

阿里云 ONS 用户指南

针对 ONS v1.1.4

©Alibaba 消息中间件项目组

2014/11/19

文档变更历史

序号	主要更改内容	更改人	更改时间
1	增加成功案例章节	誓嘉 vintage.wang@gmail.com	2014/7/17
2	更新 key 文件下载地址	誓嘉 vintage.wang@gmail.com	2014/8/8
3	为公测增加接入流程	誓嘉 vintage.wang@gmail.com	2014/9/10
4	为公测进一步完善文档 ,增加 Console 使用说明	誓嘉 vintage.wang@gmail.com	2014/9/25
5	为最终公测版本更新文档 v1.1.2	誓嘉 vintage.wang@gmail.com	2014/10/23
6	开通公网收发消息功能	誓嘉 vintage.wang@gmail.com	2014/11/19

目录

1	前言	1
2	产品背景	1
3	专业术语	1
4	ONS 成功案例	2
5	ONS 功能特性	3
6	ONS 典型业务场景	3
6.1	电商订单系统利用 ONS 来解耦	3
7	ONS 代码样例	3
7.1	发送消息	4
7.2	集群方式订阅消息	5
7.3	广播方式订阅消息（广播消息在下个版本会有大的优化，暂时不推荐您使用）	6
8	ONS 消费失败重试特性	6
8.1	集群消费方式	6
8.2	广播消费方式	7
9	ONS 服务功能开通	8
10	ONS Console 能做什么？	11
10.1	如何发送消息	11
10.2	如何订阅消息	12
10.3	如何删除消息	13
10.4	如何查询消息	14
10.4.1	按照 Message Topic 查询消息	14
10.4.2	按照 Message Key 查询消息	14
10.4.3	按照 Message Id 查询消息	14

10.5 消息发送了，但是没有收到怎么办？	15
附录 A Q&A.....	17
附录 B 参考文档、规范.....	20

1 前言

本文档旨在描述阿里云 ONS 消息系统的基本概念，以及如何使用。

2 产品背景

ONS 是从阿里 MetaQ (RocketMQ) 演化而来，MetaQ 在阿里各个业务线得到了广泛的应用，业务场景如下：

- Mysql binlog 同步
- 订单类应用
- 流计算
- IM 等实时消息领域
- Cache 同步
- 双十一、双十二的削峰填谷

每天会有海量消息产生，历经多次天猫双十一考验，稳定成熟可靠的消息产品。

同时，MetaQ 也是阿里的开源项目，开源版本名称为 RocketMQ

<https://github.com/alibaba/rocketmq>

ONS 的主要特点

- 1、无单点、无瓶颈、可自由扩展
- 2、经历过三次双十一海量消息的考验，稳定性有保障。

3 专业术语

■ Producer

消息生产者，负责产生消息，一般由业务系统负责产生消息。

■ Consumer

消息消费者，负责消费消息，一般是后台系统负责异步消费。

■ **Producer ID**

一类 Producer 的集合名称，这类 Producer 通常发送一类消息，且发送逻辑一致。

■ **Consumer ID**

一类 Consumer 的集合名称，这类 Consumer 通常消费一类消息，且消费逻辑一致。

■ **广播消费**

一条消息被多个 Consumer 消费，即使这些 Consumer 属于同一个 Consumer ID，消息也会被 Consumer ID 中的每个 Consumer 都消费一次，广播消费中的 Consumer ID 概念可以认为在消息划分方面无意义。

在 CORBA Notification 规范中，消费方式都属于广播消费。

在 JMS 规范中，相当于 JMS publish/subscribe model

■ **集群消费**

一个 Consumer ID 中的 Consumer 实例平均分摊消费消息。例如某个 Topic 有 9 条消息，其中一个 Consumer ID 有 3 个实例（可能是 3 个进程，或者 3 台机器），那么每个实例只消费其中的 3 条消息。

在 CORBA Notification 规范中，无此消费方式。

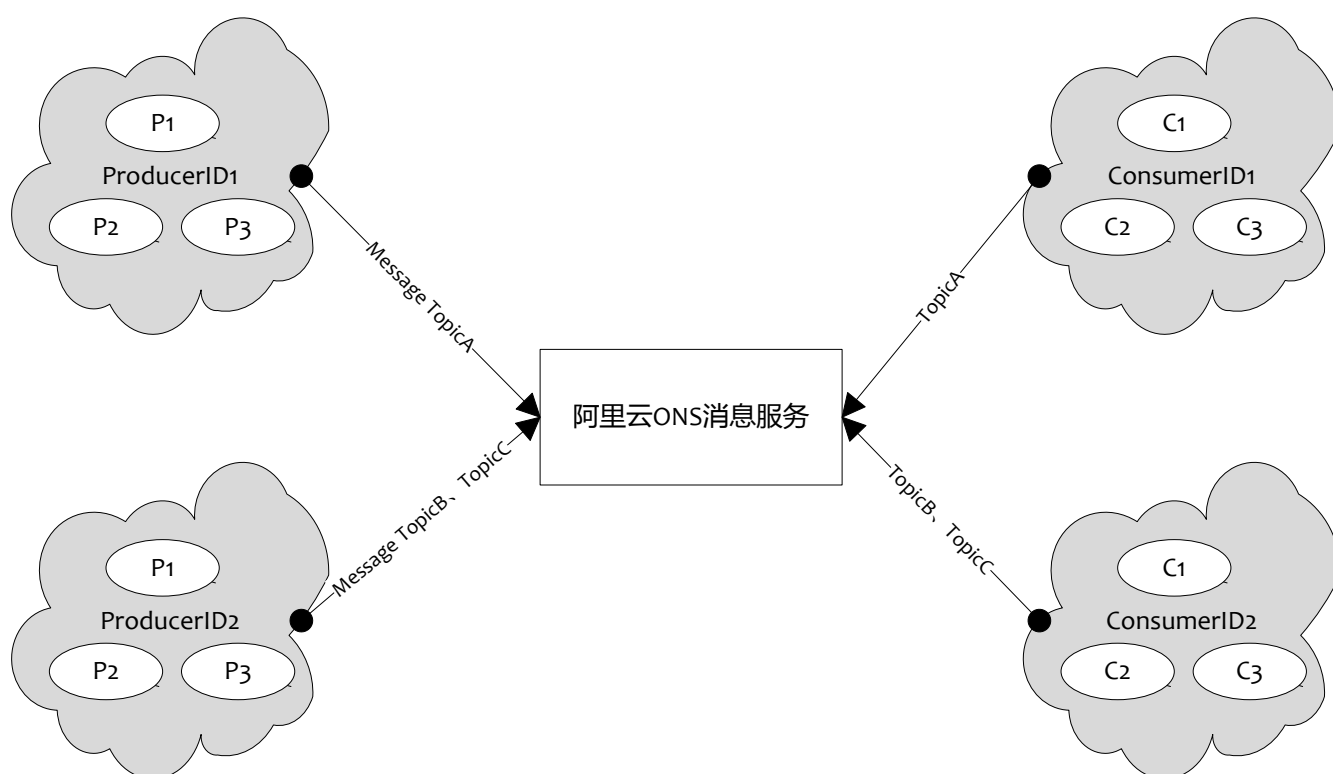
在 JMS 规范中，JMS point-to-point model 与之类似，但是 ONS 的集群消费功能大等于 PTP 模型。因为 ONS 单个 Consumer ID 内的消费者类似于 PTP，但是一个 Topic 可以被多个 Consumer ID 消费。

4 ONS 成功案例

下面所列用户已经在阿里云上正式上线使用

- 杭州安存科技
- 北京华甫达
- 上海网聚宝

5 ONS 功能特性



图表 5-1 阿里云 ONS 功能特性图

- 一、如图所示，ProducerID1 的 producer 实例有三个，可能是部署在三个机器上的三个进程，也可能是一台机器上的三个进程。每个实例都会发送 TopicA 的消息。同理，ProducerID2 与之类似。
- 二、ConsumerID1 有三个实例，如果是集群消费方式，那么每个实例消费 TopicA 的 $\frac{1}{3}$ 的消息，如果是广播消费方式，那么每个实例消费全量的消息。另外，TopicA 也可以被另外的 ConsumerId 再次消费。

6 ONS 典型业务场景

6.1 电商订单系统利用 ONS 来解耦

// TODO

7 ONS 代码样例

要运行本节描述的 Java 代码，前置条件如下：

一、通过下面两种方式可以引入依赖(任选一种)

1、Maven 方式引入依赖

```
<dependency>
  <groupId>com.aliyun.openservices</groupId>
  <artifactId>ons-client</artifactId>
  <version>1.1.4</version>
</dependency>
```

2、下载依赖 Jar 包

<http://onsall.oss-cn-hangzhou.aliyuncs.com/aliyun-ons-client-java.tar.gz>

二、代码里涉及到的 Topic , ProducerId , ConsumerId , 需要到 ONS Console 上创建。Message Tag 可以完全由应用自定义。

7.1 发送消息

```
public class ProducerTest {

    public static void main(String[] args) {
        Properties properties = new Properties();
        properties.put(PropertyKeyConst.ProducerId, "PID_001");
        properties.put(PropertyKeyConst.AccessKey,
"F0000043D88CB7EF4");
        properties.put(PropertyKeyConst.SecretKey,
"F0000043D900F191F");
        Producer producer = ONSFactory.createProducer(properties);

        // 在发送消息前，必须调用start方法来启动Producer，只需调用一次即可。
        producer.start();
        Message msg = new Message( //
            // Message Topic
            "TopicTestONS",
            // Message Tag,

            // 可理解为Gmail中的标签，对消息进行再归类，方便Consumer指定过滤条件
            // 在ONS服务器过滤
            "TagA",
            // Message Body

            // 任何二进制形式的数据，ONS不做任何干预，需要Producer与Consumer协商
```


好一致的序列化和反序列化方式

```
        "Hello ONS".getBytes());

// 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ONS Console查询消息并补发。

// 注意：不设置也不会影响消息正常收发
msg.setKey("ORDERID_100");

// 发送消息，只要不抛异常就是成功

SendResult sendResult = producer.send(msg);
System.out.println(sendResult);

// 在应用退出前，销毁Producer对象<br>

// 注意：如果不销毁也没有问题
producer.shutdown();
    }
}
```

7.2 集群方式订阅消息

```
public class ConsumerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        properties.put(PropertyKeyConst.ConsumerId, "CID_001");
        properties.put(PropertyKeyConst.AccessKey, "3D88CB7EF4");
        properties.put(PropertyKeyConst.SecretKey, "3D900F191F");
        Consumer consumer = ONSFactory.createConsumer(properties);
        consumer.subscribe("TopicTestONS", "*", new MessageListener() {

            public Action consume(Message message, ConsumeContext context)
            {
                System.out.println("Receive: " + message);
                return Action.CommitMessage;
            }
        });

        consumer.start();

        System.out.println("Consumer Started");
    }
}
```

```
}  
}
```

7.3 广播方式订阅消息（广播消息在下一个版本会有大的优化，暂时不推荐您使用）

```
public class ConsumerTest {  
    public static void main(String[] args) {  
        Properties properties = new Properties();  
        properties.put(PropertyKeyConst.ConsumerId, "CID_001");  
        properties.put(PropertyKeyConst.AccessKey, "3D88CB7EF4");  
        properties.put(PropertyKeyConst.SecretKey, "3D900F191F");  
        properties.put(PropertyKeyConst.MessageModel,  
PropertyKeyConst.Broadcasting);  
        Consumer consumer = ONSFactory.createConsumer(properties);  
        consumer.subscribe("TopicTestONS", "*", new MessageListener() {  
  
            public Action consume(Message message, ConsumeContext context)  
            {  
                System.out.println("Receive: " + message);  
                return Action.CommitMessage;  
            }  
        });  
  
        consumer.start();  
  
        System.out.println("Consumer Started");  
    }  
}
```

8 ONS 消费失败重试特性

8.1 集群消费方式

```
/**  
 * 消息监听器，Consumer注册消息监听器来订阅消息  
 */  
public interface MessageListener {  
    /**  
     * 消费消息接口，由应用来实现<br>  
     */  
}
```

```

* @param message
*         消息
* @param context
*         消费上下文

* @return 消费结果，如果应用抛出异常或者返回Null等价于返回
Action.ReconsumeLater
*/
public Action consume(final Message message, final ConsumeContext
context);
}

```

消费业务逻辑代码如果返回 Action.ReconsumeLater，或者 NULL，或者抛出异常，消息都会走重试流程，至多重试 16 次，如果重试 16 次后，仍然失败，则消息丢弃。

每次重试的间隔时间如下：

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

如何访问消息是第几次重试？

message.getReconsumeTimes()

8.2 广播消费方式

广播方式不提供失败重试特性，即消费失败后，失败消息直接丢弃，继续消费新的消息。

9 ONS 服务功能开通

ONS Console 提供了创建 ProducerId、ConsumerId、Topic 等功能，同时也可以在上面自助排查问题，例如某个消息发送了但是没有收到，或者收消息比较慢，查询消费进度等，都可以在 ONS Console 平台完成。

目前 ONS Console 处于公测阶段，您需要申请公测资格

1. 访问下面 URL

<http://ons.console.aliyun.com/>

2. 如果未开通 ONS 消息服务，会提示您开通 ONS 开放消息服务

开放消息服务ONS开通

开通需申请公测资格

申请公测资格

产品名称：开放消息服务ONS

开通方式：公测阶段不收费

注意事项：ONS只允许用户在ECS上使用

☒我已阅读并同意 开放消息服务ONS开通协议

立即开通

邀请码申请已升级为公测资源申请，获取公测资源后无需再手工激活，系统会自动判断您是否已拥有公测资源。

产品名称： 开放消息服务ONS

* 联系人姓名：

* 手机号码：

* 邮箱：

* IM选择：☒ 旺旺

* 旺旺：

* 用户类型：☒ 企业 ☐ 个人

* 企业名称：

* 申请原因：

请您注明是从哪个渠道了解到ONS产品，例如微博、QQ、Github等



申请提交

10 ONS Console 能做什么？

10.1 如何发送消息

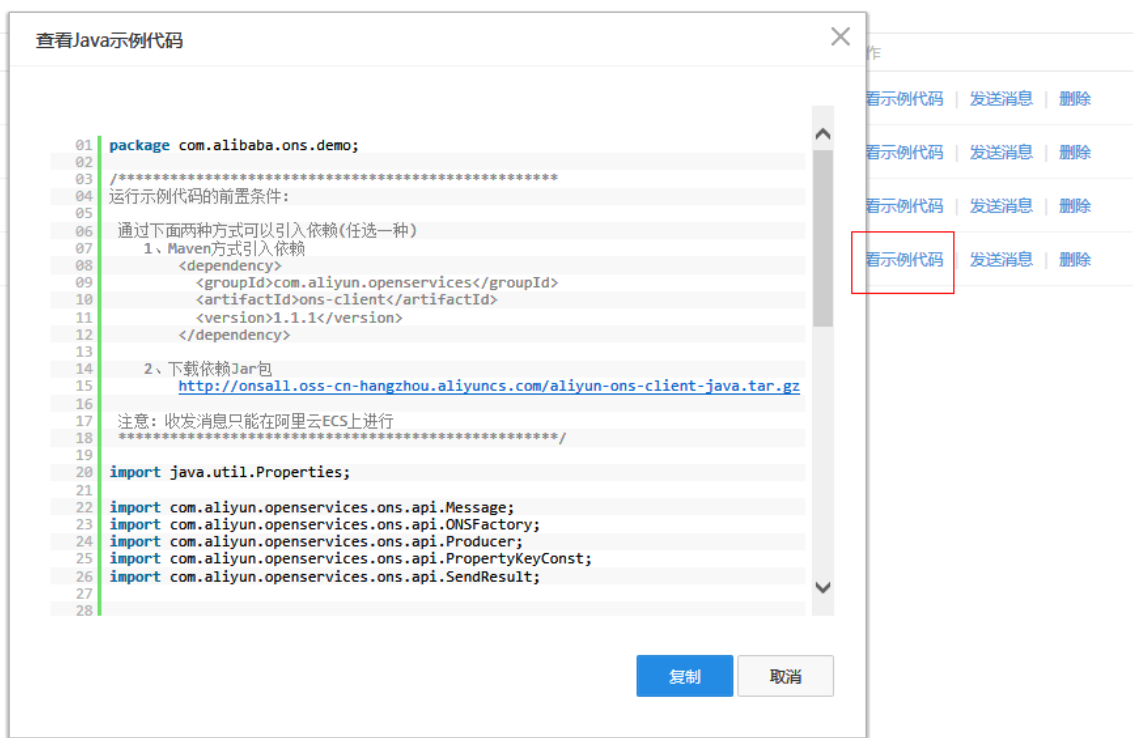
1. 发送消息前，在 ONS Console 先创建 Topic

The screenshot shows the ONS Console interface. At the top right, there is a blue button labeled '发布Topic' (Publish Topic) which is highlighted with a red rectangular box. Below this, there is a table with columns: Topic, 地域 (Region), 创建时间 (Creation Time), 状态 (Status), and 操作 (Operations). The table contains one row with Topic 'shijia7', Region '杭州', Creation Time '2014-09-23 13:29:48', and Status '可用' (Available). The '操作' column for this row contains links: '查看示例代码' (View Example Code), '发送消息' (Send Message), and '删除' (Delete). Overlaid on the table is a modal dialog box titled '发布Topic'. The dialog box has a close button (X) in the top right corner. It contains the following fields: '*Topic:' with a text input 'Topic100' and a clear button (X); '地域:' with a dropdown menu showing '杭州'; 'Producer ID:' with a text input field; and '备注:' (Remarks) with a text area. At the bottom of the dialog box are two buttons: '确定' (Confirm) and '取消' (Cancel).

2. 可以通过 ONS Console 页面直接发送消息

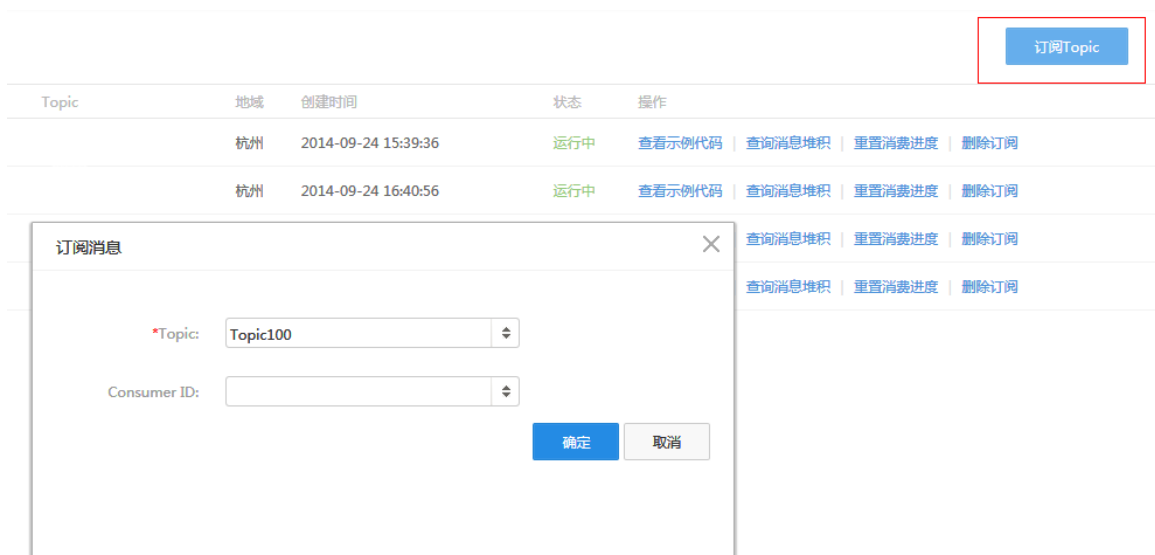
The screenshot shows the ONS Console interface. Overlaid on the table from the previous image is a modal dialog box titled '发送消息' (Send Message). The dialog box has a close button (X) in the top right corner. It contains the following fields: 'Producer ID:' with a text input 'PID25771358565-109'; 'Topic:' with a text input 'Topic100'; 'Tag:' with a text input field; 'Key:' with a text input field containing the text '限制在32字之内'; and '*Message Body:' with a text area. At the bottom of the dialog box are two buttons: '确定' (Confirm) and '取消' (Cancel). In the background, the table from the previous image is visible, and the '发送消息' link in the '操作' column is highlighted with a red rectangular box.

3. 也可以通过 SDK(目前仅支持 Java 语言)发送消息，请点击“查看示例代码”，请使用代码示例中的 Maven 依赖或者下载 Java Jar 包的方式，来运行代码。

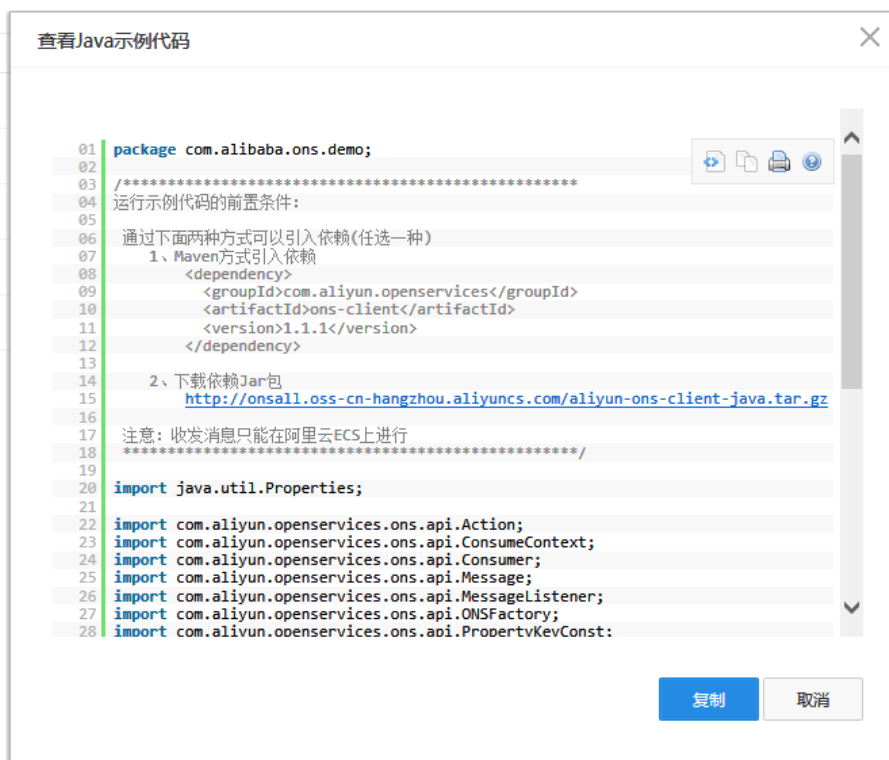


10.2 如何订阅消息

1. 请到 ONS Console 上创建 Consumer Id , 并指定要订阅的 Topic



2. 点击 “查看示例代码”, 按照代码说明来订阅消息



10.3 如何删除消息

有两种方式可以删除消息

1. 删除 Topic

整个 Topic 的消息彻底被删除，所有订阅这个 Topic 的订阅者不会再收到这个 Topic 的消息。

2. 重置消费进度，针对 Consumer，重置时间到当前时间，则会跳过历史消息，只消费当前时间之后的消息。

也可以指定更早的消息来重新消费历史消息。



10.4 如何查询消息

10.4.1按照 Message Topic 查询消息

通过 Topic 可以遍历查询这个 Topic 下的所有消息，如果 Topic 消息比较多，会查询出非常多的记录，此功能仅供您日常调试使用。

Topic查询

Message Key查询

Message ID查询

Topic查询

Topic100

搜索

Message ID	Tag	Key	Storetime	操作
0A98455000001FA40000000125AD01EF	Tag2	Key4552	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125AD0087	Tag0	Key4550	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFF7F	Tag3	Key4548	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFE47	Tag1	Key4546	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFD0F	Tag4	Key4544	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125AD0153	Tag1	Key4551	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125AD001B	Tag4	Key4549	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFEE3	Tag2	Key4547	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFDA8	Tag0	Key4545	2014-09-25 16:48:35	查看详情
0A98455000001FA40000000125ACFC73	Tag3	Key4543	2014-09-25 16:48:35	查看详情

<<

>

1

2

3

4

5

6

7

8

9

10

10.4.2 按照 Message Key 查询消息

发送消息时，如果指定了 Message Key 字段，那么就可以利用 “Message Key 查询” 机制，唯一定位到这条消息，请尽可能每条消息指定的 Key 唯一，如果多个消息都指定相同的 Key，那么 Key 查询会查询出多条记录。

限制：Key 查询至多返回 32 条记录。

Topic查询

Message Key查询

Message ID查询

Message Key查询

Topic100

Key101652

搜索

Message ID	Tag	Key	Storetime	操作
0A98455000001FA40000000126A8374B	Tag2	Key101652	2014-09-25 16:52:35	查看详情

10.4.3 按照 Message Id 查询消息

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

Producer 发送消息时，会返回 Message Id，根据 Message Id 可以在 ONS Console 上唯一查询到这条消息。

Message Body 是二进制形式，所以需要单独下载下来，使用文本编辑器查看。

| Message ID查询

0A98455000001FA40000000126A8374B

搜索

Message ID: 0A98455000001FA40000000126A8374B

Topic: Topic100

Tag: Tag2

Key: Key101652

Userproperties: {TAGS=Tag2, __KEY=Key101652, KEYS=Key101652, __TAG=Tag2}

Storetime: 2014-09-25 16:52:35

Message body: [下载](#)

消息状态:

Consumer

投递状态 ?

操作

10.5 消息发送了，但是没有收到怎么办？

1. 通过 Message Key 或者 Message Id 查询到这条消息，点击“查看详情”，会显示这条消息被哪些消费者消费了。如果通过 Message Key 未查询到消息，说明消息没有发送到 ONS 服务器，可能原因是消息发送失败，或者没有发送。消息发送失败时，send 接口会抛出异常。

MessageID 查询

0A98455000001FA4000000012C595396

搜索

Message ID: 0A98455000001FA4000000012C595396

Topic: Topic100

Tag: Tag4

Key: Key656759

Userproperties: {TAGS=Tag4, __KEY=Key656759, KEYS=Key656759, __TAG=Tag4}

Storetime: 2014-09-25 17:15:03

Message body: [下载](#)

消息状态:

Consumer	投递状态 <small>?</small>	操作
CID25771358565-108	已投递	消费验证

2. 如果您对系统判断出的投递状态有疑问，可以选择“消费验证”功能，把这条消息再投递一次，并同步向控制台页面显示消费结果。

Message ID: 0A98455000001FA4000000012C595396

Topic: Topic100

Consumer ID: CID25771358565-108

●10.171.247.2@3924_53789536_1439626187

这个功能只作为验证消费者应用是否能接受到消息所用，点击【确定】按钮后消息将会重新发送到指定的IP上，可能会引起消息重复。

确定

取消

Q：如果没有设置 Message Key，也没有记录 Message Id，怎么排查这种消息丢失问题？

A：打个比喻，“一个寻人启示写到：某人丢失，有遇到者请速联系家人”，茫茫人海，如何找到某人？

所以请您发送消息时，务必设置 Message Key，或者记录 Message Id，以方便应用认为消息丢失可以查询并补发。

附录 A Q&A

1. ONS 是否支持消息有序？

暂时不支持

2. ONS 消息在服务器保存多长时间？

保存 3 天，超过 3 天未消费的消息会被删除掉。

如果您消费落后比较厉害，会有报警短信通知，这时请您人工介入处理。

3. ONS 是否能保证消息不重复？

不能保证消息不重，但是在绝大部分情况下，消息是不重复的，在网络异常情况下，会出现消息重复。

4. ONS 支持哪些可客户端？

暂时只支持 Java 语言。

C++、.Net、PHP 等语言将来会支持。

5. ONS 客户端是如何找到服务器的？

ONS 客户端内部访问一个特定的域名地址服务器，域名地址服务器向客户端返回服务器地址，客户端即可以直接访问服务器，整个过程不需要用户进行任何配置。

6. ONS 是否可以在公网访问？

支持。

ONS 专门开辟了公网专用集群，供用户测试使用，在 ONS Console 创建 Topic 时，Region 请选择“公网测试”，此 Topic 即可通过公网访问，但是不保证这个 Topic 的可靠性。

在生产环境部署，推荐使用 ECS 来收发消息。公网收发消息不承诺系统的稳定性。

7. ONS 如何收费？

内测及公测期间不收费，后续推出具体收费计划会在阿里云官网通知。

8. 新创建的 ConsumerId 从哪里开始消费？

a) 如果订阅的 topic 第一次发送消息是在 3 天内，那么这个新创建的 ConsumerId 默认从服务器保存的最早消息开始消费

b) 如果订阅的 topic 第一次发送消息是在 3 天前，那么这个新创建的 ConsumerId 默认从服务器保存的最晚消息开始消费，也就是队列尾部开始消费，历史消息不会再消费

c) 如果这个 ConsumerId 是第二次启动，那么从上次消费的位置开始消费

d) 用户如果想从特定位置开始消费，可以通过 ONS Console 上消费进度重置功能，指定到具体的时间开始消费。每次重置只针对特定 ConsumerId 下的特定 Topic，不会影响其他 ConsumerId。

9. 开发团队的联系方式，内测期间有任何问题，欢迎向我们反馈！

ons-dev-team@list.alibaba-inc.com

附录 B 参考文档、规范

- Java Message Service 2.0
<http://jms-spec.java.net>
- Java Message Service API Tutorial
http://docs.oracle.com/javaee/1.3/jms/tutorial/1_3_1-fcs/doc/jms_tutorialTOC.html
- Java(TM) Message Service Specification Final Release 1.1
<http://www.oracle.com/technetwork/java/docs-136352.html>
- CORBA Notification Service Specification 1.1
<http://www.omg.org/spec/NOT/1.1/PDF>
- Distributed Transaction Processing: The XA Specification
<http://pubs.opengroup.org/onlinepubs/009680699/toc.pdf>
- RocketMQ Benchmark
<http://taobao.github.com/metaq/document/benchmark/benchmark.pdf>
- Documentation for /proc/sys/vm/*
<http://www.kernel.org/doc/Documentation/sysctl/vm.txt>