

消息队列 MQ

快速入门

快速入门

主账号 - 快速入门

本文主要描述主账号从开通 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使用主账号的您快速上手 MQ，为进一步使用和熟悉 MQ 的功能提供入门。

消息收发部分以 TCP 协议下调用 Java SDK 为例来演示。

说明：若您要在 TCP 协议下使用 C/C++ 和 .NET SDK 来收发消息，请参见 C/C++ 收发普通消息和.NET 收发普通消息；若要使用 MQTT 协议，请使用微消息队列概述。

MQ 快速接入流程图：



步骤一：开通服务

请按照以下步骤开通 MQ 服务：

登录阿里云主页，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。

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。

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：

- 主题（Topic）

- 生产者 (Producer ID)
- 消费者 (Consumer ID)
- 阿里云 AccessKey

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。

创建消息主题 (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。

在页面上方选择地域（比如公网域），即您要将资源创建在哪个地域。

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地域（即所部署的 ECS 地域）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。
- 有关地域的详细介绍请参见 ECS 文档中的地域和可用区。

在左侧导航栏，选择 **Topic 管理**。

单击**创建 Topic**按钮。

在**创建 Topic**对话框中的 **Topic** 一栏，输入 Topic 名称。

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。

在**消息类型**一栏，选择该 Topic 对应的消息类型，即该 Topic 用来收发何种类型的消息。

消息类型说明：

- 普通消息：无特性的消息，区分于事务消息、定时/延时消息和顺序消息。
- 事务消息：提供类似 X/Open XA 的分布事务功能，能达到分布式事务的最终一致。
- 定时/延时消息：可指定消息延迟投递，即在未来的某个特定时间点或一段特定的时间后进行投递。

- 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。同一个分区的消息严格按照 FIFO 的严格顺序进行生产和消费。
- 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。

注意：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。

在**描述**一栏，输入该 Topic 的备注内容，然后单击**确定**。您创建的 Topic 将出现在 Topic 列表中。

创建生产者 (Producer ID)

创建好 Topic 后，要为此 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 控制台的左侧导航栏，选择**Topic 管理**。

在**Topic 管理**页面，找到您刚刚创建的 Topic，单击**操作**列的**创建生产者**。

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。

创建消费者 (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 控制台的左侧导航栏，选择 **Topic 管理**。

在**Topic 管理**页面，找到您创建的 Topic，单击右侧**操作**列的**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

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 AccessKey。AccessKey 的信息包含 AccessKeyId 和 AccessKeySecret。

关于如何创建 AccessKey，请参见[创建 AccessKey](#)。

步骤三：获取接入域名

在控制台创建好资源后，您还需要通过控制台获取 Producer ID 的接入域名和 Consumer ID 的接入域名。

请按照以下步骤获取 Producer ID 的接入域名或 Consumer ID 的接入域名：

在 MQ 控制台的左侧导航栏，选择**生产者管理**或**消费者管理**。

找到您创建的 Producer ID 或 Consumer ID，单击右侧操作列中的**获取接入点**。

在**获取接入点**提示框，单击**复制**。

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。

步骤四：发送消息

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：用于快速验证 Topic 资源的可用性。

调用 SDK/API 发送消息：用于生产环境下使用 MQ。

通过控制台发送消息

控制台发送消息步骤如下：

在 MQ 控制台的左侧导航栏，选择**生产者管理**。

在**生产者管理**页面，找到您刚刚创建的 Topic，单击右侧**操作列**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>"XXX"</version>
//设置为 Java SDK 的最新版本号
</dependency>
```

关于 Java SDK 的最新版本号，请查看版本说明。

下载依赖 JAR 包：

关于 Java SDK 最新版本的下载链接，请查看版本说明。

根据以下说明设置相关参数，运行示例代码：

说明：关于 TCP 接入点域名，请进入 MQ 控制台的**生产者管理**页面，在 PID 右侧**操作列**单击**获取接入点按钮**获取。

```
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, 进入 MQ 控制台的生产者管理页面, 在右侧操作列单击获取接入点获取
        // 此处以公有云公网地域接入点为例
        properties.put(PropertyKeyConst.ONSAAddr,
            "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, 以便于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](#)。

步骤五：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。

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 的示例代码。

说明：关于 TCP 接入点域名，请进入 MQ 控制台的[消费者管理](#)页面，在 CID 右侧操作列单击[获取接入点](#)按钮获取。

```
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.ONSType;
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，进入 MQ 控制台的消费者管理页面，在右侧操作列单击获取接入点获取
        // 此处以公有云公网地域接入点为例
        properties.put(PropertyKeyConst.ONSType, "TCP");
        properties.put(PropertyKeyConst.ONSAddr,
            "http://onsaddr-internet.aliyun.com/rocketmq/nsaddr4client-internet");

        Consumer consumer = ONSType.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 控制台左侧导航栏，选择**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。

子账号 - 快速入门

如果您使用的是 RAM 子账号，在被主账号授权某 Topic 后，不可直接使用主账号为该 Topic 创建的生产者和消费者，否则会导致心跳连接异常。因此，您需要先登录 MQ 控制台查看被授权的 Topic，并为授权的 Topic 另外创建相应的生产者和消费者，然后才可通过 SDK 进行消息的收发。

本文将为您提供完整详细的操作指导，其中的消息收发部分以 TCP 协议下调用 Java SDK 为例来演示。

说明：

关于 RAM 主子账号的详细说明，请参见RAM 主子账号授权。

若您要在 TCP 协议下使用 C/C++ 和 .NET SDK 来收发消息，请参见 C/C++ 收发普通消息和.NET 收发普通消息；若要使用 MQTT 协议，请使用微消息队列（MQ for IoT）。

MQ 快速接入流程图：



前提条件

您的 RAM 子账号已具备以下条件：

主账号已为该 RAM 子账号创建了 AccessKey (AK)。

若还未创建，主账号需参考创建 RAM 用户中的**为用户创建 AK**章节进行创建。

您调用 SDK/API 收发消息时需使用 AK 进行身份验证。

主账号已给该 RAM 子账号授予了 Topic 资源的操作权限。

若还未授权，主账号需参考 RAM 主子账号授权中的**授权策略**和**给 RAM 用户授权**章节给该 RAM 子账号进行授权。

步骤一：查看 Topic

请按照以下步骤查看您的 RAM 子账号所被授权的 Topic：

登录RAM 控制台。

在 RAM 控制台左侧的产品列表中找到**消息队列**，点击即可跳转到 MQ 控制台。

若您的 RAM 子账号已在登录状态，可直接点击MQ 控制台MQ 控制台 进入。

在左侧导航栏，选择 **Topic 管理**，即可看到所有被授权的 Topic。

注意： **Topic 管理**页面会显示被授权的所有 Topic，但根据主账号所授权限的不同，可能对于某些 Topic 您只拥有为其创建生产者的权限，但不具备相应地创建消费者的权限，反之亦然。详情请见下文描述。

步骤二：创建生产者和消费者

由于 RAM 子账号被授予的权限会不同，因此创建生产者和消费者的权限也会有所不同。若您的 RAM 子账号对某 Topic：

仅拥有发布权限：仅可创建生产者，不可创建消费者

仅拥有订阅权限：仅可创建消费者，不可创建生产者

同时拥有发布和订阅权限：既可创建生产者，也可创建消费者

如果您的 RAM 子账号拥有 MQ 的管理权限，即相当于主账号的权限，那么您不仅可以创建生产者和消费者，还可以创建 Topic。关于如何创建 Topic，请参见主账号 - 快速入门中的**步骤二：创建资源**章节里关于 Topic 创建的内容。

更多关于 RAM 子账号的权限类型的说明，请参见RAM 主子账号授权中的**授权策略**。

创建生产者 (Producer ID)

若要给某个 Topic 发送消息，则需要为这个 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 控制台左侧导航栏，选择**Topic 管理**。

在**Topic 管理**页面，找到目标 Topic，单击右侧**操作**列的**创建生产者**。

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。

创建消费者 (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**。

在**Topic 管理**页面，找到目标 Topic，单击右侧**操作**列的**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。

步骤三：获取接入域名

在控制台创建好资源后，您还需要通过控制台获取 Producer ID 的接入域名和 Consumer ID 的接入域名。

请按照以下步骤获取 Producer ID 的接入域名或 Consumer ID 的接入域名：

在 MQ 控制台左侧导航栏，选择**生产者管理**或**消费者管理**。

找到您创建的 Producer ID 或 Consumer ID，单击右侧操作列中的**获取接入点**。

在**获取接入点**提示框，单击**复制**。

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。

步骤四：发送消息

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：用于快速验证 Topic 资源的可用性。

调用 SDK/API 发送消息：用于生产环境下使用 MQ。

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**生产者管理**页面，找到目标 Topic，单击右侧操作列的**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>"XXX"</version>
//设置为 Java SDK 的最新版本号
</dependency>
```

关于 Java SDK 的最新版本号，请查看版本说明。

下载依赖 JAR 包：

关于 Java SDK 最新版本的下载链接，请查看版本说明。

根据以下说明设置相关参数，运行示例代码：

说明：关于 TCP 接入点域名，请进入 MQ 控制台的生产者管理页面，在 PID 右侧操作列单击**获取接入点按钮**获取。

```
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");
        // 鉴权用的 RAM 子账号的 AccessKeyId，由主账号创建，请向主账号获取
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用的 RAM 子账号的 AccessKeySecret，由主账号创建，请向主账号获取
        properties.put(PropertyKeyConst.SecretKey, "XXX");
        // 设置 TCP 接入域名，进入 MQ 控制台的生产者管理页面，在右侧操作列单击获取接入点获取
```

```
// 此处以公有云公网地域接入点为例
properties.put(PropertyKeyConst.ONSAAddr,
"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。

步骤五：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。

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 的示例代码。

说明：关于 TCP 接入点域名，请进入 MQ 控制台的[消费者管理](#)页面，在 CID 右侧操作列单击[获取接入点](#)按钮获取。

```
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.ONSFactory;
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");
        // 鉴权用的 RAM 子账号的 AccessKeyId，由主账号创建，请向主账号获取
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用的 RAM 子账号的 AccessKeySecret，由主账号创建，请向主账号获取
        properties.put(PropertyKeyConst.SecretKey, "XXX");
        // 设置 TCP 接入域名，进入 MQ 控制台的消费者管理页面，在右侧操作列单击获取接入点获取
        // 此处以公有云公网地域接入点为例
        properties.put(PropertyKeyConst.ONSDAddr,
            "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 控制台左侧导航栏，选择**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。

使用限制

MQ 对某些具体指标进行了约束和规范，您在使用 MQ 时注意不要超过相应的限制值，以免程序出现异常。具体的限制项和限制值请参见下表。

限制项	限制值	说明
Topic 名称长度	64 个字符	Topic 名称长度不得超过该限制，否则会导致无法发送或者订阅。
消息大小	4 MB 字节	消息大小不得超过该限制，否则消息会被丢弃。
消息保存时间	3 天	消息最多保留 3 天，超过时间将自动滚动删除。
消费位点重置	3 天	支持重置消费 3 天之内任何时间点的消息。
单 Topic 的消息收发 TPS	企业铂金版 ：参考所购买的规格 标准版 ：5000 条/秒	无
定时/延时消息的延时时长	40 天	msg.setStartDeliverTime 的参数可设置 40 天内的任何时刻（单位毫秒），超过 40 天消息发送将失败。