

消息队列 MQ

快速入门

快速入门

快速入门概述

快速入门一章主要描述从开通 MQ 服务、创建 MQ 资源，到使用 MQ SDK 进行消息收发的完整流程，旨在以最简单明了的方式引导您快速上手 MQ，为进一步使用和熟悉 MQ 的功能提供入门。

MQ 快速接入流程图：



1. 步骤一：开通服务
2. 步骤二：创建资源
3. 步骤三：发送消息
4. 步骤四：订阅消息

MQ 支持的接入协议和开发语言：

	Java	C/C++	.NET	PHP	Python
TCP(SDK)	Yes	Yes	Yes	No	No
HTTP	Yes	No	No	Yes	Yes
MQTT	Yes	Yes	Yes	No	Yes

面向服务端场景的应用使用 MQ，推荐用 SDK 方式接入，操作简单且可用性高。而面向移动端的场景，推荐使用 MQTT 方式接入。

MQTT 是移动端物联网领域的标准协议，因此 MQ 不提供 SDK。任何兼容 MQTT 3.1.1协议的客户端都可以接入。MQ 推荐使用第三方 Paho 提供的 SDK，具体请参考文档 MQTT 接入准备。

本章节作为快速入门，以最常用的 TCP 协议下调用 Java SDK 为例来演示如何快速接入 MQ 进行消息收发。

步骤一：开通服务

本文向您介绍 MQ 服务的开通流程。

公共云用户开通服务

公共云用户请按以下步骤开通 MQ 服务：

1. 登录阿里云主页 (<http://www.aliyun.com>)，将鼠标依次移动到**产品 > 互联网中间件**，单击**消息队列**进入 MQ 产品主页。
2. 在 MQ 产品页上，单击**立即开通**进入 MQ 服务开通页面，根据提示完成开通服务。

如果您已经开通 MQ 服务，请直接登录MQ 控制台。

注意：金融云用户开通金融云环境 MQ 服务，请进入工单系统申请。

步骤二：创建资源

在 MQ 消息系统中，消息生产者将消息发送到某个指定的消息主题（Topic），而消息消费者则通过订阅该指定的 Topic 来获取和消费消息。

一个新的应用接入 MQ 需要先创建相关的 MQ 资源，包括：

- 创建消息主题（Topic）
- 创建消息生产者（Producer ID）
- 创建消息消费者（Consumer ID）
- 创建阿里云 Access Key ID 和 Access Key Secret

注意：当您删除资源，比如 Topic、消息生产者、消息消费者的时候，相关的资源也会在 10 分钟内进行清理。

创建 Topic

消息主题（Topic）是 MQ 里对消息进行的一级归类，比如可以创建 “Topic_Trade” 这一主题用来识别交易类消息。使用MQ的第一步需要先为您的应用创建 Topic。

您可以按照以下步骤创建 Topic：

登录MQ 控制台，默认进入**Topic 管理**页面。

在页面上方选择相应的地域（比如**公网域**），然后单击**创建 Topic**按钮。

注意：

- 如果只是测试，或者需要在本地（非阿里云 ECS 服务器）使用 MQ 服务，请将 Topic 创建在公网环境。生产端和消费端可以部署在本地或者部署在任意地域的 ECS 上，前提是本地服务器或者相应的 ECS 需要能够访问公网。
- 如果在生产环境使用 MQ 服务，需要将应用程序部署在阿里云 ECS 上，同时 Topic 也需要在应用程序所在的区域（即所部署的 ECS 区域）进行创建。
- Topic 不能跨域使用。比如 Topic 创建在“华北 2”这个域，那么消息生产端和消费端也必须运行在“华北 2”的 ECS 上。
- 有关域的详细介绍请参见 ECS 文档中的地域和可用区。

在**创建 Topic**对话框输入 Topic 名称及备注，单击**确定**。您创建的 Topic 将出现在 Topic 列表中。

注意：Topic 名称必须全局唯一。如果名称已经被其他用户使用，您将无法创建相同名称的 Topic。

消息类型

消息类型分为以下几种：

- 无序消息：不保证先入先出（FIFO）的严格顺序，包含普通消息、定时/延时消息、事务消息。**建议创建不同的 Topic 来发送不同类型的消息，例如用 Topic A 发送普通消息，Topic B 发送事务消息，Topic C 发送延时/定时消息。**
- 全局顺序消息：所有消息严格按照 FIFO 的严格顺序进行生产和消费。
- 分区顺序消息：消息根据 sharding key 进行分区，提高整体并发度与使用性能。同一个分区的信息严格按照 FIFO 的严格顺序进行生产和消费。
 - Kafka 消息：兼容 MQ 及 Kafka 协议（kafka-client-0.10 及以上版本）的消息类型。

创建 Producer ID

创建好 Topic 后，要为这个 Topic 创建消息生产端的资源，即创建 Producer ID。一个 Topic 只能对应一个 Producer ID。

请按照以下步骤为您的 Topic 创建 Producer ID：

在 MQ 控制台左侧菜单栏选择**Topic 管理**。

在 Topic 列表中找到您刚刚创建的 Topic，单击操作选项中的**创建发布者**。

在**创建生产者**对话框输入 Producer ID，单击**确定**。

注意：

- Producer ID 必须全局唯一。如果名称已存在，您将无法创建相同名称的 Producer ID。
- Topic 对应的生产端必须和这个 Topic 在同一个域，比如您在“公网”域创建了

- “Topic_open” ，那么和 “Topic_open” 对应的 Producer ID 也必须在同一个域。
- Producer ID 和 Topic 的关系是 1:N，即一个 Topic 只能绑定一个 Producer ID, 但是同一个 Producer ID 可以对应多个 Topic。

创建 Consumer ID

创建完消息生产端后，您需要为 Topic 申请相应的消息消费资源，即创建 Consumer ID。

请按以下步骤创建 Consumer ID：

在 MQ 控制台左侧菜单栏选择 **Topic 管理**。

找到您创建的 Topic，单击右侧操作选项里的**创建消费者**。

在**创建消费者**对话框输入 Consumer ID，单击**确定**。

注意：

- Consumer ID 必须全局唯一。如果名称已存在，您将无法创建相同名称的 Consumer ID。
- Consumer ID 必须和对应的 Topic 在同一个域，比如 “公网” 域的 “Topic_open” 可绑定同在 “公网” 域的 Consumer ID “CID_123”，而 “华北1” 域内的 “Topic_huabei1” 则不能绑定该 Consumer ID。
- Consumer ID 和 Topic 的关系是 N:N。同一个 Consumer ID 可以订阅多个 Topic，同一个 Topic 也可以对应多个 Consumer ID。

创建阿里云 AccessKey 和 SecretKey

在调用 SDK/API 进行消息发送和订阅的时候，除了需要指定创建的 Topic，Producer ID 以及 Consumer ID 以外，还需输入您在 RAM 控制台创建的身份验证信息，即 Access Key ID 和 Access Key Secret。

关于如何创建 AccessKey 和 SecretKey, 请参阅[创建AccessKey](#)。

完成以上准备工作后，您就可以运行示例代码，用 MQ 进行消息发送和订阅了。

步骤三：发送消息

在控制台创建 Topic 资源后，您可以通过控制台发送消息或者调用 SDK/API 发送消息。控制台发送消息主要用于快速验证 Topic 资源的可用性，在生产环境下使用 MQ 建议调用相关的 SDK/API 进行消息发送。

通过控制台发送消息

控制台发送消息步骤如下：

在 MQ 控制台的左侧菜单栏单击**生产者管理**。

在列表中找到您刚刚创建的 Topic，单击右侧操作栏里的**发送**。

在**发送消息**对话框输入消息的具体内容，单击**确定**。控制台会返回消息发送成功通知以及相应的 Message ID。

调用 SDK/API 发送消息

在生产环境使用 MQ，建议调用 SDK/API 来进行消息发送。本文以 TCP 协议下调用 Java SDK 为例进行说明。如果需要使用其他协议或者开发语言，请参见相关帮助文档。

调用 TCP Java SDK 发送消息

通过下面两种方式可以引入依赖(任选一种)：

Maven 方式引入依赖：

```
<dependency>
<groupId>com.aliyun.openservices</groupId>
<artifactId>ons-client</artifactId>
<version>1.7.0.Final</version>
</dependency>
```

下载依赖 JAR 包：[下载链接](#)

说明：关于 TCP 接入点域名，请参见[TCP 接入说明](#)。

根据以下说明设置相关参数，运行示例代码：

```
import com.aliyun.openservices.ons.api.Message;
import com.aliyun.openservices.ons.api.Producer;
import com.aliyun.openservices.ons.api.SendResult;
import com.aliyun.openservices.ons.api.ONSFactory;
import com.aliyun.openservices.ons.api.PropertyKeyConst;

import java.util.Properties;
```

```
public class ProducerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // 您在 MQ 控制台创建的 Producer ID
        properties.put(PropertyKeyConst.ProducerId, "XXX");
        // 鉴权用 AccessKey，在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用 SecretKey，在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.SecretKey, "XXX");
        // 设置 TCP 接入域名（此处以公共云的公网接入为例）
        properties.put(PropertyKeyConst.ONSAddr,
            "http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet");

        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();

        //循环发送消息
        while(true){
            Message msg = new Message( //
                // 在控制台创建的 Topic，即该消息所属的 Topic 名称
                "TopicTestMQ",
                // 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 MQ 服务器过滤
                "TagA",
                // Message Body
                // 任何二进制形式的数据，MQ 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 MQ".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 MQ 控制台查询消息并补发
            // 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
            // 打印 Message ID，以便用于消息发送状态查询
            SendResult sendResult = producer.send(msg);
            System.out.println("Send Message success. Message ID is: " + sendResult.getMessageId());
        }

        // 在应用退出前，可以销毁 Producer 对象
        // 注意：如果不销毁也没有问题
        producer.shutdown();
    }
}
```

查看消息是否发送成功

消息发送后，您可以在控制台查看消息发送状态，步骤如下：

在 MQ 控制台左侧菜单栏中单击**消息查询**。

在消息查询页面，选择按 Message ID 查询标签页。

在搜索框中输入发送消息后返回的 Message ID，单击搜索查询消息发送状态。

“储存时间”表示 MQ 服务端存储这条消息的时间。如果查询到此消息，表示消息已经成功发送到服务端。

注意：此步骤演示的是第一次使用 MQ 的场景，此时订阅端从未启动过，所以消息状态显示暂无消费数据。要启动订阅端并进行消息订阅请继续下一步操作订阅消息。更多消息状态请参见消息查询。

步骤四：订阅消息

消息发送成功后，需要启动订阅方进行消息订阅。本文以 TCP Java SDK 为例，介绍如何通过调用相关协议及开发语言的 SDK/API 来完成消息订阅。

说明：关于 TCP 接入点域名，请参见TCP 接入说明。

调用 TCP Java SDK 订阅消息

您可以运行以下示例代码来启动订阅端，并测试订阅消息的功能。请按照说明正确设置相关参数。目前控制台提供了 Java，C++，.NET，PHP 的示例代码。

```
import com.aliyun.openservices.ons.api.Action;
import com.aliyun.openservices.ons.api.ConsumeContext;
import com.aliyun.openservices.ons.api.Consumer;
import com.aliyun.openservices.ons.api.Message;
import com.aliyun.openservices.ons.api.MessageListener;
import com.aliyun.openservices.ons.api.ONFactory;
import com.aliyun.openservices.ons.api.PropertyKeyConst;

import java.util.Properties;

public class ConsumerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // 您在 MQ 控制台创建的 Consumer ID
        properties.put(PropertyKeyConst.ConsumerId, "XXX");
        // 鉴权用 AccessKey，在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用 SecretKey，在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.SecretKey, "XXX");
        // 设置 TCP 接入域名（此处以公共云公网环境接入为例）
        properties.put(PropertyKeyConst.ONAddr,
            "http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet");
    }
}
```



```
Consumer consumer = ONSFactory.createConsumer(properties);
consumer.subscribe("TopicTestMQ", "*", new MessageListener() {
public Action consume(Message message, ConsumeContext context) {
System.out.println("Receive: " + message);
return Action.CommitMessage;
}
});
consumer.start();
System.out.println("Consumer Started");
}
```

查看消息订阅是否成功

完成上述步骤后，您可以在 MQ 控制台查看订阅端是否启动成功，即消息订阅是否成功。目前控制台查看消费者状态仅支持 TCP 客户端，暂不支持 HTTP 以及 MQTT 客户端。

在 MQ 控制台左侧菜单栏单击**消费者管理**。

找到要查看的 Topic，单击右侧操作选项里的**消费者状态**。

如果**是否在线**显示为**是**，则说明订阅端已成功启动。如果消费者状态**是否在线**显示为**否**，说明消费端没有启动或者启动失败。

完成以上所有步骤后，您就成功接入了 MQ 服务，可以用 MQ 进行消息发送和订阅了。