

消息队列 RocketMQ

控制台使用指南

控制台使用指南

消息查询

如遇消息消费有问题，则可通过查询具体发送的消息内容来排查问题。消息队列 RocketMQ 提供了三种消息查询的方式，分别是按 Message ID、Message Key 以及 Topic 查询。

查询方式说明

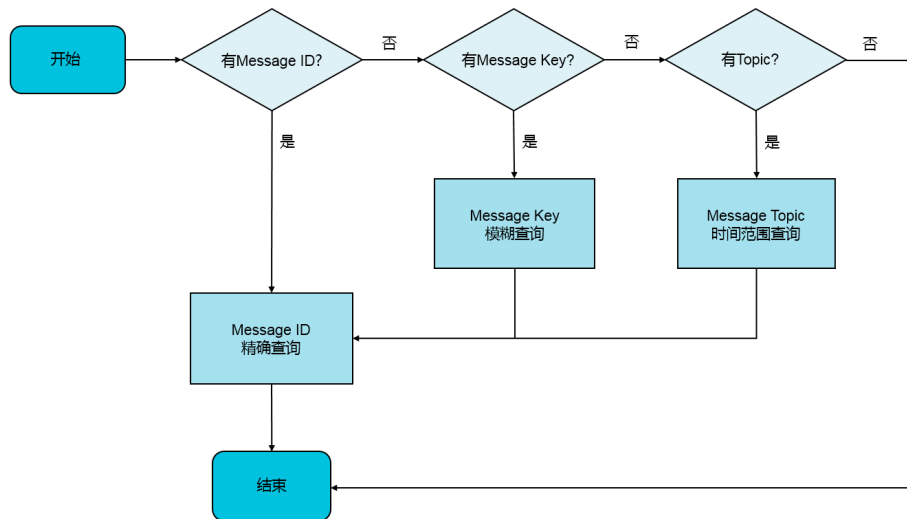
由于消息在消息队列 RocketMQ 中存储的时间为 3 天，即只能查询从当前查询时间算起 3 天内的消息。例如，当前时间是 2019 年 6 月 10 日 15 : 09 : 48，那么能查询到的某 Topic 下的消息最早的时间点为 2019 年 6 月 7 日 15 : 09 : 48。

三种查询方式的特点和对比如下表所述。

查询方式	查询条件	查询类别	说明
按 Message ID 查询	Topic+Message ID	精确查询	根据 Topic 和 Message ID 可以精确定位任意一条消息，获取消息的属性。
按 Message Key 查询	Topic+Message Key	模糊查询	根据 Topic 和 Message Key 可以匹配到包含指定 Key 的最近 64 条消息。 注意： 建议消息生产方为每条消息设置尽可能唯一的 Key，以确保相同的 Key 的消息不会超过 64 条，否则消息会漏查。
按 Topic 查询	Topic+时间段	范围查询	根据 Topic 和时间范围，批量获取符合条件的所有消息，查询量大，不易匹配。

推荐查询过程

推荐按照以下流程查询消息。



操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如华北1（杭州）。

在左侧导航栏，单击**消息查询**。

在**消息查询**页面，您可单击以下任一页签，然后按页面提示输入相应信息，再单击**搜索**按钮来查询消息。

按 Message ID 查询

按 Message ID 查询消息属于精确查询，您输入 Topic 和 Message ID 即可精确查询到任意一条消息。因此，为了尽可能精确地查询，建议在发送消息成功后将 Message ID 信息打印到日志中，方便问题排查。

以 Java SDK 为例，获取 Message ID 的方法如下：

```
SendResult sendResult = producer.send(msg);
String msgId = sendResult.getMessageId();
```

其他语言可参见 **Group 管理**页面的**示例代码**获取。

按 Message Key 查询

消息队列 RocketMQ 根据您的 Message Key 建立消息的索引信息，当您输入 Key 进行查询时

，消息队列 RocketMQ 根据该索引即可匹配相关的消息返回。

注意：

若按 Message Key 查询消息，请注意以下几点：

按 Message Key 查询的条件是用户设置的 Message Key 属性。

按 Message Key 查询仅仅返回符合条件的最近的 64 条消息，因此建议您尽可能保证设置的 Key 是唯一的，并具有业务区分度。

设置 Message Key 的方法如下：

```
Message msg = new Message("Topic", "*", "Hello MQ".getBytes());  
/**  
 * 对每条消息设置其检索的 Key，该 Key 值代表消息的业务关键属性，请尽可能全局唯一。  
 * 以方便您在无法正常收到消息情况下，可通过消息队列 RocketMQ 控制台查询消息。不设置也不会影响消息正常收发。  
 */  
msg.setKey("TestKey"+System.currentTimeMillis());
```

按 Topic 查询

按 Topic 查询一般用在 Message ID 和 Message Key 都无法获得的情况下，根据 Topic 和消息的发送时间范围，批量获取该时间范围内的所有消息，然后再找到关心的数据。

注意：

若按 Topic 查询消息，请注意以下几点：

按 Topic 查询属于范围查询，获取 Topic 下符合时间条件的所有消息，消息量大，建议尽量缩短查询区间。

按 Topic 查询数据量大，采用分页展示。

查询结果说明

您可以在控制台的**消息查询**页面看到查询到的消息。直接显示的信息包含 Message ID、Tag、Key 和存储时间。此外，您还可以在每一行消息的**操作**列下载消息内容、查询消息轨迹以及查看消息详情。

查看消息详情

其中，投递状态是消息队列 RocketMQ 根据各个 Group ID 的消费进度计算出的结果，投递状态的信息参考以

下表格。

注意：投递状态仅仅是根据消费进度估算的结果，如果需要详细的消费信息，请使用消息轨迹工具查询。消息轨迹可以展示该条消息的完整链路，详情参见[查询消息轨迹](#)。

投递状态表

投递状态	说明
已订阅，并且消息至少已被消费一次	该 Group ID 已经正常消费过这条消息。
已订阅，但消息被过滤表达式过滤，请查看 Tag	该消息的 Tag 不符合消费方的订阅关系，消息被过滤，即未被消费。您可以进入控制台中的 Group 管理 ，然后单击该 Group ID 的 操作列的消费者状态 查询其订阅关系。
已订阅，但消息未被消费	该 Group ID 订阅了该消息，但还未消费。有可能是消费过慢，或者消费出现异常导致阻塞。
已订阅，但 Group ID 当前不在线，请通过消息轨迹功能进行精确查询	该 Group ID 订阅了该消息，但是不在线。请检查消费者端应用不在线的原因。
未知异常	出现未收录的异常情况。请提交工单进行解决。

消费验证

消息队列 RocketMQ 提供了消费验证功能，该功能可以将指定消息推送给指定的在线客户端，以检测客户端消费该消息的逻辑和结果是否符合预期。

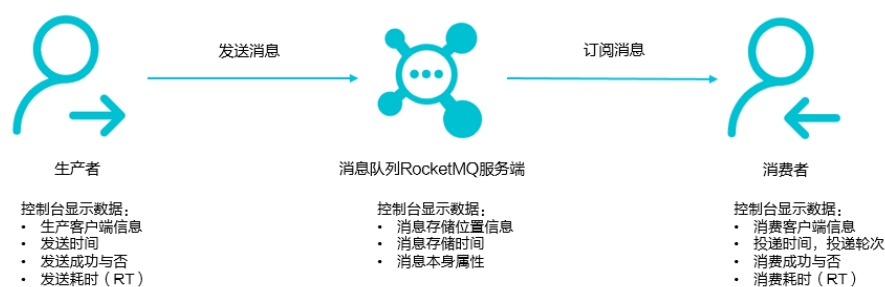
注意：消费验证功能仅仅是用于验证客户端的消费逻辑是否正常，并不会影响正常的收消息流程，因此消息的消费状态等信息在消费验证后并不会改变。

查询消息轨迹

消息轨迹是指一条消息从生产者发送到消息队列 RocketMQ 服务端，再到消费者消费处理，整个过程中的各个相关节点的时间、状态等数据汇聚而成的完整链路信息。该轨迹可作为生产环境中排查问题强有力的数据支持。本文介绍消息轨迹的使用场景、查询步骤以及查询结果的参数说明。

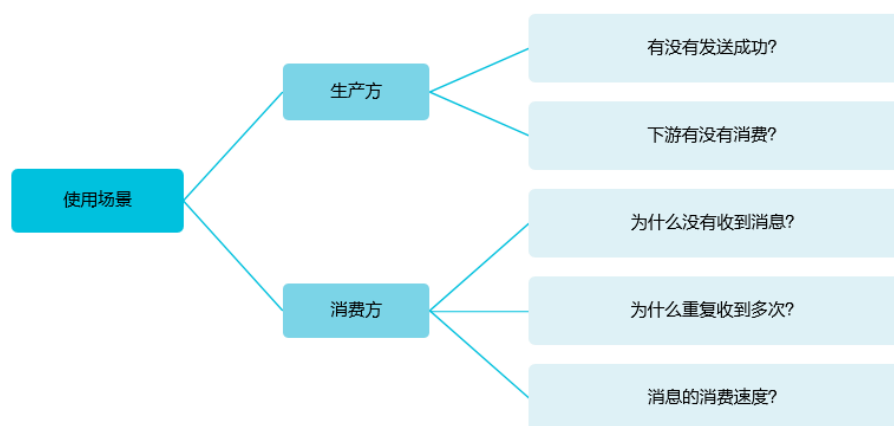
消息轨迹数据

消息队列 RocketMQ 系统中，一条消息的完整链路包含生产者、服务端、消费者三个角色，每个角色处理消息的过程中都会在轨迹链路中增加相关的信息，将这些信息汇聚即可获取任意消息当前的状态，涉及的数据如下图所示。



使用场景

在生产环境的消息收发不符合预期时, 可以通过 Message ID、Message Key 或 Topic 的时间范围查询相关的消息轨迹, 找到消息的实际收发状态, 帮助诊断问题。



使用示例

假设您根据业务日志里的信息判断某条消息一直没有收到, 则可以参照以下步骤, 利用消息轨迹来排查问题。

收集疑似有问题的消息信息, 包括 Message ID、Message Key、Topic 以及大概的发送时间。

进入消息队列 RocketMQ 控制台, 根据已有的信息创建查询任务, 查询相关的消息的轨迹。

查看结果并分析判断原因。

如果轨迹显示**尚未消费**, 则可以前往 **Group 管理** 页面查看消费者状态, 确认消息是否堆积而导致尚未消费。

如果发现已经消费, 请根据消费端的信息, 找到对应的客户端机器和时间, 登录客户端机器查看相关日志。

使用说明

消息轨迹的使用不会增加额外的接入成本。所有类型的消息正常发送后，即可根据消息的属性在消息队列 RocketMQ 控制台上查询到消息的发送轨迹，但消费轨迹需要注意以下几点：

消息类型	查询说明
普通消息	没消费前显示 尚未消费 。消费后会展示投递和消费信息。
顺序消息	没消费前显示 尚未消费 。消费后会展示投递和消费信息。
定时和延时消息	如果当前系统时间没有到达指定消费的时间，轨迹可以查询到，但是消息查询不到。
事务消息	事务未提交之前，轨迹可以查询到，但是消息查询不到。

查询消息轨迹

前提条件

请确保您的 SDK 版本支持消息轨迹功能，具体说明如下：

TCP 协议下，支持通过 Message ID、Message Key 或 Topic 的时间范围查询相关的消息轨迹。

Java SDK 版本说明：v1.2.7 及以上版本

C/C++ SDK 版本说明：v1.1.2 及以上版本

.NET SDK 版本说明：v1.1.2 及以上版本

HTTP 协议下，仅支持通过 Message ID 查询相关的消息轨迹。

除 Node.js SDK 需为 v1.0.2 及以上版本外，其余语言的 SDK 均需为 v1.0.1 及以上版本。详情请参见消息队列 RocketMQ HTTP SDK 库。

您的消息已从生产者发送。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择所需查询的消息所在实例的地域（Region），如华东1（杭州）。

在左侧导航栏，单击**消息轨迹**，然后在消息轨迹的任务列表的右上角，单击**创建查询任务**按钮。

在弹出的**创建查询任务**对话框中，按需选择查询条件并按提示输入信息，单击**确认**完成创建。

注意：查询时，尽可能地精确设置**时间范围**，以便缩小查询范围，提高速度。

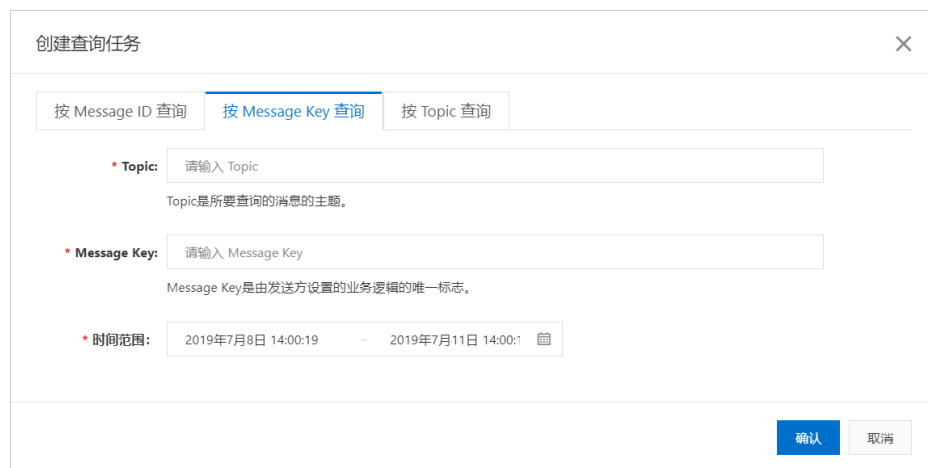
消息轨迹查询功能支持三种查询方式，具体说明如下：

按 Message ID 查询：该方式属于精确查询，速度快，精确匹配，推荐使用。



The screenshot shows the '创建查询任务' (Create Query Task) dialog box. The '按 Message ID 查询' (Query by Message ID) tab is selected. It contains three input fields: 'Topic' (with placeholder '请输入 Topic'), 'Message ID' (with placeholder '请输入 Message ID'), and '时间范围' (Time Range) (with values '2019年6月28日 15:37:02' and '2019年7月1日 15:37:02'). Below each field is a descriptive text: 'Topic是所要查询的消息的主题。', 'Message ID是由发送方Send方法返回的32位Hex字符串。', and '时间范围' is followed by a calendar icon. At the bottom right are '确认' (Confirm) and '取消' (Cancel) buttons.

按 Message Key 查询：该方式属于模糊查询，最多查询 1000 条轨迹。仅适用于您没有记录 Message ID 但是设置了 Message Key，同时 Message Key 具有区分度的情况。



The screenshot shows the '创建查询任务' (Create Query Task) dialog box. The '按 Message Key 查询' (Query by Message Key) tab is selected. It contains three input fields: 'Topic' (with placeholder '请输入 Topic'), 'Message Key' (with placeholder '请输入 Message Key'), and '时间范围' (Time Range) (with values '2019年7月8日 14:00:19' and '2019年7月11日 14:00:19'). Below each field is a descriptive text: 'Topic是所要查询的消息的主题。', 'Message Key是由发送方设置的业务逻辑的唯一标志。', and '时间范围' is followed by a calendar icon. At the bottom right are '确认' (Confirm) and '取消' (Cancel) buttons.

按 Topic 查询：该方式属于范围查询，适用于没有上述 Message ID 和 Message Key，而且消息量比较小的场景。因为时间范围内消息很多，且不具备区分度，所以不推荐使用。

创建查询任务

按 Message ID 查询

按 Message Key 查询

按 Topic 查询

Topic: 请输入 Topic

Topic是所要查询的消息的主题。

时间范围: 2019年7月8日 14:00:19 - 2019年7月11日 14:00:1

确认

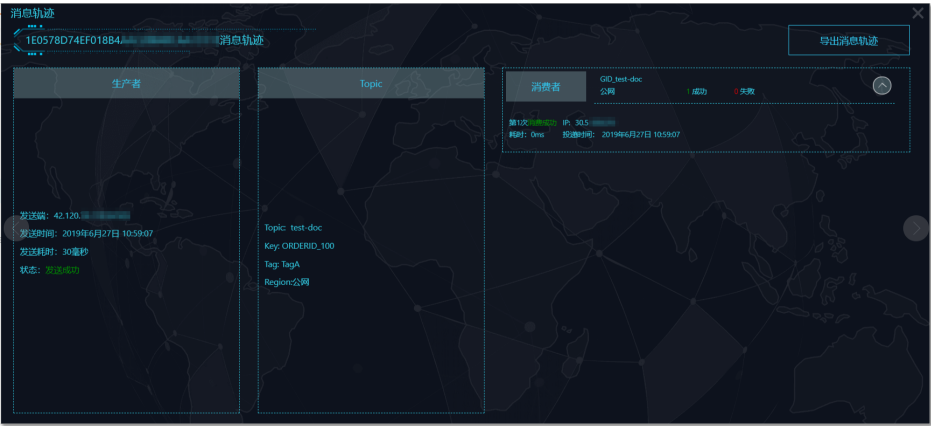
取消

您创建查询任务后，即可在消息轨迹页面查看到刚创建的查询任务，且任务状态显示为**查询中**，说明暂不能查看消息轨迹。

在消息轨迹的任务列表的右上角，单击**刷新**按钮，直到状态切换至**查询完成**。单击图标可查看到轨迹的简要信息，主要是消息本身的属性以及接收状态的信息。

Topic	Message ID		Message Key	任务状态	创建时间	操作
test-doc	1E0578D74EF018B4AC28840CAa1311E			查询完成	2019年6月27日 10:59:48	— 刷新
Message ID	Tag	Message Key	发送状态	发送时间	消费状态	操作
1E0578D74EF018B4AC28840CA...	TagA	ORDERID_100	发送成功	2019年6月27日 10:59:07	全部成功	查看轨迹

在展开的区域框中，单击**查看轨迹**即可查看完整的链路图。下图示例为在 TCP 协议下，按 Message ID 查询普通消息的轨迹。



对于 Message Key 和 Topic 查询方式，如果匹配到多条轨迹，可以进行上下翻页，查看比对轨迹数据。

查询结果说明

消息轨迹名词解释

消息轨迹链路图中的部分关键字段的解释如下。

链路部分	字段	说明
------	----	----

生产者	发送时间	消息从生产者发送时的客户端时间戳
	发送耗时	生产者调用 Send 方法发送消息的毫秒耗时
Topic	Region	消息存储的地域
消费者	耗时	消息推送到客户端之后执行 consumeMessage 方法的耗时
	投递时间	客户端执行 consumeMessage 方法开始消费消息时的时间戳

状态信息说明

在消息轨迹的任务列表页，单击某任务的  图标会显示消息的**发送状态**和**消费状态**；而消息轨迹链路图中的**生产者**和**消费者**信息区域框也会分别显示消息的发送和消费状态。这些状态字段及其解释如下。

发送/消费状态	状态字段	说明
发送状态	发送成功	消息发送成功，服务端已经存储成功
	发送失败	消息发送失败，服务端没有存储消息，需要重试
	消息定时中	该消息是定时或者延时消息，且尚未到达投递时间
	事务未提交	该消息是事务消息，且尚未提交状态
	事务回滚	该消息是事务消息，并且已经回滚
消费状态	全部成功	该消息的所有投递都已成功消费
	部分成功	该消息投递中存在消费失败的情况，或消费失败并重试成功的情况
	全部失败	该消息的所有投递都消费失败
	尚未消费	该消息尚未投递给任何消费方
	消费结果未返回	消费消息的方法尚未返回结果，或者被中断，导致本次消费结果未传回服务端
	消费成功	该消息已被成功消费
	消费失败	消费消息的方法主动返回失败标志，或者是消费方法抛异常

更多信息

如需删除某个查询任务，可在消息轨迹任务列表页找到需删除的任务，在其**操作**列，单击  图标，按提示完成删除。

如果对消息轨迹的查询结果有疑问，请查看常见问题中的消息轨迹。

查看消费者状态

如遇消息消费异常，可在控制台查看消费者状态初步诊断原因。您可查看每个 Group ID 代表的消费者实例群组的信息，也可查看该群组下单个消费者实例的信息。

应用场景

消费者状态信息包括在线状态、订阅关系、消费 TPS、消息堆积量和线程堆栈信息等，经常被应用于以下故障排查场景。

订阅关系不一致

现象：在**消费者状态**页面，看到 Group ID 的**订阅关系是否一致**一栏显示否。

处理：订阅关系不一致的处理方法，请参见收到订阅关系不一致告警后怎么办。

消息堆积

现象：在**消费者状态**页面，看到 Group ID 的**实时消息堆积量**一栏的值高于预期。

处理：处理消息堆积的具体办法，请参见如何处理消息堆积。在顶部导航栏，选择地域（Region），如**华北1（杭州）**。

查看 Group ID 综合信息

请按以下步骤查看 Group ID 所代表的消费者实例集群的综合信息：

登录消息队列 RocketMQ 控制台。

在左侧导航栏单击 **Group 管理**。

在 **Group 管理** 页面，找到要查看的 Group ID，单击其右侧**操作**列中的**消费者状态**，查看结果如下图所示。



字段说明

该页面的主要字段的说明如下表所示。

字段/图标	说明
在线状态图标	只要该 Group ID 下有一个消费者实例在线，就显示 在线 ，此时您可以看到页面下方的 连接信息 区域框显示了在线的消费者实例信息；若该 Group ID 下所有消费者实例都不在线，则显示 离线 ，且 连接信息 区域框无信息显示。
订阅关系是否一致	显示该 Group ID 下消费者实例群组的订阅关系是否一致。订阅关系一致的概念请参见 订阅关系一致 。
实时消费速度	该 Group ID 下消费者实例群组接收消息的总 TPS，单位为“条/秒”。
实时消息堆积量	该 Group ID 下消费者实例群组的未消费消息的总量。
最近消费时间	该 Group ID 下消费者实例群组最近一次消费消息的时间。
消息延迟时间	该 Group ID 下消费者实例群组最早的一条未消费消息的生产时间与当前时间的差值。

查看 Group ID 下单个消费者实例的信息

若此时该 Group ID 的在线状态为**在线**，则在**连接信息**区域框您可以查看此时在线的每个消费者实例的具体信息，包括客户端 ID、宿主机 IP/公网 IP、当前进程 ID 和消息堆积量等。

查看更多信息详细信息

除上述信息，还可在某个消费者实例信息右侧的**详细说明**列，单击**详细信息**查看更多信息详细信息，包括消费线程

数、消费启动时间、订阅关系和消费统计数据等，如下图所示。

消费者状态 > 连接详情

返回

连接状态

消费者名称

CID-test-topic

消费类型

PUSH

消费线程数

20

消费启动时间

2019年4月29日 19:26:55

订阅关系

Topic	Tag
test-test- test-topic	*

消费统计

Topic	业务处理时间（毫秒/条）	消息成功(条/秒)	消息失败(条/秒)	消息堆积量
test-test- test-topic	0	0	0	0

查看线程堆栈信息

如需查看某消费者实例当前进程的堆栈信息，找到需查看堆栈信息的消费者实例，在其所在行右侧的**堆栈信息**列，单击**堆栈信息**，展现的信息如下图所示。

消费者状态		
Java Stack		
Thread		Stack
RobalenceService		TID: 17 STATE: TIMED_WAITING run (mex.Lmash.parkNative Method) java.util.concurrent.locks.LockSupport.park(NanoTime)(LockSupport.java:215) java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1037) java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1028) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.CountDownLatch2.java(CountDownLatch2.java:114) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.ServiceThread.waitForRunning(ServiceThread.java:110) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.client.impl.consumer.RebalanceService.run(RebalanceService.java:40) java.lang.Thread.run(Thread.java:748)
		TID: 46 STATE: WAITING run (mex.Lmash.parkNative Method) java.util.concurrent.locks.LockSupport.park(LockSupport.java:175) java.util.concurrent.locks.AbstractQueuedSynchronizer.ConditionObject.await(AbstractQueuedSynchronizer.java:2039) java.util.concurrent.LimitedReleaserQueue.await(LimitedReleaserQueue.java:442) java.util.concurrent.ThreadPoolExecutor.getTask(ThreadPoolExecutor.java:1054) java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1134) java.lang.Thread.run(Thread.java:748)
ConsumerMessageThread_1		TID: 32 STATE: TIMED_WAITING run (mex.Lmash.parkNative Method) java.util.concurrent.locks.LockSupport.park(NanoTime)(LockSupport.java:215) java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1037) java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1028) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.CountDownLatch2.java(CountDownLatch2.java:114) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.ServiceThread.waitForRunning(ServiceThread.java:110) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.client.impl.consumer.RebalanceService.run(RebalanceService.java:40) java.lang.Thread.run(Thread.java:748)
RobalenceService		TID: 32 STATE: TIMED_WAITING run (mex.Lmash.parkNative Method) java.util.concurrent.locks.LockSupport.park(NanoTime)(LockSupport.java:215) java.util.concurrent.locks.AbstractQueuedSynchronizer.doAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1037) java.util.concurrent.locks.AbstractQueuedSynchronizer.tryAcquireShared(NanoTime)(AbstractQueuedSynchronizer.java:1028) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.CountDownLatch2.java(CountDownLatch2.java:114) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.common.ServiceThread.waitForRunning(ServiceThread.java:110) com.alibabab.ospemrvice.shade.com.alibabab.rocketmq.client.impl.consumer.RebalanceService.run(RebalanceService.java:40) java.lang.Thread.run(Thread.java:748)

更多信息

如果您设置过监控项，且接收到消息堆积告警通知，请参见[收到堆积告警后怎么办处理](#)。

重置消费位点

您可通过重置消费位点，按需清除堆积的或不想消费的这部分消息再开始消费，或直接跳转到某个时间点消费。

该时间点之后的消息（不论是否消费过该时间点之前的消息）。

使用重置消费位点功能有以下注意事项：

广播消费模式不支持重置消费位点。

目前不支持指定 Message ID、Message Key 和 Tag 来重置消息的消费位点。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如华北1（杭州）。

在左侧导航栏，单击 **Group 管理**。

找到需要重置消费位点的 Group ID 和针对的 Topic，在该行的**操作**列中单击  图标，在快捷菜单中，选择**重置消费位点**。

在**重置消费位点**对话框中，按需选择以下选项。

清除所有堆积消息，从最新位点开始消费：若选择此项，该 Group ID 在消费该 Topic 下的消息时会跳过当前堆积（未被消费）的所有消息，从这之后发送的最新消息开始消费。

对于程序返回 “reconsumeLater”，即走重试流程的这部分消息来说，清除无效。

按时间点进行消费位点重置：选择该选项后会出现时间点选择的控件。请选择一个时间点，这个时间点之后发送的消息才会被消费。

可选时间范围中的起始和终止时间分别是该 Topic 中储存的最早的和最晚的一条消息的生产时间。不能选择超过可选时间范围的时间点。

单击**确认**执行消费位点重置。

死信队列

死信队列用于处理无法被正常消费的消息，即死信消息。本文通过介绍如何查询、导出和重新发送进入死信队列的死信消息，以便您按需管理死信消息，避免消息漏处理。

背景信息

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）。

特性说明

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。

消息队列 RocketMQ 控制台提供对死信消息的查询、导出和重发的功能。

查询死信消息

消息队列 RocketMQ 提供的查询死信消息的方式对比如下表所示。

查询方式	查询条件	查询类别	说明
按 Group ID 查询	Group ID+时间段	范围查询	根据 Group ID 和时间范围，批量获取符合条件的所有消息；查询量大，不易匹配。
按 Message ID 查询	Group ID+Message ID	精确查询	根据 Group ID 和 Message ID 可以精确定位任意一条消息。

按 Group ID 查询

您可以根据 Group ID 和死信消息产生的时间范围，批量查询该 Group ID 在某段时间内产生的所有死信消息。

注意：死信消息产生的时间是指一条消息在投递重试达到最大次数后被发送到死信队列的时间。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如**华北1（杭州）**。

在左侧导航栏，单击**死信队列**。

在**死信队列**页面，单击实例，然后再单击**按 Group ID 查询**页签。

在 Group ID 的输入框中，输入关键词搜索想要查找的 Group ID，然后单击选中，或直接在下拉列表中选择想要查找的 Group ID。

单击时间选择框，选择想要查询的时间范围的起止日期和具体时间点。

单击**搜索**按钮。页面中会显示所有符合以上条件的死信消息。

找到想要查看的死信消息，单击其**操作**列中的**查看详情**查看该消息的详细信息，包括其基本属性、消息体的下载链接、消息轨迹和投递状态。

按 Message ID 查询

按 Message ID 查询消息属于精确查询。您可以根据 Group ID 与 Message ID 精确查询到任意一条消息。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域，如**华北1（杭州）**。

在左侧导航栏，单击**死信队列**。

在**死信队列**页面，单击实例，然后再单击**按 Message ID 查询**页签。

在 Group ID 的输入框中，输入关键词搜索想要查找的 Group ID，然后单击选中，或直接在下拉列表中选择想要查找的 Group ID。

在 Message ID 的输入框中输入想要查找的消息的 Message ID。

单击**搜索**按钮。页面中会显示所有符合以上条件的死信消息。

找到想要查看的死信消息，单击其**操作**列中的**查看详情**查看该消息的详细信息，包括其基本属性、消息体的下载链接、消息轨迹和投递状态。

导出死信消息

若您暂时无法处理死信消息，可以在消息队列 RocketMQ 控制台上将其导出，以免超过有效期。

消息队列 RocketMQ 控制台提供对死信消息的**单条导出**和**批量导出**功能。导出的文件格式为 CSV。

导出的消息内容如下表所示。

消息字段	字段含义
topic	消息所属的 Topic
msgId	消息的 ID
bornHost	消息产生的地址
bornTimestamp	消息产生的时间
storeTimestamp	死信消息产生的时间
reconsumeTimes	消费失败的次数
properties	消息属性；JSON 格式
body	消息体；base64 编码格式
bodyCRC	消息体 CRC

单条导出

在消息队列 RocketMQ 控制台按任意方式查询到死信消息后，在某条死信消息的**操作列**，单击**导出**，导出该条死信消息。

批量导出

在消息队列 RocketMQ 控制台按 Group ID 查询到死信消息后，勾选目标死信消息，然后单击页面下方的**批量导出**按钮，导出所有勾选的死信消息。

重新发送死信消息

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，您可以在消息队列 RocketMQ 控制台重新发送该消息，让消费者重新消费一次。

注意：死信消息被重新发送后，不会在死信队列中被立即删除。

单条重发

在消息队列 RocketMQ 控制台按任意方式查询到死信消息后，在某条死信消息的**操作列**，单击**重新发送**，重发该条死信消息。

批量重发

在消息队列 RocketMQ 控制台按 Group ID 查询到死信消息后，勾选目标死信消息，然后单击页面下方的**批量重发**按钮，重新发送所有勾选的死信消息。

全球消息路由

本文通过介绍什么是全球消息路由以及全球消息路由任务管理，以便您按需同步消息，实现数据跨地域（Region）的一致性。

什么是全球消息路由

消息队列 RocketMQ 的全球消息路由功能常用于不同地域之间同步所有或按过滤规则（例如通过特定的 Tag 过滤）过滤后的消息，保证地域之间的数据一致性。该功能依托阿里云优质基础设施实现的高速通道专线（高速通道无需另行开通），高效地实现国内外不同地域之间的消息同步复制。具体实现如下图所示。



注意：

不支持传递路由，比如地域 A 到地域 B 再到地域 C，地域 A 的消息不会经过地域 B 再路由到地域 C，如有需要请直接创建地域 A 到地域 C 的路由。

目前仅公网地域不支持全球消息路由功能，其他地域均支持。

源 Topic 和目标 Topic 的消息类型需保持一致，如果源 Topic 是普通消息，则同步到的目标 Topic 也应是普通消息。

考虑到需要同步的消息量以及网络带宽瓶颈，消息同步可能会产生一定的时间延迟，即消息到达目标 Topic 的时间会晚于该消息进入源 Topic 的时间。

创建全球消息路由任务

您可以登录消息队列 RocketMQ 控制台，根据业务需求创建跨地域的全球消息同步任务。

前提条件

创建全球消息路由任务前，请确保需同步消息的源地域和目标地域支持全球消息路由功能。

源实例和源 Topic 以及目标实例和目标 Topic 已创建并可用，且源 Topic 和目标 Topic 的消息类型保持一致。

关于创建实例和 Topic 的具体步骤，请根据您所使用的账号类型参见主账号 - 快速入门或子账号 - 快速入门中的创建资源部分。

操作步骤

请按照以下步骤创建全球消息路由任务：

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域，如**华北1（杭州）**。

在左侧导航栏，单击**全球消息路由**。

在**全球消息路由**页面的右上方，单击**创建任务**按钮。

在**创建任务**对话框，填写以下信息后，单击**确认**。您需填写的信息如下表所示。

参数	说明
源地域	选择需要同步的消息所属 Topic 的实例所在地域
源实例	选择需要同步的消息所属 Topic 所在实例
源Topic	输入需要同步的消息所属 Topic 名称
目标地域	选择消息将被同步到的 Topic 所属实例所在地域
目标实例	选择消息将被同步到的 Topic 所属实例
目标Topic	输入消息将被同步到的 Topic 名称
起始同步位点	选择从源 Topic 中的消息队列的哪个位置开始进行消息同步，即从这个位置之后进入队列的消息都会被同步到目标 Topic。目前仅支持最大位点，即从队列当前最新的一条消息开始同步。
过滤规则	输入过滤规则，按照该规则过滤后的消息才会被同步。您可选择按 Tag 过滤，直接输入 Tag 名即可

	，例如 “TagA” 。具体过滤方法可参考消息过滤。
描述	输入对该同步任务的具体描述或备注，以作区分和记录。

任务创建完成后，您可以在**全球消息路由**页面的任务列表中看到刚才创建的任务。

注意：全球消息路由任务创建成功后，该任务将处于**待运行**状态，您需手动启动该任务。

您可以在**全球消息路由**页面搜索、启停、查看和删除您创建的任务，也可为该任务编辑过滤规则。

搜索任务

在**全球消息路由**页面，在**源**和**目标**两栏按需输入源地域、源实例、源 Topic、目标地域、目标实例、目标 Topic 后，单击**搜索**按钮，即可查询到符合搜索条件的任务。

启停任务

在**全球消息路由**页面的任务列表中，找到需要启停的任务，在其**操作**列单击开始或停止图标，启动或停止该任务。

任务启动后，该任务的状态会切换至**运行中**。

任务停止后，该任务的状态会切换至**已停止**。

删除任务

前提条件：删除任务前需要先停止任务。

在**全球消息路由**页面的任务列表中，找到需要删除的任务，在其**操作**列单击**删除**，即可删除该任务。

任务被删除后将不会再在搜索结果中展示。

查看详情

在**全球消息路由**页面的任务列表中，找到需要查看的任务，在其**操作**列单击查看详情图标，即可查看该任务的任务详情和消息详情。同时，您还可以根据需求，编辑过滤规则。

任务详情页的关键字段的说明如下表所示。

参数	说明
任务状态	任务状态可以是 待运行 、 运行中 、 已停止 。
消息同步 TPS	每分钟统计一次消息同步的 TPS 值；TPS 即平均每秒传输的消息数量

消息延迟	当前同步到的位点和最新消息位点之间的时间差
消息堆积量	当前还没有来得及同步的消息数量
最新同步时间	最近一次消息同步发生的时间

编辑过滤规则

前提条件：编辑过滤规则，需先停止任务。

在**任务详情**区域的**过滤规则**一栏，单击编辑图标，即可对过滤规则进行修改。

查看订阅关系

在消息队列 RocketMQ 控制台，您可实时查看 Group ID 和 Topic 之间的订阅关系，即某个 Topic 被哪些 Group ID 订阅了，以及某个 Group ID 订阅了哪些 Topic。

什么是订阅关系

消息队列 RocketMQ 里的一个 Group ID 代表一个 Consumer 实例群组。对于大多数分布式应用来说，一个 Group ID 下通常会挂载多个 Consumer 实例。消息队列 RocketMQ 的订阅关系主要由 Topic+Tag 共同组成，即一个 Group ID 下所有的 Consumer 实例订阅的 Topic 以及这些 Topic 中的过滤规则 Tag。

同一个 Group ID 下所有的 Consumer 实例需保持订阅关系一致；否则，消息消费的逻辑就会混乱，甚至导致消息丢失。详情请参见订阅关系一致。

查看 Topic 被哪些 Group ID 订阅

前提条件

订阅该 Topic 的 Group ID 至少有一个处于在线状态。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如**华北1（杭州）**。

在左侧导航栏，单击**Topic 管理**。

在 **Topic 管理** 页面找到目标 Topic，然后在其 **操作列** 单击 **订阅关系**。

结果验证

在弹出的 **订阅关系** 对话框，除了看到该 Topic 的名称和最新一条消息的发送时间外，您还可以看到所有订阅该 Topic 的在线 Group ID 及其消费模式。

此外，您还可以单击任一 Group ID 的 **操作列** 的 **消费者状态**，查看该 Group ID 的消息消费详情。

查看 Group ID 订阅的 Topic

前提条件

需查询的 Group ID 处于在线状态。

操作步骤

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域，如 **华北1（杭州）**。

在左侧导航栏，单击 **Group 管理**。

在 **Group 管理** 页面找到目标 Group ID，在其 **操作列** 单击 **订阅关系**。

结果验证

在弹出的 **订阅关系** 对话框，除了看到该 Group ID 的名称、在线状态和消费模式外，您还可以看到该 Group ID 订阅的所有 Topic，以及这些 Topic 的消息过滤规则。

资源报表

资源报表是消息生产和消费数据的统计功能。通过该功能，您可查询在一段时间范围内发送至某 Topic 的消息总量或者 TPS（消息生产数据），也可查询在一个时间段内某 Topic 投递给某 Group ID 的消息总量或 TPS（消息消费数据）。

查看消息生产数据

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如**华北1（杭州）**。

在左侧导航栏，单击**资源报表**。

在**资源报表**页面，单击**消息生产**标签页。

在 **Topic** 一栏，选择您要查询的 Topic。

指定其他信息，然后单击**搜索**。

字段说明：

采集类型：分为总量和 TPS。总量提供每个周期内 Topic 接收的消息总量，TPS 提供每个周期内该 Topic 接收消息的平均 TPS。

采集周期：包括 1 分钟、10 分钟、30 分钟、1 小时。采集周期决定了数据采集的时间间隔，周期越短，采集点越密集，消息生产数据越详细。

时间范围：消息队列 RocketMQ 最多可以提供最近 3 天之内的消息的生产查询。

查询结果以图表形式显示。

查看消息消费数据

登录消息队列 RocketMQ 控制台。在顶部导航栏，选择地域，如**华北1（杭州）**。

在左侧导航栏，单击**资源报表**。

在**资源报表**页面，单击**消息消费**标签页。

在 **Group ID** 和 **Topic** 两栏，分别选择您要查询的 Group ID 和 Topic。

指定其他信息，然后单击**搜索**。

字段说明：

采集类型：分为总量和 TPS。总量提供每个周期内该 Topic 投递给该 Group ID 的消息总量，TPS 提供每个周期内该 Topic 投递给该 Group ID 消息的平均 TPS。

采集周期：包括 1 分钟、10 分钟、30 分钟、1 小时。采集周期决定了数据采集的时间间隔，周期越短，采集点越密集，消息消费数据越详细。

时间范围：消息队列 RocketMQ 最多可以提供最近 3 天之内的消息的消费查询。

查询结果以图表形式显示。

监控报警

您可使用消息队列 RocketMQ 提供的监控报警功能，监控某 Group ID 订阅的某 Topic 的消息消费状态并接收报警短信，帮助您实时掌握消息消费状态，以便及时处理消费异常。

新建监控项

前提条件

请确认实例所在地域（Region）支持监控报警功能，目前支持监控报警功能的地域如下所示：

- 公网
- 华东1（杭州）
- 华东2（上海）
- 华北1（青岛）
- 华北2（北京）
- 华北3（张家口）
- 华北5（呼和浩特）
- 华南1（深圳）
- 香港
- 华北2 阿里政务云1
- 新加坡
- 日本（东京）

请确保您需监控的 Group ID 已订阅相应 Topic。订阅消息的详细步骤，请参见以下文档：

- TCP Java 订阅消息
- TCP C/C++ 订阅消息
- TCP .NET 订阅消息
- HTTP 协议
- STOMP 协议

操作步骤

请按以下步骤创建监控项：

登录消息队列 RocketMQ 控制台消息队列 RocketMQ 控制台。在顶部导航栏，选择地域（Region），如华北1（杭州）。

在左侧导航栏，单击**监控报警**。

在**监控项管理**页面右上角，单击**新增监控项**。

在**新增监控项**对话框输入相关信息，单击**确认**。新建的监控项会出现在**监控项管理**页面。

字段说明：

Group ID：要监控的 Group ID。

Topic：要监控的该 Group ID 订阅的 Topic。

堆积量阈值：消费堆积的报警阈值。阈值的可填范围是 1 到 100,000,000。如果您选择的 Group ID 在消费对应的 Topic 时产生了消息堆积，并且堆积超过了报警阈值，消息队列 RocketMQ 就会给报警接收人发送短信通知。

消费延迟阈值：消费延迟的值是指该 Group ID 最近一次消费该 Topic 消息的时间和该 Topic 最近一次投递消息的时间之间的差值。可设置的消费延迟阈值的最小值是 1 分钟。

报警时间：时间精确到分钟，最大范围为 00:00-23:59。您只会在设置的接收时间范围内才会收到报警短信。

报警频率：每隔 5 分钟、15 分钟或 30 分钟。

报警接收人：包括接收人的昵称和手机号码。昵称长度限制在 100 个字以内。

管理监控项

在**监控项管理**页面，您可编辑、禁用、删除已经建好的监控项。对于已禁用的监控项，也可以在其**操作**列单击**启用**重新启用。