

# RocketMQ源码阅读（2）



Chown · 1 个月前

在（1）中大概理了一下每个组件是如何交互的，这里看看Consumer拉取消息时的具体过程，



关注专栏

写文章

FullConsumer小例代码

RocketMQ项目本身包含了这些example。

```
public class PullConsumer {  
    /**  
     * 记录消费每个消息队列的进度  
     */  
    private static final Map<MessageQueue, Long> offseTable = new HashMap<Mess  
  
    public static void main(String[] args) throws MQClientException {  
        DefaultMQPullConsumer consumer = new DefaultMQPullConsumer("ConsumerGr  
        consumer.setNamesrvAddr("127.0.0.1:9876");  
    }  
}
```

```

consumer.setInstanceName("Consumer");

consumer.start();
/**
 * 获取所有存在该topic的消息队列(broker + queueId)
 */
Set<MessageQueue> mqs = consumer.fetchSubscribeMessageQueues("MyTopic1");
for (MessageQueue mq : mqs) {
    System.out.println("Consume from the queue: " + mq);
    SINGLE_MQ: while (true) {
        try {
            PullResult pullResult =
                consumer.pullBlockIfNotFound(mq, null, getMessageQ
            System.out.println(pullResult);
            putMessageQueueOffset(mq, pullResult.getNextBeginOffset());
            switch (pullResult.getPullStatus()) {
                case FOUND:
                    break;
                case NO_MATCHED_MSG:
                    break;
                case NO_NEW_MSG:
                    break SINGLE_MQ;
                case OFFSET_ILLEGAL:
                    break;
                default:
                    break;
            }
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}

consumer.shutdown();
}

/** 需要消费者自己记录消费的偏移量 因为消息在broker上是持久的 (定期删除) */
private static long getMessageQueueOffset(MessageQueue mq) {
    Long offset = offseTable.get(mq);
    if (offset != null)
        return offset;

    return 0;
}

```

```
    }
```

```
}
```

## 获取要读取的消息队列

其实就是根据topic的路由信息，找到对应的broker，然后那些broker上的read queue就是我们要查询消息的地方。

```
/**
 * 根据 topic 得到需要订阅的消息队列信息
 */
public Set<MessageQueue> fetchSubscribeMessageQueues(String topic) throws MQCl
    try {
        TopicRouteData topicRouteData = this.mQClientFactory.getMQClientAPIImp
        if (topicRouteData != null) {
            Set<MessageQueue> mqList = MQClientInstance.topicRouteData2TopicSu
            if (!mqList.isEmpty()) {
                return mqList;
            }
            else {
                throw new MQClientException("Can not find Message Queue for th
            }
        }
    }
    catch (Exception e) {
        throw new MQClientException(
            "Can not find Message Queue for this topic, " + topic + FAQUrl.sug
            e);
    }

    throw new MQClientException("Unknow why, Can not find Message Queue for th
}

/**
 * 把 topic route info 转化为 subscribe info
 * 其实就是为了得到 topic 所在的 broker 上的所有读队列信息
 */
public static Set<MessageQueue> topicRouteData2TopicSubscribeInfo(final String
    Set<MessageQueue> mqList = new HashSet<MessageQueue>();
    List<QueueData> qds = route.getQueueDatas();
    for (QueueData qd : qds) {
```

```

        if (PermName.isReadable(qd.getPerm())) {
            for (int i = 0; i < qd.getReadQueueNums(); i++) {
                MessageQueue mq = new MessageQueue(topic, qd.getBrokerName(),
                    mqList.add(mq);
            }
        }
    }

    return mqList;
}

```

## 从消息队列拉取消息

有了目的地后就进行消息读取操作，可以看到这里的设计，每次读取一个消息队列的时候可以  
进行批量操作（读取一个block），减小网络开销。

```

/**
 * 从消息队列 mq 中拉取 从 offset 开始的 maxNums个消息
 */
private PullResult pullSyncImpl(MessageQueue mq, String subExpression, long of
    boolean block, long timeout) throws MQClientEx
        InterruptedException {
    this.makeSureStateOK();

    // 边界情况处理

    this.subscriptionAutomatically(mq.getTopic());

    int sysFlag = PullSysFlag.buildSysFlag(false, block, true, false);

    SubscriptionData subscriptionData;
    try {
        subscriptionData = FilterAPI.buildSubscriptionData(this.defaultMQPullC
            mq.getTopic(), subExpression);
    } catch (Exception e) {
        throw new MQClientException("parse subscription error", e);
    }

    long timeoutMillis =
        block ? this.defaultMQPullConsumer.getConsumerTimeoutMillisWhenSus

    PullResult pullResult = this.pullAPIWrapper.pullKernelImpl(//

```

```

        mq, // 1
        subscriptionData.getSubString(), // 2
        0L, // 3
        offset, // 4
        maxNums, // 5
        sysFlag, // 6
        0, // 7
        this.defaultMQPullConsumer.getBrokerSuspendMaxTimeMillis(), // 8
        timeoutMillis, // 9
        CommunicationMode.SYNC, // 10
        null // 11
    );

    return this.pullAPIWrapper.processPullResult(mq, pullResult, subscriptionD
}

/**
 * 构造请求参数
 */
public PullResult pullKernelImpl(//
    final MessageQueue mq, // 1
    final String subExpression, // 2
    final long subVersion, // 3
    final long offset, // 4
    final int maxNums, // 5
    final int sysFlag, // 6
    final long commitOffset, // 7
    final long brokerSuspendMaxTimeMillis, // 8
    final long timeoutMillis, // 9
    final CommunicationMode communicationMode, // 10
    final PullCallback pullCallback // 11
) throws MQClientException, RemotingException, MQBrokerException, InterruptedE
    FindBrokerResult findBrokerResult =
        this.mQClientFactory.findBrokerAddressInSubscribe(mq.getBrokerName
            this.recalculatePullFromWhichNode(mq), false);
    if (null == findBrokerResult) {
        this.mQClientFactory.updateTopicRouteInfoFromNameServer(mq.getTopic())
        findBrokerResult =
            this.mQClientFactory.findBrokerAddressInSubscribe(mq.getBroker
                this.recalculatePullFromWhichNode(mq), false);
    }

    if (findBrokerResult != null) {
        int sysFlagInner = sysFlag;

```

```
if (findBrokerResult.isSlave()) {
    sysFlagInner = PullSysFlag.clearCommitOffsetFlag(sysFlagInner);
}

PullMessageRequestHeader requestHeader = new PullMessageRequestHeader(
    requestHeader.setConsumerGroup(this.consumerGroup);
    requestHeader.setTopic(mq.getTopic());
    requestHeader.setQueueId(mq.getQueueId());
    requestHeader.setQueueOffset(offset);
    requestHeader.setMaxMsgNums(maxNums);
    requestHeader.setSysFlag(sysFlagInner);
    requestHeader.setCommitOffset(commitOffset);
    requestHeader.setSuspendTimeoutMillis(brokerSuspendMaxTimeMillis);
    requestHeader.setSubscription(subExpression);
    requestHeader.setSubVersion(subVersion);

String brokerAddr = findBrokerResult.getBrokerAddr();
if (PullSysFlag.hasClassFilterFlag(sysFlagInner)) {
    brokerAddr = computePullFromWhichFilterServer(mq.getTopic(), broker
}

// 发送请求到broker, 可以同步或异步发送
PullResult pullResult = this.mQClientFactory.getMQClientAPIImpl().pull
    brokerAddr,//
    requestHeader,//
    timeoutMillis,//
    communicationMode,//
    pullCallback);

return pullResult;
}

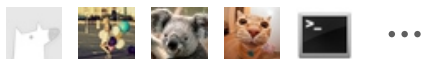
throw new MQClientException("The broker[" + mq.getBrokerName() + "] not ex
}
```

## 小结

- 可以带着问题去阅读源码
- 也可以只是为了看看别人是怎么设计的

[消息队列](#)[Java Web](#)[源码阅读](#)

9

[分享](#)[举报](#)

## 文章被以下专栏收录

**编程之路**[进入专栏](#)

## 还没有评论



## 推荐阅读

### Storm安装

因为畏惧，我们停滞不前。下载需要的包1.ZooKeeper2.Storm配置ZooKeeper把ZooKeeper对应的conf目录下的zoo-sample.cfg重命名为zoo.cfg，配置... [查看全文](#) >

Chown · 1 个月前

发表于 编程之路



### ExceptionHandler 的一点理解

前我在Spring MVC Controller 中处理异常一般如下，当逻辑复杂之后代码变得相当复杂，逻辑里面又掺杂着各种校验，异常处理等等。@Controller public class SimpleControllerU... [查看全文](#) >

Chown · 2 个月前

发表于 编程之路

## 西班牙足球地理：太阳海岸奔放热情

这里是，欧陆西南边陲，三围濒海之地，漫长的海岸线自地中海之滨与大西洋交汇，阳光和沙滩，涤荡着太阳国度的山河原野与风物人情。这里是，哥... [查看全文](#) >

冯浩然 · 5 天前 · 编辑精选

发表于 绿洲足球

## 你越深入，这个世界就越复杂：古巴人的第一个独立游戏《救星》

我们的游戏诞生在这个古巴史上最特别的时刻。因特网正在岛上普及，我们和美国的关系正在发展，经济正在复苏。大家都对未来很乐观，但我们并不... [查看全文](#) >

楚乐玫 · 2 天前 · 编辑精选

发表于 触乐-移动游戏新媒体