

Java应用接入RocketMQ生产消息

接入步骤

1. 新建Maven工程，添加依赖，注意Maven工程依赖与RocketMQ版本保持一致

```
<repositories>
  <repository>
    <id>aliyun</id>
    <name>aliyun</name>
    <url>https://maven.aliyun.com/repository/public</url>
  </repository>
</repositories>
<pluginRepositories>
  <pluginRepository>
    <id>aliyun</id>
    <name>aliyun</name>
    <url>https://maven.aliyun.com/repository/public</url>
  </pluginRepository>
</pluginRepositories>
...
<dependency>
  <groupId>org.apache.rocketmq</groupId>
  <artifactId>rocketmq-client</artifactId>
  <version>4.9.2</version>
</dependency>
```

2. 编写生产者代码

```
public static void main(String[] args) throws MQClientException,
RemotingException, InterruptedException, MQBrokerException {
    //创建DefaultMQProducer消息生产者对象
    DefaultMQProducer producer = new DefaultMQProducer("pg1");
    //设置NameServer
    producer.setNamesrvAddr("192.168.31.103:9876");

    //设置NameServer节点地址，多个节点间用分号分割
    //producer.setNamesrvAddr("192.168.31.103:9876");

    try{
        //与NameServer建立长连接
        producer.start();
        //发送一百条数据
        for(int i = 0 ; i < 100 ; i++) {
            //数据正文
            String data = "{\"title\":\"X市2021年度第四季度税务汇总数据\"}";
            /*创建消息
            Message消息三个参数
            topic 代表消息主题，自定义为tax-data-topic说明是税务数据
```

tags 代表标志，用于消费者接收数据时进行数据筛选。2021S1代表2021年第一季度数据

body 代表消息内容

```
*/  
Message message = new Message("TAX", "2021S4", data.getBytes());  
//发送消息，获取发送结果  
SendResult result = null;  
result = producer.send(message);  
//将发送结果对象打印在控制台  
System.out.println("消息已发送: MsgId:" + result.getMsgId() + ", 发送状态:"  
+ result.getSendStatus());  
}  
} finally {  
    try {  
        producer.shutdown();  
    } catch (Exception e){  
  
    }  
}  
}
```

3. 执行结果，同步复制下，Master被写入主队列，但从复制失败。

消息已发送: MsgId:C0A81F6D310418B4AAC26F9FD63F0061, 发送状态:SLAVE_NOT_AVAILABLE

消息已发送: MsgId:C0A81F6D310418B4AAC26F9FD6430062, 发送状态:SLAVE_NOT_AVAILABLE

消息已发送: MsgId:C0A81F6D310418B4AAC26F9FD6460063, 发送状态:SLAVE_NOT_AVAILABLE

同步复制与异步复制的区别

#关闭Broker

```
sh /usr/local/rocketmq-4.9.2/bin/mqshutdown broker
```

#重新生成配置文件，改为异步复制

```
cat > single-master.properties <<-'EOF'
```

#集群名称，同一个集群下的broker要求统一

```
brokerClusterName=DefaultCluster
```

#broker名称

```
brokerName=broker-a
```

#brokerId=0代表主节点，大于零代表从节点

```
brokerId=0
```

#删除日志文件时间点，默认凌晨 4 点

```
deleteWhen=04
```

#日志文件保留时间，默认 48 小时

fileReservedTime=48

#Broker 复制策略

#- ASYNC_MASTER 异步复制Master

#- SYNC_MASTER 同步双写Master

brokerRole=ASYNC_MASTER

#刷盘方式

#- ASYNC_FLUSH 异步刷盘，性能好宕机会丢数

#- SYNC_FLUSH 同步刷盘，性能较差不会丢数

flushDiskType=SYNC_FLUSH

#末尾追加，NameServer节点列表，使用分号分割

namesrvAddr=192.168.31.103:9876

autoCreateTopicEnable=true

EOF

#防火墙放行

firewall-cmd --zone=public --add-port=10911/tcp --permanent

firewall-cmd --reload

#启动broker服务

nohup sh ./bin/mqbroker -c ./single-master.properties > broker.log &

改为异步复制后，直接SEND_OK

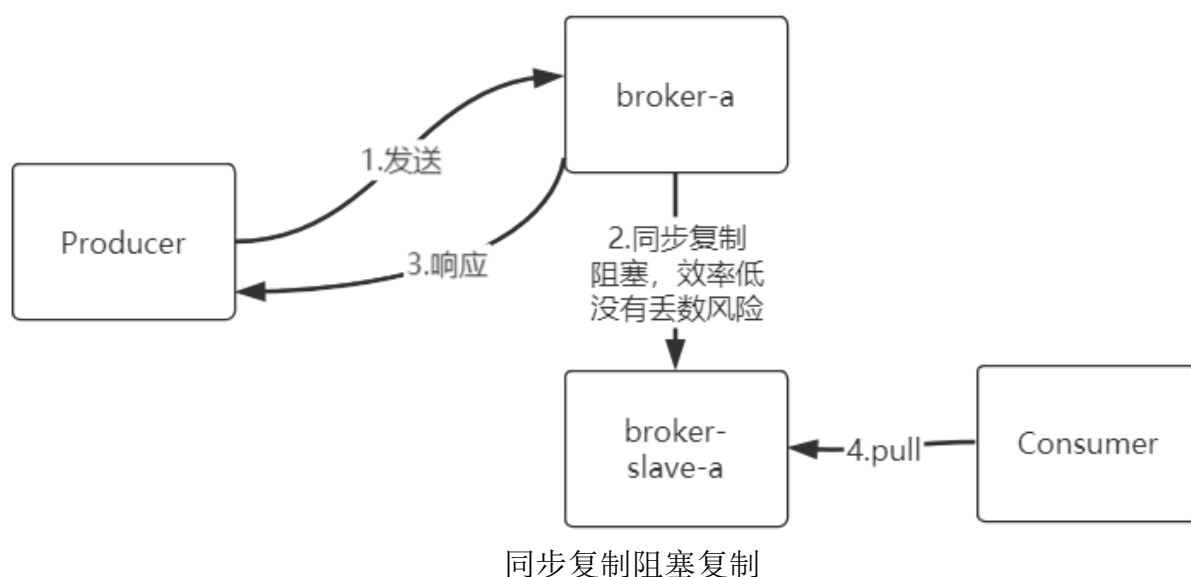
消息已发送: MsgId:C0A81F6D264418B4AAC26FA3EF500060, 发送状态:SEND_OK

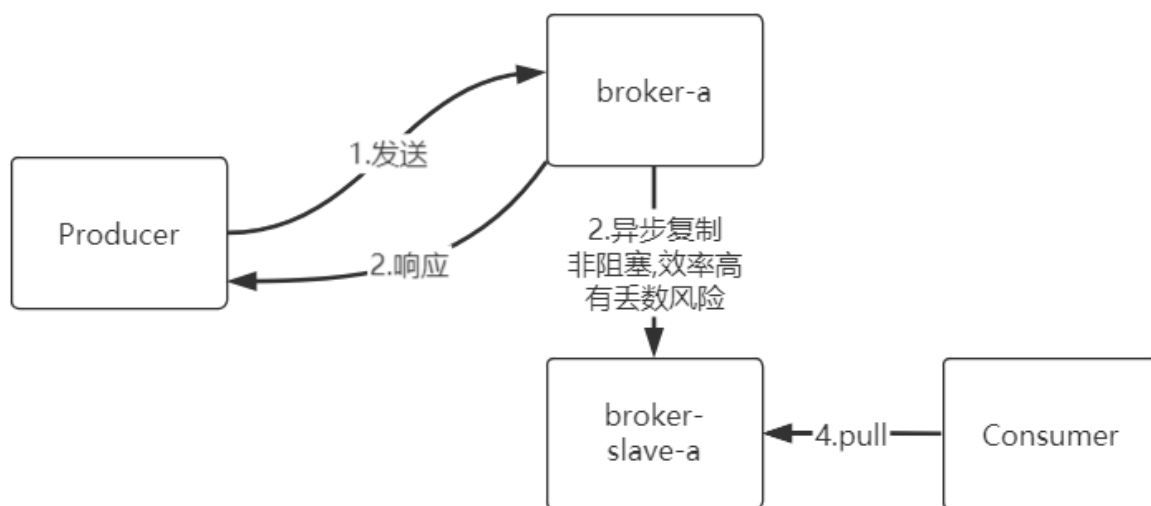
消息已发送: MsgId:C0A81F6D264418B4AAC26FA3EF530061, 发送状态:SEND_OK

消息已发送: MsgId:C0A81F6D264418B4AAC26FA3EF550062, 发送状态:SEND_OK

消息已发送: MsgId:C0A81F6D264418B4AAC26FA3EF580063, 发送状态:SEND_OK

同步复制与异步复制的区别





异步复制非阻塞