

消息队列 MQ

控制台使用指南

控制台使用指南

MQ 控制台提供几种消息查询方式用于问题排查。

适用场景

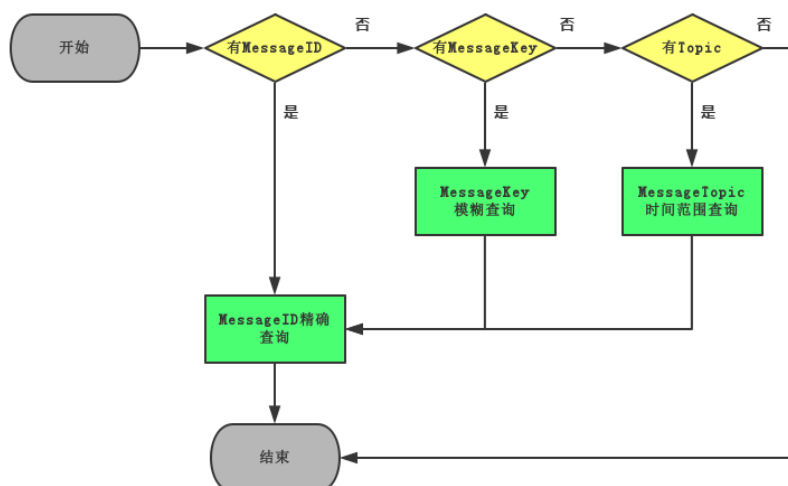
MQ 提供了三种消息查询的方式，分别是根据 Message ID，Message Key 以及 Topic 查询。

注意： HTTP 消息目前只支持 Topic 和 Message ID 两种查询方式。

三种方式的特点和对比如下。

查询方式	查询条件	查询类别	说明
Message ID 查询	Topic+Message ID	精确查询	根据 Message ID 可以精确定位任意一条消息，获取消息的属性
Message Key 查询	Topic+Message Key	模糊查询	根据 Message Key 可以匹配到包含指定 Key 的最近 64 条消息,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
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

建议查询过程：



根据 Message ID 查询

根据 Message ID 查询消息属于精确查询，用户输入 Message ID 信息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，建议发送方调用 Producer 发送消息时，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();
```

查询结果说明

查询到消息后，您可以在消息详情页面看到每条消息的基本属性，消息体下载链接以及消息的消费状态，如下图所示：

The screenshot shows the 'Message ID 查询' (Message ID Query) interface in the Message Queue console. The left sidebar has 'Message ID 查询' highlighted. The main area displays the details of a specific message with the ID '1E094C984A7E55C4D59486206980000F'. Below the details, there is a 'Message body' section with a '下载' (Download) link. At the bottom, a table shows the consumption status of the message by different consumers.

Consumer	投递状态	操作
CID_MsgTraceDemo1	已订阅，但是订阅组当前不在线	消费验证
CID_MsgTraceDemo3	已订阅，并且消息已被消费	消费验证
CID_MsgTraceDemo2	已订阅，但是订阅组当前不在线	消费验证

其中，投递状态是 MQ 根据各个 Consumer 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，具体请参见[查询消息轨迹](#)。

投递状态表

投递状态	可能的原因
已订阅，并且消息已被消费	该订阅组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，可以在控制台中的 订阅管理 > 消费者状态查询 菜单查询订阅方的订阅关系
已订阅，但消息未被消费	该订阅组订阅了该消息，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，但是订阅组当前不在线	该订阅组订阅了该消息，但是不在线，请检查消费者端应用不在线的原因
未知异常	出现未收录异常，可以进工单系统提工单解决

消费验证

MQ 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点击消费验证后并不会改变。
- 采用 HTTP 协议接入时，由于消息发送是短连接方式，因此不支持 HTTP 消息的消费验证功能。

根据 Message Key 查询

Message Key 查询消息的原理是，MQ 根据用户设置的 Message Key 信息建立消息的索引信息，当用户输入 Key 进行查询时，根据该索引即可匹配相关的消息返回。

注意：

- Message Key 查询条件是用户设置的 Message Key 属性。
- Message Key 查询仅仅返回符合条件的最近的 64 条消息，因此建议业务方设置 Key 时尽可能唯一，并具有业务区分度。

设置 Message Key 的方法：

```
Message msg = new Message("Topic", "*", "Hello MQ".getBytes());  
/**  
 * 对每条消息设置其检索的Key，该Key值设置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MQ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());
```

查询样例如下图所示：

消息队列 MQ

公网测试 华北1 华南1 华东1 华北2 华东2

发布订阅管理

消息查询

Topic查询

Message Key查询

Message ID查询

资源探索

消息轨迹(公测)

监控报警

帮助支持

Message Key查询 Key是指发送消息时，设置到消息对象中的Key字段。限制：一个Key查询出的结果集合不超过64条

MsgTraceDemo TestKey1467003870187 搜索

Message ID	Tag	Key	Storetime	操作
1E094C984A7E55C4D594862068300000	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068980001	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068B10003	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068CA0005	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068E40007	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068FD0009	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情
1E094C984A7E55C4D59486206915000B	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情

根据 Topic 查询

根据 Topic 查询一般用在前两种查询条件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：

- 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。
- Topic 查询数据量大，采用分页展示。

查询样例如下图所示：

发布订阅管理

消息查询

Topic查询

Message Key查询

Message ID查询

资源探索

消息轨迹(公测)

监控报警

帮助支持

Topic查询

MsgTraceDemo 2016-06-27 12:08 至 2016-06-27 13:38 搜索

Message ID	Tag	Key	Storetime	操作
1E094C984A7E55C4D594862068980001	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068B10003	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068CA0005	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068E40007	TestTag	TestKey1467003870187	2016-06-27 13:04:30	消息详情
1E094C984A7E55C4D594862068FD0009	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情
1E094C984A7E55C4D59486206915000B	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情
1E094C984A7E55C4D5948620696A000E	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情
1E094C984A7E55C4D59486206980000F	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情
1E094C984A7E55C4D594862069980011	TestTag	TestKey1467003870187	2016-06-27 13:04:31	消息详情

消息轨迹

本文档介绍 MQ 消息轨迹的基本原理、使用场景、以及使用案例。

目前 MQ 支持 TCP 和 HTTP 协议下的消息轨迹查询。

基本原理

定义：消息轨迹指的是一条消息从生产方发出到消费方消费处理，整个过程中的各个相关节点的时间地点等数据汇聚而成的完整链路信息。

原理：MQ 系统中，一条消息的完整链路包含生产方、服务方、消费方三个角色，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，从而为生产环境中的问题排查提供强有力的数据支持。

消息轨迹的数据包含：

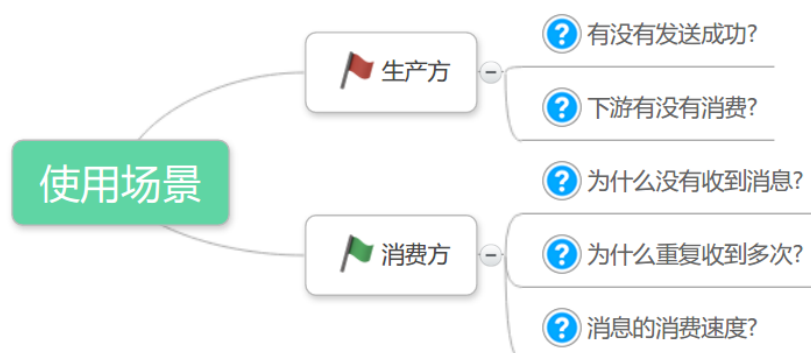
生产方信息	消费方信息	服务方信息
生产客户端信息	消费客户端信息	消息存储位置信息
发送时间	投递时间，投递轮次	消息存储时间
发送成功与否	消费成功与否	消息本身的属性
发送耗时	消费耗时	-

消息轨迹查询规则：

消息类型	可以查询的时间	查询说明
普通消息	消息发送后	消息发送之后有发送轨迹，没消费前提示没有消费。消费后会展示拉取和消费。
顺序消息	消息发送后	消息发送之后有发送轨迹，没消费前提示没有消费。消费后会展示拉取和消费。
定时/延时消息	当前时间到达消息指定消费的时间	当前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	消息发送后	消息发送之后有发送轨迹。事务未提交之前，轨迹可以查询到，但是消息查询不到。

使用场景

在生产环境的消息收发不符合预期时可以使用消息轨迹工具排查问题。通过消息的属性（Message ID、Message Key、Topic）搜索相关的消息轨迹，找到消息的实际收发状态，帮助诊断问题。



示例

假设业务方根据业务日志里的信息判断某条消息一直没有收到，则可以参考以下步骤，利用消息轨迹来排查 MQ 问题。

收集怀疑的消息的信息，Message ID，Message Key，Topic 以及大概的发送时间范围。

进入 MQ 控制台，根据已有的信息新建查询任务，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，则可以去订阅管理页面查询，确认是否有堆积导致消息尚未消费。

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，登录查看相关日志。

消息轨迹的使用对于业务方不会增加额外的接入成本，仅仅需要确保客户端 SDK 版本支持该特性。正常收发消息后以消息的相关属性在 MQ 控制台上查询即可。

1. 接入条件

消息轨迹功能目前支持 Java 客户端（1.2.2版本及以上），C++，.NET 以及 HTTP 客户端。详细的客户端信息，请参考控制台给出的升级提示，获取最新的 SDK。

2. 新建轨迹查询

在 MQ 控制台左侧菜单栏选择**消息轨迹**（现处于公测阶段），选择所在的区域，并单击右上角**新建查询**按钮。

新建查询

根据MsgID查询 根据MsgKey查询 根据Topic查询

Topic :
topic是所要查询的消息的主题

MsgId :
msgid是由发送方Send方法返回的32位Hex字符串

时间范围: 2016-06-12 08 : 00 至 2016-06-15 22 : 00

确定 取消

消息轨迹查询功能支持三种查询方式，请按照对应方式输入查询条件，新建查询。

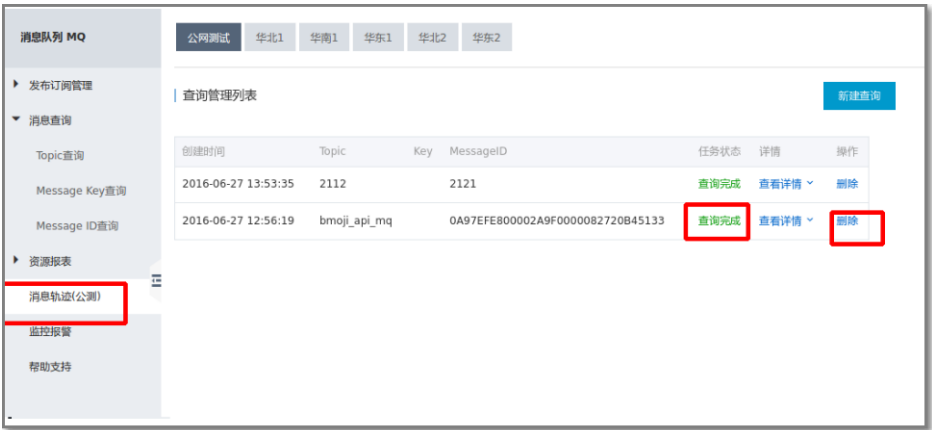
- 根据 Message ID 查询：需要输入消息的唯一 Message ID，Topic 名称以及消息的大致发送时间。
- 根据 Message Key 查询：需要输入消息的 Message Key 和 Topic 以及大致发送时间，适用于没有记录 MessageID，但记录了 Message Key 的场景。
- 根据 Topic 查询：仅仅输入 Topic 和时间段，批量查询，适用于没有上述 Message ID 和 Message Key，而且消息量比较小的场景。

注意：

1. 查询时，尽可能设置最为精确的时间区间，以便缩小查询范围，提高速度。
2. 根据 MsgID 查询属于精确查询，速度快，精确匹配，推荐用户使用。
3. 根据 MsgKey 查询属于模糊查询，仅适用于业务方没有记录 MessageID 但是设置了 MessageKey，同时 MessageKey 具有区分度的情况，MessageKey 查询最多查询 1000 条轨迹。
4. 根据 Topic 分段查询属于范围查询，不推荐使用，因为时间范围内消息很多，不具备区分度。

3. 查询任务管理

新建查询后，会生成一个查询任务，MQ 后台会异步执行，并将任务状态反馈到管理页面，查询结束时，任务状态显示查询完成，否则显示查询中。

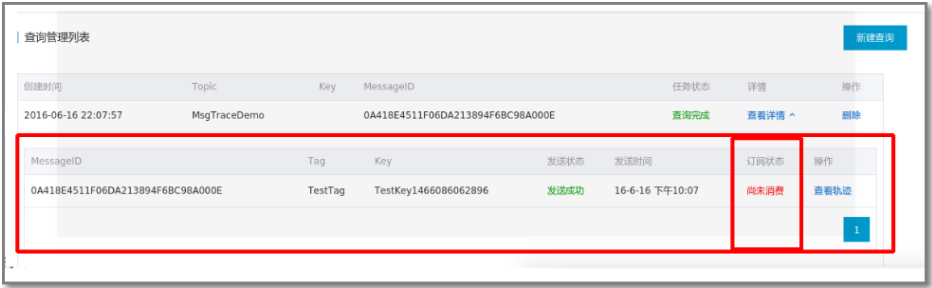


根据任务的状态可以选择查看轨迹，或者删除查询任务。

4. 查看轨迹

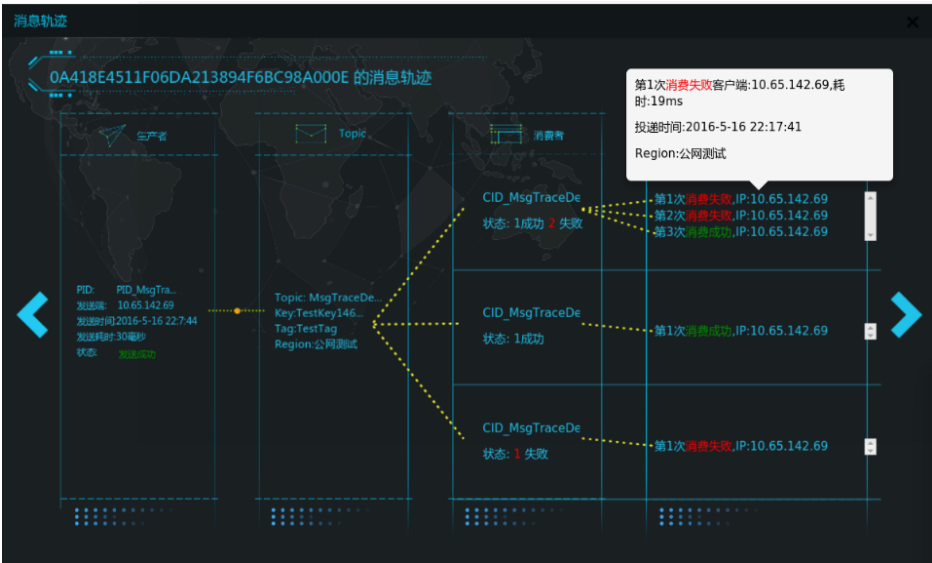
完成查询后，单击右侧操作选项里的**查看详情**按钮查看轨迹。如果发现没有结果，请参考弹窗链接，排查原因。

如果查询到轨迹信息，可以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统计，如下图所示：



5. 查看轨迹链路图

单击**查看轨迹**按钮即可查看完整的链路图，如图所示：



消息链路图包含4个部分：

1. 生产者信息
2. Topic 信息
3. 消费者信息
4. 详情信息

各个字段区域均可以通过鼠标悬停的方式获取详细信息。对于 MsgKey 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。

消息轨迹名词解释

消息轨迹查询页面中涉及到的名词概念列表如下。

相关概念	含义（TCP场景）	含义(HTTP场景)
发送时间	记录消息从发送端发送时的客户端时间戳	同TCP
发送耗时	记录发送端调用 send 方法发送消息的毫秒耗时	同TCP
Region	记录消息存储的 Region 信息，或者消费方机器所在的 Region 信息	同TCP
消费耗时	记录消息推送到客户端之后执行 consumeMessage 方法的耗时	从HTTP客户端执行GET拉取消息到执行DELETE操作的时间间隔
投递时间	记录客户端执行 consumeMessage 方法开始消费消息时的时间戳	HTTP客户端执行GET拉取消息的时间戳

消息轨迹状态说明

消息轨迹查询页面中涉及到的状态列表如下。

相关概念	含义（TCP场景）	含义(HTTP场景)
发送成功	消息发送成功，服务端已经存储成功	同TCP
发送失败	消息发送失败，服务端没有存储消息，需要重试	同TCP
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	同TCP
事务未提交	该消息是事务消息，且尚未提交状态	无
事务回滚	该消息是事务消息，并且已经回滚	无
全部成功	该消息所有投递都已成功消费	同TCP

部分成功	该消息投递中存在消费失败并重试成功的情况	同TCP
尚未消费	该消息尚未投递给任何消费方	HTTP客户端尚未拉取消息
消费未返回	消费消息的方法尚未返回，或者被中断，导致本次消费结果未传回服务端	HTTP客户端执行GET拉取消息后，没有调用DELETE方法返回消费结果
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	无

如果对消息轨迹的查询结果有疑问，也可以参考相关的 FAQ。

资源报表是 MQ 提供的消息发送和消费的统计数据查询功能，用户通过资源报表可以对某个 Topic 在一段时间范围内的发送总量或者 TPS 进行查询，也可以对一个 Consumer ID 消费某个 Topic 的总量或者 TPS 进行查询。

注意：MQ 资源报表暂时不支持 HTTP 消费数据的统计查询。

消息消费查询

消息消费查询用于对一个 Consumer ID 消费某个 Topic 的总量或者 TPS 进行查询。

具体操作步骤如下：

在 MQ 控制台左侧菜单栏选择**资源报表**>**消息消费**。

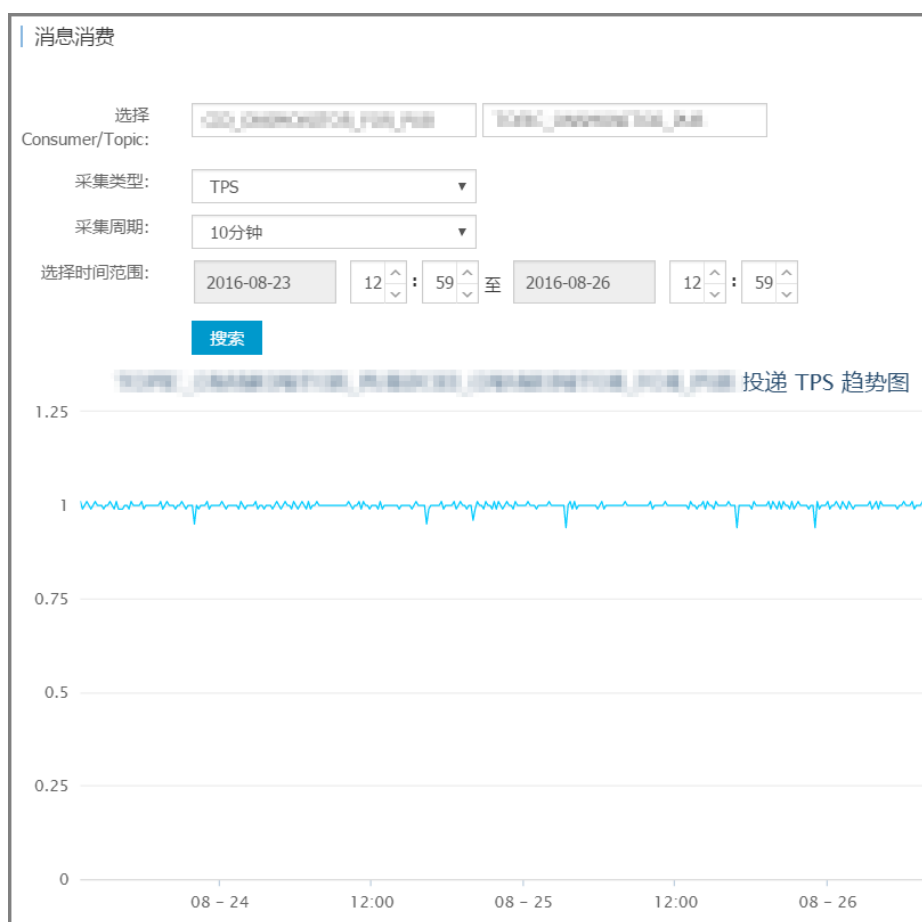
选择您要查询的 Consumer ID 所在的域。

在消息消费页面输入相应信息，单击搜索。

选项说明：

- **采集类型**：分为 TPS 和总量，总量提供该周期内消费该 Topic 数据总量，TPS 提供消费该 Topic 的平均 TPS。
- **采集周期**：包括 1 分钟、10 分钟后、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**：MQ 最多可以提供最近三天之内的消息消费查询。

查询结果以图表的形式返回：



消息生产查询

消息生产查询用于对某个 Topic 在一个时间段内的消息发送总量或者 TPS 进行查询。

具体操作步骤如下：

在 MQ 控制台左侧菜单栏选择**资源报表**>**消息生产**。

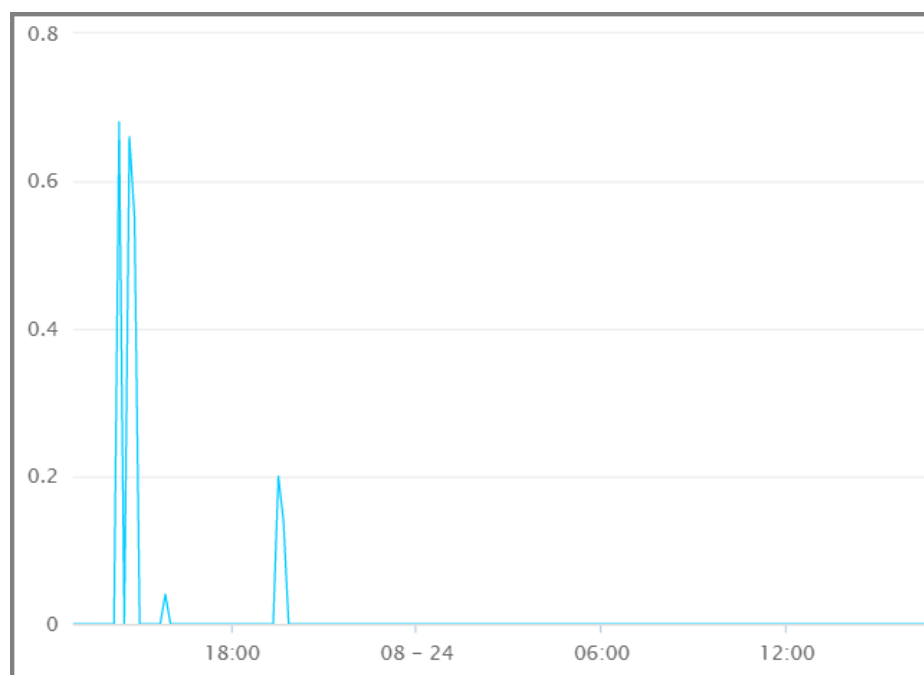
选择您要查询的 Topic 所在的域。

在消息消费页面输入相应信息，单击搜索。

选项说明：

- **采集类型**：分为 TPS 和总量，总量提供该周期内 Topic 的发送总量，TPS 提供发送平均 TPS。
- **采集周期**：包括 1 分钟、10 分钟后、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**：MQ 最多可以提供最近三天之内的消息消费查询。

查询结果以图表的形式返回:



MQ 提供了对用户所订阅 Topic 的消费状态进行监控并且发送短信报警的功能，帮助您实时掌握消息消费状态并及时处理消费异常等情况。

使用监控报警功能，首先需要在 MQ 上通过 Consumer ID 对 Topic 进行订阅，然后对相关监控项进行设置。当消费者在线，并且消息堆积数或者延迟时间超过阈值之后，MQ 会对报警接收人发送短信进行报警。

新建监控项

用户可以通过新建监控功能对一个 Consumer ID 的消费状态进行监控：

登录 MQ 控制台，在左侧菜单栏选择**监控报警**。

选择您要的 Topic 所在的域。

在监控项管理页面，单击右上角**新增监控项**。

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

选项说明：

- **Consumer ID**：选择要监控的 Consumer ID。
- **Topic**：该 Consumer ID 对应的 Topic。
- **报警阈值**：是指消费堆积的阈值，堆积报警阈值的范围是 1 到 100,000,000。如果您选

择的 Consumer 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，MQ 就会给报警接收人发送短信通知。

- **消费延迟阈值**：是指该 Consumer ID 最后一次消费该 Topic 消息的时间和该 Topic 最近发送时间之间的差值，消费延迟阈值的最小值是 1 分钟。
- **报警时间**：范围精确到分钟，最大范围为 00:00-23:59，用户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**：根据需要选择 5 分钟、15 分钟或者 30 分钟。
- **报警接收人**：填写接收人的昵称和手机号码，昵称范围为 32 个字。

监控项管理

在监控项管理页面，您可以对已经建好的监控项进行编辑、禁用、删除等操作。对于禁用的监控项，也可以在监控项管理页面单击启用重新启用。