharsets.UTF_8));
85             messageList.add(message);
86         }
87         return rocketMQTemplate.syncSend("test_topic", messageList);
88     }
89
90 }

```

发送带tag的消息

```

1  package mao.rocketmq_spring_boot.controller;
2
3  import org.apache.rocketmq.client.producer.SendResult;
4  import org.apache.rocketmq.common.message.Message;
5  import org.apache.rocketmq.spring.core.RocketMQTemplate;
6  import org.slf4j.Logger;
7  import org.slf4j.LoggerFactory;
8  import org.springframework.web.bind.annotation.GetMapping;
9  import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import javax.annotation.Resource;
13 import java.nio.charset.StandardCharsets;
14 import java.util.ArrayList;
15 import java.util.List;

```

```

16
17 /**
18  * Project name(项目名称): rocketMQ_spring_boot
19  * Package(包名): mao.rocketmq_spring_boot.controller
20  * Class(类名): RocketController
21  * Author(作者): mao
22  * Author QQ: 1296193245
23  * Github: https://github.com/maomao124/
24  * Date(创建日期): 2022/12/11
25  * Time(创建时间): 18:29
26  * Version(版本): 1.0
27  * Description(描述): 无
28  */
29
30 @RestController
31 public class RocketController
32 {
33     /**
34      * 日志
35      */
36     private static final Logger log =
37         LoggerFactory.getLogger(RocketController.class);
38
39     @Resource
40     private RocketMQTemplate rocketMQTemplate;
41
42     /**
43      * 发送消息
44      *
45      * @param data 数据
46      * @return {@link String}
47      */
48     @GetMapping("/save/{data}")
49     public String save(@PathVariable String data)
50     {
51         log.info("开始发送消息");
52         rocketMQTemplate.convertAndSend("test_topic", data);
53         return data;
54     }
55
56     /**
57      * 发送同步消息
58      *
59      * @param data 数据
60      * @return {@link SendResult}
61      */
62     @GetMapping("/save2/{data}")
63     public SendResult save2(@PathVariable String data)
64     {
65         log.info("开始发送同步消息");
66         Message message = new Message();
67         message.setBody(("同步消息:" +
68             data).getBytes(StandardCharsets.UTF_8));
69         return rocketMQTemplate.syncSend("test_topic", message);
70     }
71
72     /**
73      * 发送批量消息

```

```

72     *
73     * @param data 数据
74     * @return {@link SendResult}
75     */
76     @GetMapping("/save3/{data}")
77     public SendResult save3(@PathVariable String data)
78     {
79         log.info("开始发送批量消息");
80         List<Message> messageList = new ArrayList<>(10);
81         for (int i = 0; i < 10; i++)
82         {
83             Message message = new Message();
84             message.setBody(("批量消息:" + data + " -" +
85 i).getBytes(StandardCharsets.UTF_8));
86             messageList.add(message);
87         }
88         return rocketMQTemplate.syncSend("test_topic", messageList);
89     }
90
91     /**
92     * 发送带tag的消息
93     *
94     * @param data 数据
95     * @return {@link String}
96     */
97     @GetMapping("/save4/{data}")
98     public String save4(@PathVariable String data)
99     {
100         log.info("开始发送带tag的消息");
101         //发送带tag的消息，直接在topic后面加上":tag"
102         rocketMQTemplate.convertAndSend("test_topic:tag1", data);
103         return data;
104     }
105 }

```

发送同步消息，并指定队列

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.client.producer.SendResult;
4 import org.apache.rocketmq.common.message.Message;
5 import org.apache.rocketmq.spring.core.RocketMQTemplate;
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8 import org.springframework.web.bind.annotation.GetMapping;

```

```

9  import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RestController;
11
12 import javax.annotation.Resource;
13 import java.nio.charset.StandardCharsets;
14 import java.util.ArrayList;
15 import java.util.List;
16
17 /**
18  * Project name(项目名称): rocketMQ_spring_boot
19  * Package(包名): mao.rocketmq_spring_boot.controller
20  * Class(类名): RocketController
21  * Author(作者): mao
22  * Author QQ: 1296193245
23  * Github: https://github.com/maomao124/
24  * Date(创建日期): 2022/12/11
25  * Time(创建时间): 18:29
26  * Version(版本): 1.0
27  * Description(描述): 无
28  */
29
30 @RestController
31 public class RocketController
32 {
33     /**
34      * 日志
35      */
36     private static final Logger log =
37         LoggerFactory.getLogger(RocketController.class);
38
39     @Resource
40     private RocketMQTemplate rocketMQTemplate;
41
42     /**
43      * 发送消息
44      *
45      * @param data 数据
46      * @return {@link String}
47      */
48     @GetMapping("/save/{data}")
49     public String save(@PathVariable String data)
50     {
51         log.info("开始发送消息");
52         rocketMQTemplate.convertAndSend("test_topic", data);
53         return data;
54     }
55
56     /**
57      * 发送同步消息
58      *
59      * @param data 数据
60      * @return {@link SendResult}
61      */
62     @GetMapping("/save2/{data}")
63     public SendResult save2(@PathVariable String data)
64     {
65         log.info("开始发送同步消息");
66         Message message = new Message();

```



```

66         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
67         return rocketMQTemplate.syncSend("test_topic", message);
68     }
69
70     /**
71      * 发送批量消息
72      *
73      * @param data 数据
74      * @return {@link SendResult}
75      */
76     @GetMapping("/save3/{data}")
77     public SendResult save3(@PathVariable String data)
78     {
79         log.info("开始发送批量消息");
80         List<Message> messageList = new ArrayList<>(10);
81         for (int i = 0; i < 10; i++)
82         {
83             Message message = new Message();
84             message.setBody(("批量消息:" + data + " -" +
i).getBytes(StandardCharsets.UTF_8));
85             messageList.add(message);
86         }
87         return rocketMQTemplate.syncSend("test_topic", messageList);
88     }
89
90
91     /**
92      * 发送带tag的消息
93      *
94      * @param data 数据
95      * @return {@link String}
96      */
97     @GetMapping("/save4/{data}")
98     public String save4(@PathVariable String data)
99     {
100         log.info("开始发送带tag的消息");
101         //发送带tag的消息，直接在topic后面加上":tag"
102         rocketMQTemplate.convertAndSend("test_topic:tag1", data);
103         return data;
104     }
105
106     /**
107      * 发送消息
108      *
109      * @param data 数据
110      * @return {@link String}
111      */
112     @GetMapping("/save5/{data}")
113     public String save5(@PathVariable String data)
114     {
115         log.info("开始发送消息");
116         Message message = new Message();
117         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
118         SendResult sendResult =
rocketMQTemplate.syncSendOrderly("test_topic", message, "1");
119         log.info("发送结果: " + sendResult);

```

```

120         return data;
121     }
122 }

```

```

1  2022-12-11 19:11:50.283 INFO 7936 --- [nio-8080-exec-1]
   m.r.controller.RocketController      : 开始发送消息
2  2022-12-11 19:11:50.372 INFO 7936 --- [nio-8080-exec-1]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E22400000,
   offsetMsgId=AC17A00100002A9F0000000009A7BD8F, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61959]
3  2022-12-11 19:11:52.012 INFO 7936 --- [nio-8080-exec-2]
   m.r.controller.RocketController      : 开始发送消息
4  2022-12-11 19:11:52.042 INFO 7936 --- [nio-8080-exec-2]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E28CD0003,
   offsetMsgId=AC17A00100002A9F0000000009A7C099, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61960]
5  2022-12-11 19:11:52.519 INFO 7936 --- [nio-8080-exec-3]
   m.r.controller.RocketController      : 开始发送消息
6  2022-12-11 19:11:52.525 INFO 7936 --- [nio-8080-exec-3]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E2AC70006,
   offsetMsgId=AC17A00100002A9F0000000009A7C3A3, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61961]
7  2022-12-11 19:11:58.421 INFO 7936 --- [nio-8080-exec-4]
   m.r.controller.RocketController      : 开始发送消息
8  2022-12-11 19:11:58.453 INFO 7936 --- [nio-8080-exec-4]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E41D50009,
   offsetMsgId=AC17A00100002A9F0000000009A7C6AC, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61962]
9  2022-12-11 19:11:58.879 INFO 7936 --- [nio-8080-exec-5]
   m.r.controller.RocketController      : 开始发送消息
10 2022-12-11 19:11:58.904 INFO 7936 --- [nio-8080-exec-5]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E439F000C,
   offsetMsgId=AC17A00100002A9F0000000009A7C9B6, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61963]
11 2022-12-11 19:11:59.205 INFO 7936 --- [nio-8080-exec-6]
   m.r.controller.RocketController      : 开始发送消息
12 2022-12-11 19:11:59.215 INFO 7936 --- [nio-8080-exec-6]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E44E5000F,
   offsetMsgId=AC17A00100002A9F0000000009A7CCC0, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61964]
13 2022-12-11 19:11:59.493 INFO 7936 --- [nio-8080-exec-7]
   m.r.controller.RocketController      : 开始发送消息
14 2022-12-11 19:11:59.511 INFO 7936 --- [nio-8080-exec-7]
   m.r.controller.RocketController      : 发送结果: SendResult
   [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E46060012,
   offsetMsgId=AC17A00100002A9F0000000009A7CFCA, messageQueue=MessageQueue
   [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61965]
15 2022-12-11 19:11:59.765 INFO 7936 --- [nio-8080-exec-8]
   m.r.controller.RocketController      : 开始发送消息

```

```

16 2022-12-11 19:11:59.783 INFO 7936 --- [nio-8080-exec-8]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E47150015,
    offsetMsgId=AC17A00100002A9F0000000009A7D2D4, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61966]
17 2022-12-11 19:11:59.983 INFO 7936 --- [nio-8080-exec-9]
    m.r.controller.RocketController      : 开始发送消息
18 2022-12-11 19:11:59.991 INFO 7936 --- [nio-8080-exec-9]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E47F00018,
    offsetMsgId=AC17A00100002A9F0000000009A7D5DE, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61967]
19 2022-12-11 19:12:04.639 INFO 7936 --- [io-8080-exec-10]
    m.r.controller.RocketController      : 开始发送消息
20 2022-12-11 19:12:04.669 INFO 7936 --- [io-8080-exec-10]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5A20001B,
    offsetMsgId=AC17A00100002A9F0000000009A7D8E7, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61968]
21 2022-12-11 19:12:04.965 INFO 7936 --- [nio-8080-exec-1]
    m.r.controller.RocketController      : 开始发送消息
22 2022-12-11 19:12:04.974 INFO 7936 --- [nio-8080-exec-1]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5B65001E,
    offsetMsgId=AC17A00100002A9F0000000009A7DBF1, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61969]
23 2022-12-11 19:12:05.222 INFO 7936 --- [nio-8080-exec-2]
    m.r.controller.RocketController      : 开始发送消息
24 2022-12-11 19:12:05.245 INFO 7936 --- [nio-8080-exec-2]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5C670021,
    offsetMsgId=AC17A00100002A9F0000000009A7DEFA, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61970]
25 2022-12-11 19:12:05.454 INFO 7936 --- [nio-8080-exec-3]
    m.r.controller.RocketController      : 开始发送消息
26 2022-12-11 19:12:05.466 INFO 7936 --- [nio-8080-exec-3]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5D4F0024,
    offsetMsgId=AC17A00100002A9F0000000009A7E204, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61971]
27 2022-12-11 19:12:05.699 INFO 7936 --- [nio-8080-exec-4]
    m.r.controller.RocketController      : 开始发送消息
28 2022-12-11 19:12:05.707 INFO 7936 --- [nio-8080-exec-4]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5E440027,
    offsetMsgId=AC17A00100002A9F0000000009A7E50E, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61972]
29 2022-12-11 19:12:05.875 INFO 7936 --- [nio-8080-exec-5]
    m.r.controller.RocketController      : 开始发送消息
30 2022-12-11 19:12:05.882 INFO 7936 --- [nio-8080-exec-5]
    m.r.controller.RocketController      : 发送结果: SendResult
    [sendStatus=SEND_OK, msgId=7F0000011F0066D3C617379E5EF3002A,
    offsetMsgId=AC17A00100002A9F0000000009A7E817, messageQueue=MessageQueue
    [topic=test_topic, brokerName=broker-a, queueId=1], queueOffset=61973]

```

将hashkey更改成2

```
1 | SendResult sendResult = rocketMQTemplate.syncSendOrderly("test_topic",  
    message, "2");
```

```
1 | 2022-12-11 19:13:39.399 INFO 23852 --- [nio-8080-exec-1]  
    m.r.controller.RocketController      : 开始发送消息  
2 | 2022-12-11 19:13:39.482 INFO 23852 --- [nio-8080-exec-1]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C617379FCC770000,  
    offsetMsgId=AC17A00100002A9F0000000009A7EB20, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61872]  
3 | 2022-12-11 19:13:39.888 INFO 23852 --- [nio-8080-exec-2]  
    m.r.controller.RocketController      : 开始发送消息  
4 | 2022-12-11 19:13:39.914 INFO 23852 --- [nio-8080-exec-2]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C617379FCE300003,  
    offsetMsgId=AC17A00100002A9F0000000009A7EE2A, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61873]  
5 | 2022-12-11 19:13:40.186 INFO 23852 --- [nio-8080-exec-3]  
    m.r.controller.RocketController      : 开始发送消息  
6 | 2022-12-11 19:13:40.200 INFO 23852 --- [nio-8080-exec-3]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C617379FCF5B0006,  
    offsetMsgId=AC17A00100002A9F0000000009A7F134, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61874]  
7 | 2022-12-11 19:14:18.792 INFO 23852 --- [nio-8080-exec-4]  
    m.r.controller.RocketController      : 开始发送消息  
8 | 2022-12-11 19:14:18.819 INFO 23852 --- [nio-8080-exec-4]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C61737A066290009,  
    offsetMsgId=AC17A00100002A9F0000000009A7F43E, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61875]  
9 | 2022-12-11 19:14:18.979 INFO 23852 --- [nio-8080-exec-5]  
    m.r.controller.RocketController      : 开始发送消息  
10 | 2022-12-11 19:14:18.989 INFO 23852 --- [nio-8080-exec-5]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C61737A066E3000C,  
    offsetMsgId=AC17A00100002A9F0000000009A7F748, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61876]  
11 | 2022-12-11 19:14:19.116 INFO 23852 --- [nio-8080-exec-6]  
    m.r.controller.RocketController      : 开始发送消息  
12 | 2022-12-11 19:14:19.123 INFO 23852 --- [nio-8080-exec-6]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015D2C66D3C61737A0676C000F,  
    offsetMsgId=AC17A00100002A9F0000000009A7FA52, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=2], queueOffset=61877]
```

将hashkey更改成7

```
1 | SendResult sendResult = rocketMQTemplate.syncSendOrderly("test_topic",  
    message, "7");
```

```
1 | 2022-12-11 19:15:05.288 INFO 24388 --- [nio-8080-exec-1]  
    m.r.controller.RocketController      : 开始发送消息  
2 | 2022-12-11 19:15:05.416 INFO 24388 --- [nio-8080-exec-1]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015F4466D3C61737A11BFA0000,  
    offsetMsgId=AC17A00100002A9F0000000009A7FD5B, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=3], queueOffset=62182]  
3 | 2022-12-11 19:15:06.419 INFO 24388 --- [nio-8080-exec-2]  
    m.r.controller.RocketController      : 开始发送消息  
4 | 2022-12-11 19:15:06.495 INFO 24388 --- [nio-8080-exec-2]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015F4466D3C61737A120330003,  
    offsetMsgId=AC17A00100002A9F0000000009A80065, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=3], queueOffset=62183]  
5 | 2022-12-11 19:15:06.580 INFO 24388 --- [nio-8080-exec-3]  
    m.r.controller.RocketController      : 开始发送消息  
6 | 2022-12-11 19:15:06.587 INFO 24388 --- [nio-8080-exec-3]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015F4466D3C61737A120D40006,  
    offsetMsgId=AC17A00100002A9F0000000009A8036F, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=3], queueOffset=62184]  
7 | 2022-12-11 19:15:06.718 INFO 24388 --- [nio-8080-exec-4]  
    m.r.controller.RocketController      : 开始发送消息  
8 | 2022-12-11 19:15:06.729 INFO 24388 --- [nio-8080-exec-4]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015F4466D3C61737A1215F0009,  
    offsetMsgId=AC17A00100002A9F0000000009A80678, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=3], queueOffset=62185]  
9 | 2022-12-11 19:15:06.866 INFO 24388 --- [nio-8080-exec-5]  
    m.r.controller.RocketController      : 开始发送消息  
10 | 2022-12-11 19:15:06.877 INFO 24388 --- [nio-8080-exec-5]  
    m.r.controller.RocketController      : 发送结果: SendResult  
    [sendStatus=SEND_OK, msgId=7F0000015F4466D3C61737A121F2000C,  
    offsetMsgId=AC17A00100002A9F0000000009A80982, messageQueue=MessageQueue  
    [topic=test_topic, brokerName=broker-a, queueId=3], queueOffset=62186]
```

发送异步消息

```

1 //发送普通异步消息-Object
2 asyncSend(String destination, Object payload, SendCallback sendCallback);
3
4 //发送普通异步消息-Message
5 asyncSend(String destination, Message<?> message, SendCallback
  sendCallback);
6
7 //发送普通异步消息-Object，并设置发送超时时间
8 asyncSend(String destination, Object payload, SendCallback sendCallback,
  long timeout);
9
10 //发送普通异步消息-Message，并设置发送超时时间
11 asyncSend(String destination, Message<?> message, SendCallback sendCallback,
  long timeout);
12
13 //发送普通异步延迟消息，并设置超时
14 asyncSend(String destination, Message<?> message, SendCallback sendCallback,
  long timeout,int delayLevel);

```

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.client.producer.SendCallback;
4 import org.apache.rocketmq.client.producer.SendResult;
5 import org.apache.rocketmq.common.message.Message;
6 import org.apache.rocketmq.spring.core.RocketMQTemplate;
7 import org.slf4j.Logger;
8 import org.slf4j.LoggerFactory;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.RestController;
12
13 import javax.annotation.Resource;
14 import java.nio.charset.StandardCharsets;
15 import java.util.ArrayList;
16 import java.util.List;
17
18 /**
19  * Project name(项目名称): rocketmq_spring_boot
20  * Package(包名): mao.rocketmq_spring_boot.controller
21  * Class(类名): RocketController
22  * Author(作者): mao
23  * Author QQ: 1296193245
24  * GitHub: https://github.com/maomao124/
25  * Date(创建日期): 2022/12/11
26  * Time(创建时间): 18:29
27  * Version(版本): 1.0
28  * Description(描述): 无
29  */
30
31 @RestController
32 public class RocketController

```

```

33 {
34     /**
35      * 日志
36      */
37     private static final Logger log =
38         LoggerFactory.getLogger(RocketController.class);
39
40     @Resource
41     private RocketMQTemplate rocketMQTemplate;
42
43     /**
44      * 发送消息
45      *
46      * @param data 数据
47      * @return {@link String}
48      */
49     @GetMapping("/save/{data}")
50     public String save(@PathVariable String data)
51     {
52         log.info("开始发送消息");
53         rocketMQTemplate.convertAndSend("test_topic", data);
54         return data;
55     }
56
57     /**
58      * 发送同步消息
59      *
60      * @param data 数据
61      * @return {@link SendResult}
62      */
63     @GetMapping("/save2/{data}")
64     public SendResult save2(@PathVariable String data)
65     {
66         log.info("开始发送同步消息");
67         Message message = new Message();
68         message.setBody(("同步消息:" +
69             data).getBytes(StandardCharsets.UTF_8));
70         return rocketMQTemplate.syncSend("test_topic", message);
71     }
72
73     /**
74      * 发送批量消息
75      *
76      * @param data 数据
77      * @return {@link SendResult}
78      */
79     @GetMapping("/save3/{data}")
80     public SendResult save3(@PathVariable String data)
81     {
82         log.info("开始发送批量消息");
83         List<Message> messageList = new ArrayList<>(10);
84         for (int i = 0; i < 10; i++)
85         {
86             Message message = new Message();
87             message.setBody(("批量消息:" + data + " -" +
88                 i).getBytes(StandardCharsets.UTF_8));
89             messageList.add(message);
90         }
91     }

```



```

88         return rocketMQTemplate.syncSend("test_topic", messageList);
89     }
90
91
92     /**
93      * 发送带tag的消息
94      *
95      * @param data 数据
96      * @return {@link String}
97      */
98     @GetMapping("/save4/{data}")
99     public String save4(@PathVariable String data)
100     {
101         log.info("开始发送带tag的消息");
102         //发送带tag的消息，直接在topic后面加上":tag"
103         rocketMQTemplate.convertAndSend("test_topic:tag1", data);
104         return data;
105     }
106
107     /**
108      * 发送消息
109      *
110      * @param data 数据
111      * @return {@link String}
112      */
113     @GetMapping("/save5/{data}")
114     public String save5(@PathVariable String data)
115     {
116         log.info("开始发送消息");
117         Message message = new Message();
118         message.setBody(("同步消息:" +
119 data).getBytes(StandardCharsets.UTF_8));
120         SendResult sendResult =
121 rocketMQTemplate.syncSendOrderly("test_topic", message, "1");
122         log.info("发送结果: " + sendResult);
123         return data;
124     }
125
126     /**
127      * 发送异步消息
128      *
129      * @param data 数据
130      * @return {@link String}
131      */
132     @GetMapping("/save6/{data}")
133     public String save6(@PathVariable String data)
134     {
135         log.info("开始发送异步消息");
136         rocketMQTemplate.asyncSend("test_group", data, new SendCallback()
137         {
138             @Override
139             public void onSuccess(SendResult sendResult)
140             {
141                 log.info("异步消息发送成功: " + sendResult);
142             }
143
144             @Override
145             public void onException(Throwable throwable)

```



```

144         {
145             log.error("异步消息发送失败: " + throwable);
146         }
147     });
148     return data;
149 }
150 }

```

```

1 2022-12-11 19:09:27.245 INFO 24200 --- [nio-8080-exec-1]
   m.r.controller.RocketController      : 开始发送异步消息
2 2022-12-11 19:09:27.296 INFO 24200 --- [ublicExecutor_1]
   m.r.controller.RocketController      : 异步消息发送成功: SendResult
   [sendStatus=SEND_OK, msgId=7F0000015E8866D3C617379BF35E0000,
   offsetMsgId=AC17A00100002A9F0000000009A7B3FB, messageQueue=MessageQueue
   [topic=test_group, brokerName=broker-a, queueId=2], queueOffset=25]
3 2022-12-11 19:09:53.704 INFO 24200 --- [nio-8080-exec-2]
   m.r.controller.RocketController      : 开始发送异步消息
4 2022-12-11 19:09:53.752 INFO 24200 --- [ublicExecutor_2]
   m.r.controller.RocketController      : 异步消息发送成功: SendResult
   [sendStatus=SEND_OK, msgId=7F0000015E8866D3C617379C5AA90003,
   offsetMsgId=AC17A00100002A9F0000000009A7B660, messageQueue=MessageQueue
   [topic=test_group, brokerName=broker-a, queueId=1], queueOffset=25]
5 2022-12-11 19:09:54.330 INFO 24200 --- [nio-8080-exec-3]
   m.r.controller.RocketController      : 开始发送异步消息
6 2022-12-11 19:09:54.420 INFO 24200 --- [ublicExecutor_3]
   m.r.controller.RocketController      : 异步消息发送成功: SendResult
   [sendStatus=SEND_OK, msgId=7F0000015E8866D3C617379C5D1A0006,
   offsetMsgId=AC17A00100002A9F0000000009A7B8C5, messageQueue=MessageQueue
   [topic=test_group, brokerName=broker-a, queueId=2], queueOffset=26]
7 2022-12-11 19:09:54.834 INFO 24200 --- [nio-8080-exec-5]
   m.r.controller.RocketController      : 开始发送异步消息
8 2022-12-11 19:09:54.862 INFO 24200 --- [ublicExecutor_4]
   m.r.controller.RocketController      : 异步消息发送成功: SendResult
   [sendStatus=SEND_OK, msgId=7F0000015E8866D3C617379C5F130009,
   offsetMsgId=AC17A00100002A9F0000000009A7BB2A, messageQueue=MessageQueue
   [topic=test_group, brokerName=broker-a, queueId=3], queueOffset=25]

```

发送单向消息

```

1 package mao.rocketmq_spring_boot.controller;
2
3 import org.apache.rocketmq.client.producer.SendCallback;
4 import org.apache.rocketmq.client.producer.SendResult;
5 import org.apache.rocketmq.common.message.Message;
6 import org.apache.rocketmq.spring.core.RocketMQTemplate;
7 import org.slf4j.Logger;
8 import org.slf4j.LoggerFactory;
9 import org.springframework.web.bind.annotation.GetMapping;
10 import org.springframework.web.bind.annotation.PathVariable;
11 import org.springframework.web.bind.annotation.RestController;
12
13 import javax.annotation.Resource;
14 import java.nio.charset.StandardCharsets;
15 import java.util.ArrayList;
16 import java.util.List;
17
18 /**
19  * Project name(项目名称): rocketMQ_spring_boot
20  * Package(包名): mao.rocketmq_spring_boot.controller
21  * Class(类名): RocketController
22  * Author(作者): mao
23  * Author QQ: 1296193245
24  * Github: https://github.com/maomao124/
25  * Date(创建日期): 2022/12/11
26  * Time(创建时间): 18:29
27  * Version(版本): 1.0
28  * Description(描述): 无
29  */
30
31 @RestController
32 public class RocketController
33 {
34     /**
35      * 日志
36      */
37     private static final Logger log =
38         LoggerFactory.getLogger(RocketController.class);
39
40     @Resource
41     private RocketMQTemplate rocketMQTemplate;
42
43     /**
44      * 发送消息
45      *
46      * @param data 数据
47      * @return {@link String}
48      */
49     @GetMapping("/save/{data}")
50     public String save(@PathVariable String data)
51     {
52         log.info("开始发送消息");
53         rocketMQTemplate.convertAndSend("test_topic", data);
54         return data;
55     }
56
57     /**
58      * 发送同步消息

```

```

58     *
59     * @param data 数据
60     * @return {@link SendResult}
61     */
62     @GetMapping("/save2/{data}")
63     public SendResult save2(@PathVariable String data)
64     {
65         log.info("开始发送同步消息");
66         Message message = new Message();
67         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
68         return rocketMQTemplate.syncSend("test_topic", message);
69     }
70
71     /**
72     * 发送批量消息
73     *
74     * @param data 数据
75     * @return {@link SendResult}
76     */
77     @GetMapping("/save3/{data}")
78     public SendResult save3(@PathVariable String data)
79     {
80         log.info("开始发送批量消息");
81         List<Message> messageList = new ArrayList<>(10);
82         for (int i = 0; i < 10; i++)
83         {
84             Message message = new Message();
85             message.setBody(("批量消息:" + data + " -" +
i).getBytes(StandardCharsets.UTF_8));
86             messageList.add(message);
87         }
88         return rocketMQTemplate.syncSend("test_topic", messageList);
89     }
90
91
92     /**
93     * 发送带tag的消息
94     *
95     * @param data 数据
96     * @return {@link String}
97     */
98     @GetMapping("/save4/{data}")
99     public String save4(@PathVariable String data)
100     {
101         log.info("开始发送带tag的消息");
102         //发送带tag的消息，直接在topic后面加上":tag"
103         rocketMQTemplate.convertAndSend("test_topic:tag1", data);
104         return data;
105     }
106
107     /**
108     * 发送消息
109     *
110     * @param data 数据
111     * @return {@link String}
112     */
113     @GetMapping("/save5/{data}")

```

```

114     public String save5(@PathVariable String data)
115     {
116         log.info("开始发送消息");
117         Message message = new Message();
118         message.setBody(("同步消息:" +
data).getBytes(StandardCharsets.UTF_8));
119         SendResult sendResult =
rocketMQTemplate.syncSendOrderly("test_topic", message, "1");
120         log.info("发送结果: " + sendResult);
121         return data;
122     }
123
124     /**
125      * 发送异步消息
126      *
127      * @param data 数据
128      * @return {@link String}
129      */
130     @GetMapping("/save6/{data}")
131     public String save6(@PathVariable String data)
132     {
133         log.info("开始发送异步消息");
134         rocketMQTemplate.asyncSend("test_group", data, new SendCallback()
135         {
136             @Override
137             public void onSuccess(SendResult sendResult)
138             {
139                 log.info("异步消息发送成功: " + sendResult);
140             }
141
142             @Override
143             public void onException(Throwable throwable)
144             {
145                 log.error("异步消息发送失败: " + throwable);
146             }
147         });
148         return data;
149     }
150
151
152     /**
153      * 发送单向消息
154      *
155      * @param data 数据
156      * @return {@link String}
157      */
158     @GetMapping("/save7/{data}")
159     public String save7(@PathVariable String data)
160     {
161         log.info("开始发送单向消息");
162         rocketMQTemplate.sendOneway("test_topic", data);
163         return data;
164     }
165 }

```

消息消费

一个消费者只能绑定一个队列，一个队列可以绑定多个消费者

@RocketMQMessageListener注解参数说明：

- consumerGroup：消费者订阅组，它是必需的，并且必须是唯一的
- topic：主题名字，生产发送的主题名
- consumeMode：消费模式，可选择并发或有序接收消息；默认CONCURRENTLY同时接收异步传递的消息
- messageModel：消息模式，默认CLUSTERING集群消费；如果希望所有订阅者都接收消息，可以设置广播BROADCASTING
- consumeThreadMax：消费者最大线程数，默认64
- consumeTimeout：消息阻塞最长时间，默认15分钟
- nameServer：服务器地址，默认读取配置文件地址，可以单独为消费者设置指定位置
- selectorExpression：消费指定的Tag标签的业务消息

```
1 package mao.rocketmq_spring_boot.consumer;
2
3
4 import org.apache.rocketmq.spring.annotation.RocketMQMessageListener;
5 import org.apache.rocketmq.spring.core.RocketMQListener;
6 import org.slf4j.Logger;
7 import org.slf4j.LoggerFactory;
8 import org.springframework.stereotype.Component;
9
10 /**
11  * Project name(项目名称): rocketmq_spring_boot
12  * Package(包名): mao.rocketmq_spring_boot.consumer
13  * Class(类名): RocketCustomer
14  * Author(作者): mao
15  * Author QQ: 1296193245
16  * GitHub: https://github.com/maomao124/
17  * Date(创建日期): 2022/12/11
18  * Time(创建时间): 19:25
19  * Version(版本): 1.0
20  * Description(描述): 消费者
21  */
22
23 @Component
24 @RocketMQMessageListener(consumerGroup = "mao_group", topic = "test_topic")
25 public class RocketCustomer implements RocketMQListener<Object>
26 {
27
28     /**
```

```

29     * 日志
30     */
31     private static final Logger log =
LoggerFactory.getLogger(RocketCustomer.class);
32
33     @Override
34     public void onMessage(Object o)
35     {
36         //直接调用toString方法打印
37         log.info("消费者监听到一条消息: " + o.toString());
38     }
39 }

```

```

1  2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_7]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息75      属性
a: 7      属性b: 15
2  2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_17]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息83      属性
a: 2      属性b: 5
3  2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_15]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=", "tran
sactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":
true,"buyerId":null}
4  2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_6]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息95      属性
a: 6      属性b: 8
5  2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_18]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息0      属性
a: 0      属性b: 17
6  2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_8]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息12      属性
a: 1      属性b: 2
7  2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_16]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息8      属性
a: 10     属性b: 9
8  2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_2]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息: 消息16      属性
a: 6      属性b: 9
9  2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_18]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=", "tran
sactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":
true,"buyerId":null}
10 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_17]
m.r.consumer.RocketCustomer      : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=", "tran
sactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":
true,"buyerId":null}

```

```

11 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_15]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息:
    {"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

12 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_4]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息20      属性
    a: 4      属性b: 3

13 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_5]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息24      属性
    a: 8      属性b: 12

14 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_16]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息:
    {"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

15 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_1]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息32      属性
    a: 1      属性b: 5

16 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_3]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息44      属性
    a: 9      属性b: 17

17 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_10]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息64      属性
    a: 3      属性b: 4

18 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_16]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息:
    {"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

19 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_9]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息76      属性
    a: 1      属性b: 1

20 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_11]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息80      属性
    a: 10     属性b: 3

21 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_12]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息84      属性
    a: 5      属性b: 15

22 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_20]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息97      属性
    a: 9      属性b: 6

23 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_19]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息: 消息88      属性
    a: 9      属性b: 0

24 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_14]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息:
    {"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

25 2022-12-11 19:29:57.189 INFO 4456 --- [ad_mao_group_13]
    m.r.consumer.RocketCustomer          : 消费者监听到一条消息:
    {"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

```

26 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_16]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

27 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_10]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

28 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_11]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

29 2022-12-11 19:29:57.189 INFO 4456 --- [ead_mao_group_7]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

30 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_6]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

31 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_8]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

32 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_2]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

33 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_15]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

34 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_17]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

35 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_6]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

36 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_18]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}


```

37 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_4]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

38 2022-12-11 19:29:57.191 INFO 4456 --- [ead_mao_group_6]
m.r.consumer.RocketCustomer : 消费者监听到一条消息: 消息96 属性
a: 6 属性b: 1

39 2022-12-11 19:29:57.191 INFO 4456 --- [ad_mao_group_15]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

40 2022-12-11 19:29:57.191 INFO 4456 --- [ad_mao_group_18]
m.r.consumer.RocketCustomer : 消费者监听到一条消息: 2

41 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_5]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

42 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_1]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

43 2022-12-11 19:29:57.191 INFO 4456 --- [ead_mao_group_4]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

44 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_20]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

45 2022-12-11 19:29:57.190 INFO 4456 --- [ead_mao_group_3]
m.r.consumer.RocketCustomer : 消费者监听到一条消息: 2

46 2022-12-11 19:29:57.190 INFO 4456 --- [ad_mao_group_12]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjI=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

47 2022-12-11 19:29:57.191 INFO 4456 --- [ead_mao_group_4]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

48 2022-12-11 19:29:57.191 INFO 4456 --- [ad_mao_group_20]
m.r.consumer.RocketCustomer : 消费者监听到一条消息:
{"topic":null,"flag":0,"properties":null,"body":"5ZCM5q2l5raI5oGvOjk=","transactionId":null,"keys":null,"tags":null,"delayTimeLevel":0,"waitStoreMsgOK":true,"buyerId":null}

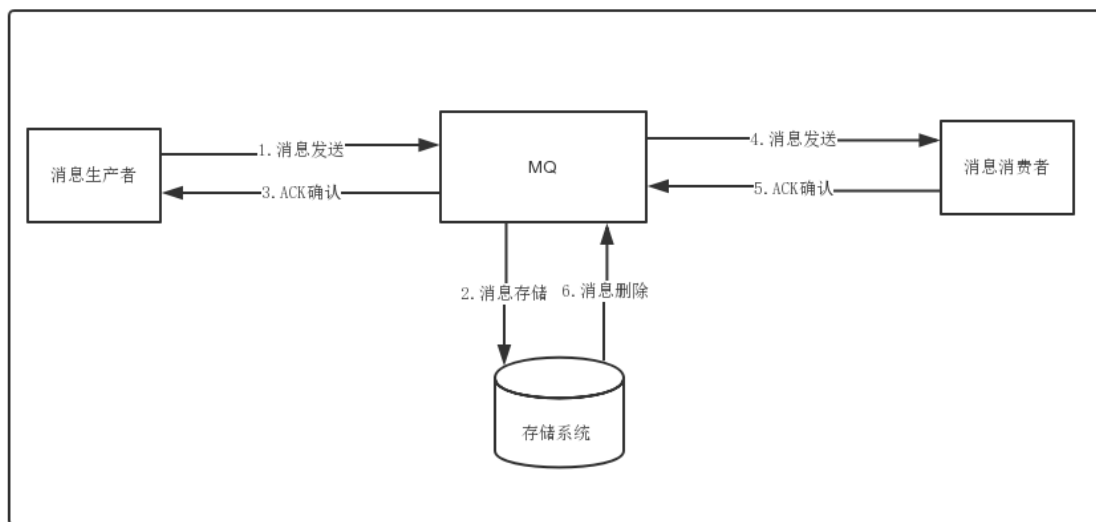
49 2022-12-11 19:29:57.191 INFO 4456 --- [ead_mao_group_3]
m.r.consumer.RocketCustomer : 消费者监听到一条消息: 9

```

高级功能

消息存储

分布式队列因为有高可靠性的要求，所以数据要进行持久化存储。



步骤：

1. 消息生成者发送消息
2. MQ收到消息，将消息进行持久化，在存储中新增一条记录
3. 返回ACK给生产者
4. MQ push 消息给对应的消费者，然后等待消费者返回ACK
5. 如果消息消费者在指定时间内成功返回ack，那么MQ认为消息消费成功，在存储中删除消息，即执行第6步；如果MQ在指定时间内没有收到ACK，则认为消息消费失败，会尝试重新push消息,重复执行4、5、6步骤
6. MQ删除消息

存储介质

存储系统主要有两种，一种是关系型数据库，一种是文件系统

关系型数据库

Apache下开源的另外一款MQ—ActiveMQ（默认采用的KahaDB做消息存储）可选用JDBC的方式来做消息持久化，通过简单的xml配置信息即可实现JDBC消息存储。由于，普通关系型数据库（如Mysql）在单表数据量达到千万级别的情况下，其IO读写性能往往会出现瓶颈。在可靠性方面，该种方案非常依赖DB，如果一旦DB出现故障，则MQ的消息就无法落盘存储会导致线上故障

文件系统

目前业界较为常用的几款产品（RocketMQ/Kafka/RabbitMQ）均采用的是消息刷盘至所部署虚拟机/物理机的文件系统来做持久化（刷盘一般可以分为异步刷盘和同步刷盘两种模式）。消息刷盘为消息存储提供了一种高效率、高可靠性和高性能的数据持久化方式。除非部署MQ机器本身或是本地磁盘挂了，否则一般是不会出现无法持久化的故障问题。

性能对比

文件系统的性能要大于关系型数据库

消息的存储和发送

消息存储

磁盘如果使用得当，磁盘的速度完全可以匹配上网络的数据传输速度。目前的高性能磁盘，顺序写速度可以达到600MB/s，超过了一般网卡的传输速度。但是磁盘随机写的速度只有大概100KB/s，和顺序写的性能相差6000倍！因为有如此巨大的速度差别，好的消息队列系统会比普通的消息队列系统速度快多个数量级。RocketMQ的消息用顺序写,保证了消息存储的速度。

消息发送

Linux操作系统分为【用户态】和【内核态】，文件操作、网络操作需要涉及这两种形态的切换，免不了进行数据复制。

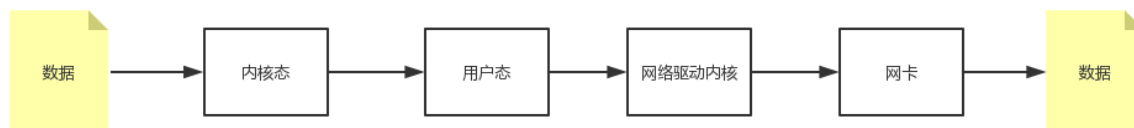
一台服务器 把本机磁盘文件的内容发送到客户端，一般分为两个步骤：

1. read；读取本地文件内容
2. write；将读取的内容通过网络发送出去

这两个看似简单的操作，实际进行了4次数据复制，分别是：

1. 从磁盘复制数据到内核态内存；
2. 从内核态内存复制到用户态内存；

3. 然后从用户态内存复制到网络驱动的内核态内存；
4. 最后是从网络驱动的内核态内存复制到网卡中进行传输。



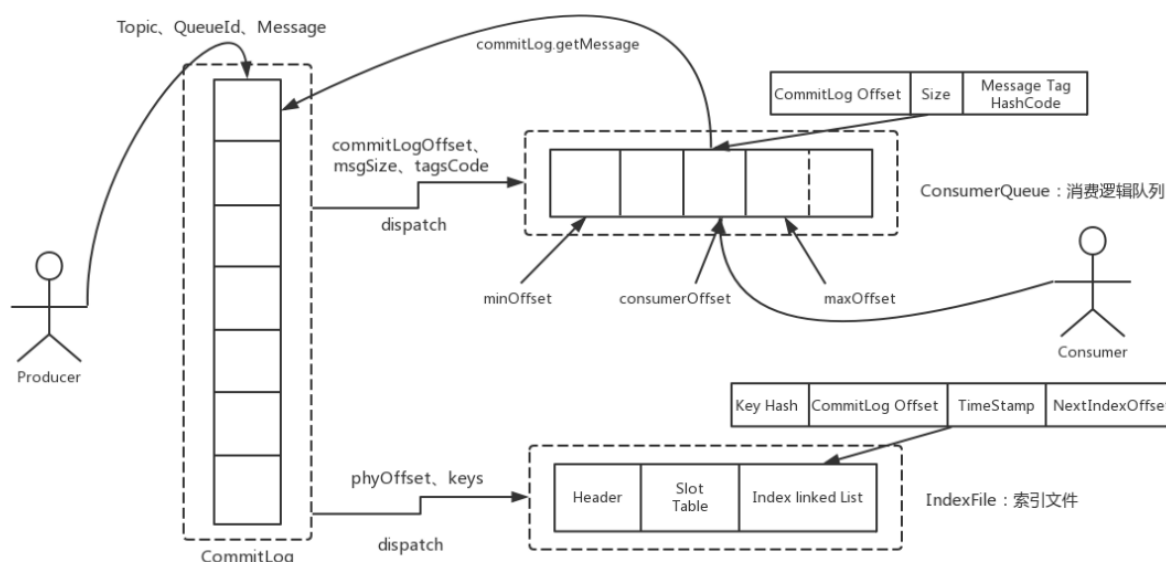
rocketMQ是通过使用mmap的方式，可以省去向用户态的内存复制，提高速度。这种机制在Java中是通过MappedByteBuffer实现的

RocketMQ充分利用了上述特性，也就是所谓的“零拷贝”技术，提高消息存盘和网络发送的速度。

这里需要注意的是，采用MappedByteBuffer这种内存映射的方式有几个限制，其中之一是一次只能映射1.5~2G 的文件至用户态的虚拟内存，这也是为何RocketMQ默认设置单个CommitLog日志数据文件为1G的原因了

消息存储结构

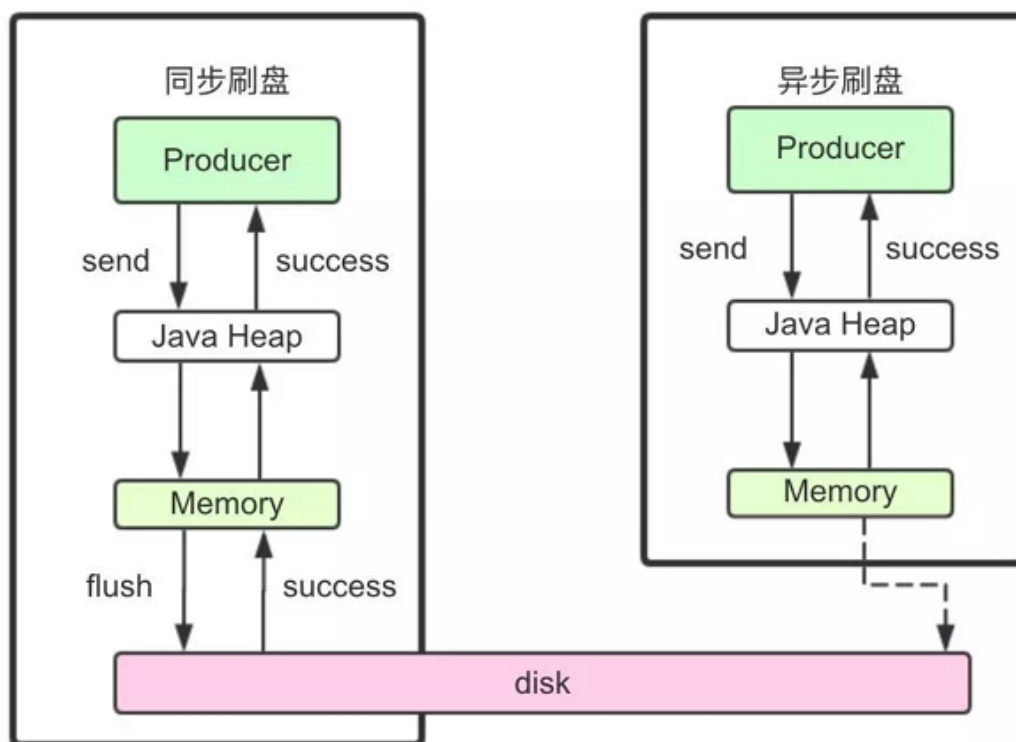
RocketMQ消息的存储是由ConsumeQueue和CommitLog配合完成的，消息真正的物理存储文件是CommitLog，ConsumeQueue是消息的逻辑队列，类似数据库的索引文件，存储的是指向物理存储的地址。每个Topic下的每个Message Queue都有一个对应的ConsumeQueue文件。



- CommitLog: 存储消息的元数据
- ConsumerQueue: 存储消息在CommitLog的索引
- IndexFile: 为了消息查询提供了一种通过key或时间区间来查询消息的方法，这种通过IndexFile来查找消息的方法不影响发送与消费消息的主流程

刷盘机制

RocketMQ的消息是存储到磁盘上的，这样既能保证断电后恢复，又可以让存储的消息量超出内存的限制。RocketMQ为了提高性能，会尽可能地保证磁盘的顺序写。消息在通过Producer写入RocketMQ的时候，有两种写磁盘方式，分布式同步刷盘和异步刷盘。



同步刷盘还是异步刷盘，都是通过Broker配置文件里的flushDiskType 参数设置的，这个参数被配置成 SYNC_FLUSH、ASYNC_FLUSH中的一个

同步刷盘

在返回写成功状态时，消息已经被写入磁盘。具体流程是，消息写入内存的PAGECACHE后，立刻通知刷盘线程刷盘，然后等待刷盘完成，刷盘线程执行完成后唤醒等待的线程，返回消息写成功的状态。

异步刷盘

在返回写成功状态时，消息可能只是被写入了内存的PAGECACHE，写操作的返回快，吞吐量大；当内存里的消息量积累到一定程度时，统一触发写磁盘动作，快速写入。

高可用性机制

RocketMQ分布式集群是通过Master和Slave的配合达到高可用性的。

Master和Slave的区别：在Broker的配置文件中，参数 brokerId的值为0表明这个Broker是Master，大于0表明这个Broker是 Slave，同时brokerRole参数也会说明这个Broker是Master还是Slave。

Master角色的Broker支持读和写，Slave角色的Broker仅支持读，也就是 Producer只能和Master角色的Broker连接写入消息；Consumer可以连接 Master角色的Broker，也可以连接Slave角色的Broker来读取消息。

消息消费高可用

在Consumer的配置文件中，并不需要设置是从Master读还是从Slave 读，当Master不可用或者繁忙的时候，Consumer会被自动切换到从Slave 读。有了自动切换Consumer这种机制，当一个Master角色的机器出现故障后，Consumer仍然可以从Slave读取消息，不影响Consumer程序。这就达到了消费端的高可用性。

消息发送高可用

在创建Topic的时候，把Topic的多个Message Queue创建在多个Broker组上（相同Broker名称，不同brokerId的机器组成一个Broker组），这样当一个Broker组的Master不可用后，其他组的Master仍然可用，Producer仍然可以发送消息。RocketMQ目前还不支持把Slave自动转成Master，如果机器资源不足，需要把Slave转成Master，则要手动停止Slave角色的Broker，更改配置文件，用新的配置文件启动Broker。

消息主从复制

如果一个Broker组有Master和Slave，消息需要从Master复制到Slave 上，有同步和异步两种复制方式

同步复制和异步复制是通过Broker配置文件里的brokerRole参数进行设置的，这个参数可以被设置成ASYNC_MASTER、SYNC_MASTER、SLAVE三个值中的一个

同步复制

同步复制方式是等Master和Slave均写成功后才反馈给客户端写成功状态

在同步复制方式下，如果Master出故障，Slave上有全部的备份数据，容易恢复，但是同步复制会增大数据写入 延迟，降低系统吞吐量

异步复制

异步复制方式是只要Master写成功即可反馈给客户端写成功状态

在异步复制方式下，系统拥有较低的延迟和较高的吞吐量，但是如果Master出了故障，有些数据因为没有被写入Slave，有可能会丢失

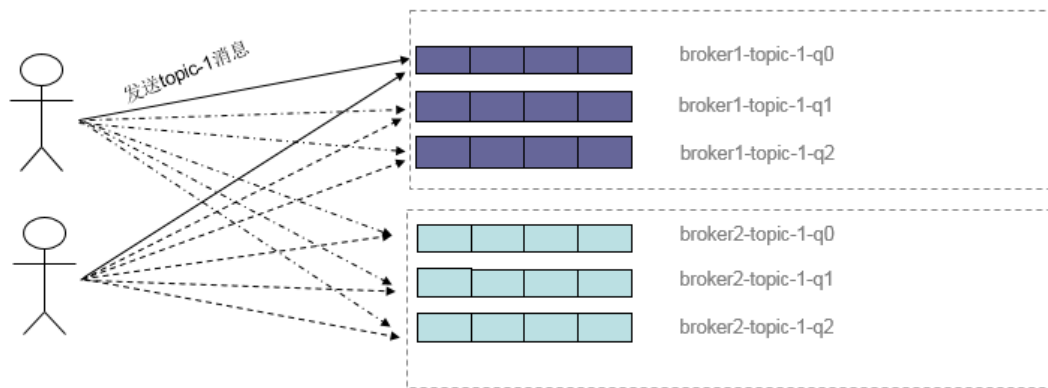
总结

实际应用中要结合业务场景，合理设置刷盘方式和主从复制方式，尤其是SYNC_FLUSH方式，由于频繁地触发磁盘写动作，会明显降低 性能。通常情况下，应该把Master和Slave配置成ASYNC_FLUSH的刷盘方式，主从之间配置成SYNC_MASTER的复制方式，这样即使有一台机器出故障，仍然能保证数据不丢，是个不错的选择。

负载均衡

Producer负载均衡

Producer端，每个实例在发消息的时候，默认会轮询所有的message queue发送，以达到让消息平均落在不同的queue上。而由于queue可以散落在不同的broker，所以消息就发送到不同的broker下



发布方会把第一条消息发送至 Queue 0，然后第二条消息发送至 Queue 1，以此类推

Consumer负载均衡

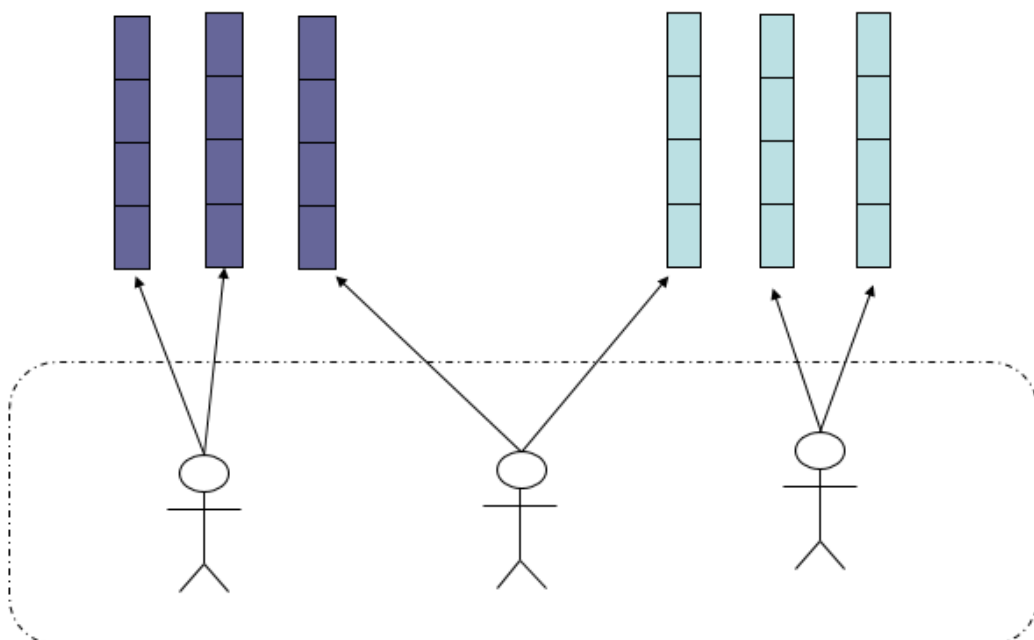
集群模式

在集群消费模式下，每条消息只需要投递到订阅这个topic的Consumer Group下的一个实例即可。RocketMQ采用主动拉取的方式拉取并消费消息，在拉取的时候需要明确指定拉取哪一条message queue。

而每当实例的数量有变更，都会触发一次所有实例的负载均衡，这时候会按照queue的数量和实例的数量平均分配queue给每个实例

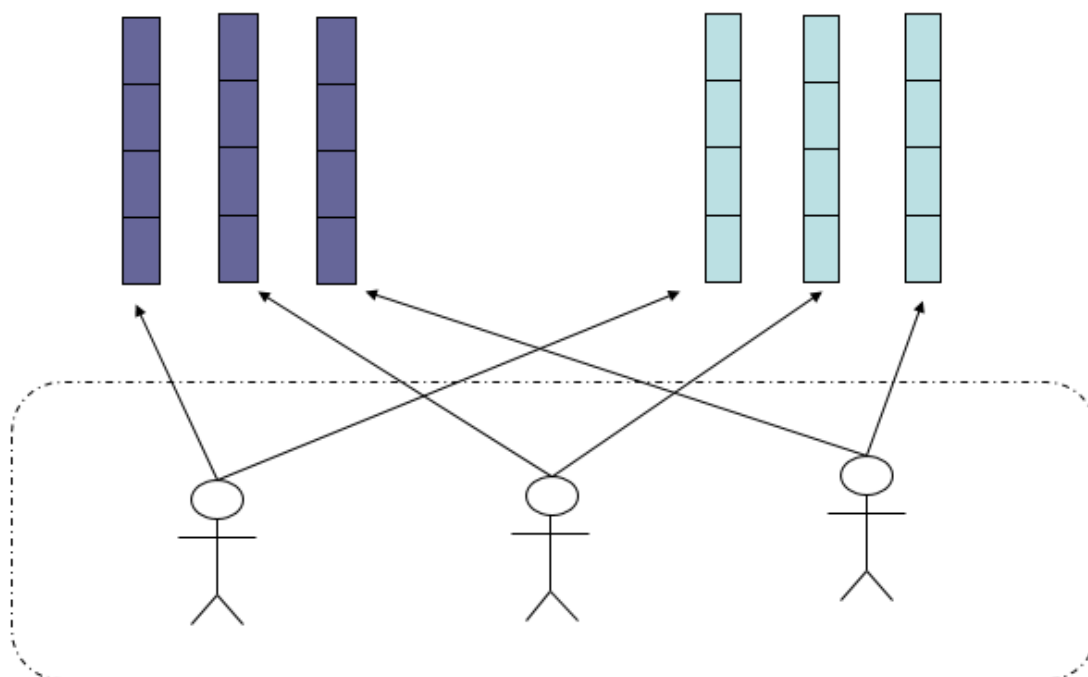
默认的分配算法是AllocateMessageQueueAveragely

每个consumer实例平均分配每个consume queue



还有另外一种平均的算法是AllocateMessageQueueAveragelyByCircle，也是平均分摊每一条queue，只是以环状轮流分queue的形式

每个consumer实例平均分配每个consume queue



需要注意的是，集群模式下，queue都是只允许分配只一个实例，这是由于如果多个实例同时消费一个queue的消息，由于拉取哪些消息是consumer主动控制的，那样会导致同一个消息在不同的实例下被消费多次，所以算法上都是一个queue只分给一个consumer实例，一个consumer实例可以同时分到不同的queue。

通过增加consumer实例去分摊queue的消费，可以起到水平扩展的消费能力的作用。而有实例下线的时候，会重新触发负载均衡，这时候原来分配到的queue将分配到其他实例上继续消费。

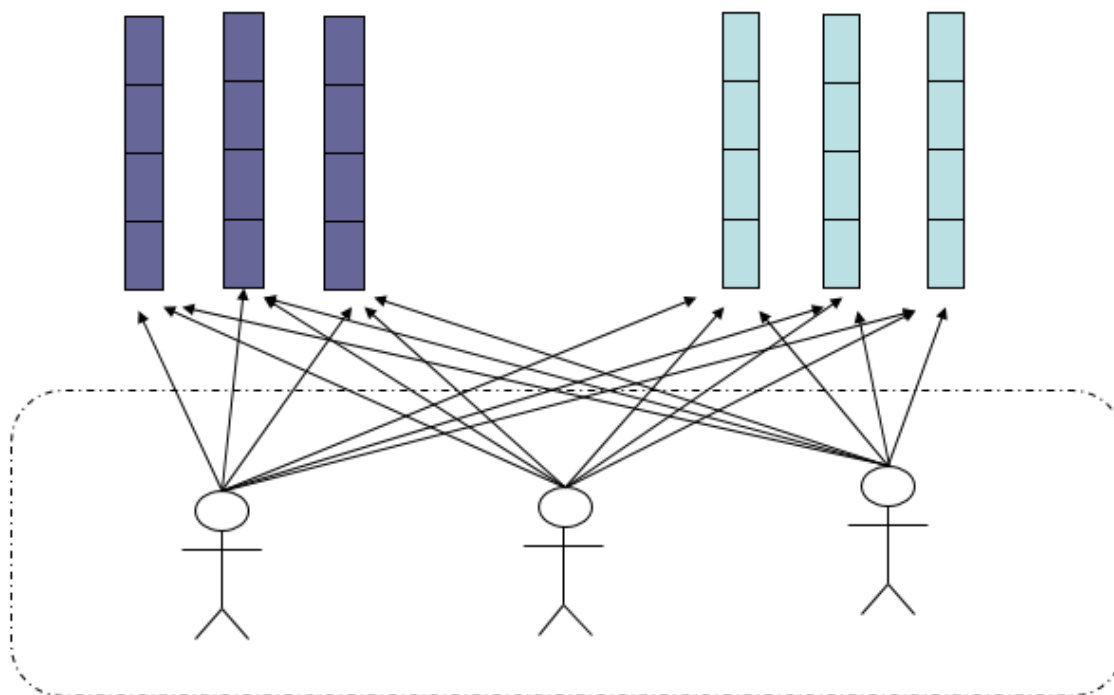
但是如果consumer实例的数量比message queue的总数量还多的话，多出来的consumer实例将无法分到queue，也就无法消费到消息，也就无法起到分摊负载的作用了。所以需要控制让queue的总数量大于等于consumer的数量。

广播模式

由于广播模式下要求一条消息需要投递到一个消费组下面所有的消费者实例，所以也就没有消息被分摊消费的说法。

在实现上，其中一个不同就是在consumer分配queue的时候，所有consumer都分到所有的queue。

每个consumer实例分配每个consume queue



消息重试

顺序消息的重试

对于顺序消息，当消费者消费消息失败后，消息队列RocketMQ会自动不断进行消息重试（每次间隔时间为1秒），这时，应用会出现消息消费被阻塞的情况。因此，在使用顺序消息时，务必保证应用能够及时监控并处理消费失败的情况，避免阻塞现象的发生。

无序消息的重试

对于无序消息（普通、定时、延时、事务消息），当消费者消费消息失败时，您可以通过设置返回状态达到消息重试的结果。

无序消息的重试只针对集群消费方式生效；广播方式不提供失败重试特性，即消费失败后，失败消息不再重试，继续消费新的消息。

重试次数

消息队列 RocketMQ 默认允许每条消息最多重试 16 次，每次重试的间隔时间如下：

第几次重试	与上次重试的间隔时间	第几次重试	与上次重试的间隔时间
1	10 秒	9	7 分钟
2	30 秒	10	8 分钟
3	1 分钟	11	9 分钟
4	2 分钟	12	10 分钟
5	3 分钟	13	20 分钟
6	4 分钟	14	30 分钟
7	5 分钟	15	1 小时
8	6 分钟	16	2 小时

如果消息重试 16 次后仍然失败，消息将不再投递。如果严格按照上述重试时间间隔计算，某条消息在一直消费失败的前提下，将会在接下来的 4 小时 46 分钟之内进行 16 次重试，超过这个时间范围消息将不再重试投递。

一条消息无论重试多少次，这些重试消息的 Message ID 不会改变

配置方式

消费失败后，重试配置方式

集群消费方式下，消息消费失败后期望消息重试，需要在消息监听器接口的实现中明确进行配置（三种方式任选一种）：

- 返回 Action.ReconsumeLater（推荐）
- 返回 Null
- 抛出异常

消费失败后，不重试配置方式

集群消费方式下，消息失败后期望消息不重试，需要捕获消费逻辑中可能抛出的异常，最终返回 Action.CommitMessage，此后这条消息将不会再重试

自定义消息最大重试次数

消息队列 RocketMQ 允许 Consumer 启动的时候设置最大重试次数，重试时间间隔将按照如下策略：

- 最大重试次数小于等于 16 次，则重试时间间隔同上表描述。
- 最大重试次数大于 16 次，超过 16 次的重试时间间隔均为每次 2 小时。

```
1 Properties properties = new Properties();
2 //配置对应 Group ID 的最大消息重试次数为 20 次
3 properties.put(PropertyKeyConst.MaxReconsumeTimes,"20");
4 Consumer consumer =ONSFactory.createConsumer(properties);
```

- 消息最大重试次数的设置对相同 Group ID 下的所有 Consumer 实例有效。
- 如果只对相同 Group ID 下两个 Consumer 实例中的其中一个设置了 MaxReconsumeTimes，那么该配置对两个 Consumer 实例均生效。
- 配置采用覆盖的方式生效，即最后启动的 Consumer 实例会覆盖之前的启动实例的配置

获取消息重试次数

消费者收到消息后，可按照如下方式获取消息的重试次数：

```
1 public class MessageListenerImpl implements MessageListener {
2     @Override
3     public Action consume(Message message, ConsumeContext context) {
4         //获取消息的重试次数
5         System.out.println(message.getReconsumeTimes());
6         return Action.CommitMessage;
7     }
8 }
```

死信队列

当一条消息初次消费失败，消息队列 RocketMQ 会自动进行消息重试；达到最大重试次数后，若消费依然失败，则表明消费者在正常情况下无法正确地消费该消息，此时，消息队列 RocketMQ 不会立刻将消息丢弃，而是将其发送到该消费者对应的特殊队列中。

在消息队列 RocketMQ 中，这种正常情况下无法被消费的消息称为死信消息（Dead-Letter Message），存储死信消息的特殊队列称为死信队列（Dead-Letter Queue）。

死信特性

死信消息具有以下特性

- 不会再被消费者正常消费。
- 有效期与正常消息相同，均为 3 天，3 天后会被自动删除。因此，请在死信消息产生后的 3 天内及时处理。

死信队列具有以下特性：

- 一个死信队列对应一个 Group ID，而不是对应单个消费者实例。
- 如果一个 Group ID 未产生死信消息，消息队列 RocketMQ 不会为其创建相应的死信队列。
- 一个死信队列包含了对应 Group ID 产生的所有死信消息，不论该消息属于哪个 Topic。

一条消息进入死信队列，意味着某些因素导致消费者无法正常消费该消息，因此，通常需要您对其进行特殊处理。排查可疑因素并解决问题后，可以在消息队列 RocketMQ 控制台重新发送该消息，让消费者重新消费一次。

消费幂等

消息队列 RocketMQ 消费者在接收到消息以后，有必要根据业务上的唯一 Key 对消息做幂等处理的必要性

消费幂等的必要性

在互联网应用中，尤其在网络不稳定的情况下，消息队列 RocketMQ 的消息有可能会重复，这个重复简单可以概括为以下情况：

- 发送时消息重复

当一条消息已被成功发送到服务端并完成持久化，此时出现了网络闪断或者客户端宕机，导致服务端对客户端应答失败。如果此时生产者意识到消息发送失败并尝试再次发送消息，消费者后续会收到两条内容相同并且 Message ID 也相同的消息。

- 投递时消息重复

消息消费的场景下，消息已投递到消费者并完成业务处理，当客户端给服务端反馈应答的时候网络闪断。为了保证消息至少被消费一次，消息队列 RocketMQ 的服务端将在网络恢复后再次尝试投递之前已被处理过的消息，消费者后续会收到两条内容相同并且 Message ID 也相同的消息。

- 负载均衡时消息重复（包括但不限于网络抖动、Broker 重启以及订阅方应用重启）

当消息队列 RocketMQ 的 Broker 或客户端重启、扩容或缩容时，会触发 Rebalance，此时消费者可能会收到重复消息。

处理方式

因为 Message ID 有可能出现冲突（重复）的情况，所以真正安全的幂等处理，不建议以 Message ID 作为处理依据。最好的方式是以业务唯一标识作为幂等处理的关键依据，而业务的唯一标识可以通过消息 Key 进行设置：

```
1 Message message = new Message();
2 message.setKey("ORDERID_100");
3 SendResult sendResult = producer.send(message);
```

订阅方收到消息时可以根据消息的 Key 进行幂等处理：

```
1 consumer.subscribe("test", "*", new MessageListener() {
2     public Action consume(Message message, ConsumeContext context) {
3         String key = message.getKey()
4         // 根据业务唯一标识的 key 做幂等处理
5     }
6 });
```

小结

- RocketMQ消息消费方式分别为集群模式、广播模式。
- 消息队列负载由RebalanceService线程默认每隔20s进行一次消息队列负载，根据当前消费者组内消费者个数与主题队列数量按照某一种负载算法进行队列分配，分配原则为同一个消费者可以分配多个消息消费队列，同一个消息消费队列同一个时间只会分配给一个消费者。
- 消息拉取由PullMessageService线程根据RebalanceService线程创建的拉取任务进行拉取，默认每次拉取32条消息，提交给消费者消费线程后继续下一次消息拉取。如果消息消费过慢产生消息堆积会触发消息消费拉取流控。
- 并发消息消费指消费线程池中的线程可以并发对同一个消息队列的消息进行消费，消费成功后，取出消息队列中最小的消息偏移量作为消息消费进度偏移量存储在于消息消费进度存储文件中，集群模式消息消费进度存储在Broker（消息服务器），广播模式消息消费进度存储在消费者端。
- RocketMQ不支持任意精度的定时调度消息，只支持自定义的消息延迟级别，例如1s、2s、5s等，可通过在broker配置文件中设置messageDelayLevel。
- 顺序消息一般使用集群模式，是指对消息消费者内的线程池中的线程对消息消费队列只能串行消费。并发的消息消费最本质的区别是消息消费时必须成功锁定消息消费队列，在Broker端会存储消息消费队列的锁占用情况。

end

by mao
2022 12 12
