

消息队列 MQ

快速入门

快速入门

主账号快速入门

如果您使用的是阿里云主账号，则可以通过本文来体验从开通服务、创建资源、到使用 SDK 收发消息的完整流程，快速上手消息队列 MQ。

无论您使用的是消息队列 MQ 支持的何种协议、何种语言，前三个步骤都一致，只是在控制台上具体填写的信息会略有不同，请以控制台说明为准。但在调用 SDK 时，不同协议和语言的示例代码有所不同，本文以 TCP 协议下的 Java SDK 为例进行说明。

步骤一：开通服务

如果您已开通消息队列 MQ 服务，请直接跳转至步骤二：创建资源开始操作。

在消息队列 MQ 产品页，单击**立即开通**。

页面跳转至**确认订单**页面。

在**确认订单**页面，选择**我已阅读并同意《消息队列MQ服务协议》**，再单击**立即开通**即可完成开通。

步骤二：创建资源

在使用消息队列 MQ 时，请注意以下网络访问限制：

Topic 和 Group ID 需创建在同一个地域（Region）下的同一个实例中才能互通。例如，当某 Topic 创建在**华东1（杭州）**下的**实例 A**中，那么该 Topic 只能被在**华东1（杭州）**下的**实例 A**中创建的 Group ID 对应的生产端和消费端访问。

如果只是测试，或者需要在本地（非阿里云 ECS 服务器）使用消息队列 MQ 的服务，请将 Topic 和 Group ID 都创建在**公网**地域下的实例中。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。

地域的详细介绍请参见[地域和可用区](#)。

创建实例

实例是用于消息队列 MQ 服务的虚拟机资源，会存储消息主题（Topic）和客户端 ID（Group ID）信息。

登录消息队列 MQ 控制台。在页面顶部导航栏，选择地域，如公网地域。

在左侧导航栏，单击**实例管理**。

在**实例管理**页面右上角，单击**创建实例**按钮。

在**创建实例**对话框，选择**实例类型**，并输入**实例名**和**描述**，然后单击**确认**。

注意：当您删除某实例时，该实例中的所有 Topic 和 Group ID 也会在 10 分钟内被清理；若单独删除 Topic 或 Group ID，则不会对其他资源造成影响。

创建 Topic

Topic 是消息队列 MQ 里对消息的一级归类，比如可以创建 Topic_Trade 这一 Topic 来识别交易类消息，消息生产者将消息发送到 Topic_Trade，而消息消费者则通过订阅该 Topic 来获取和消费消息。

在控制台左侧导航栏，单击**Topic 管理**。

在**Topic 管理**页面上方选择刚创建的实例，单击**创建 Topic**按钮。

在**创建 Topic**对话框中的**Topic**一栏，输入 Topic 名称，选择该 Topic 对应的消息类型，输入该 Topic 的备注内容，然后单击**确定**。

消息类型的更多信息，请参见[消息类型](#)。

注意：

Topic 不能跨实例使用，比如在实例 A 中创建的 Topic A 不能在实例 B 中使用。

Topic 名称必须在同一实例中是唯一的。

您可创建不同的 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 发送定时/延时消息。

您创建的 Topic 将出现在 Topic 列表中。

创建 Group ID

创建完实例和 Topic 后，您需要为消息的消费者（或生产者）创建客户端 ID，即 Group ID 作为标识。

说明：消费者必须有对应的 Group ID，生产者不做强制要求。

在控制台左侧导航栏，单击 **Group 管理**。

在 **Group 管理**页面上方选择刚创建的实例，然后选择 **TCP 协议 > 创建 Group ID**。本文以 TCP 协议为例。

说明：TCP 和 HTTP 协议下的 Group ID 不可以共用，因此需分别创建。

在**创建 Group ID**对话框中，输入 Group ID 和描述，然后单击**确认**。

注意：

Group ID 必须在同一实例中是唯一的。

Group ID 和 Topic 的关系是 N : N，即一个消费者可以订阅多个 Topic，同一个 Topic 也可以被多个消费者订阅；一个生产者可以向多个 Topic 发送消息，同一个 Topic 也可以接收来自多个生产者的消息。

创建阿里云 AccessKey

阿里云 AccessKey 用于收发消息时进行账户鉴权。

在调用 SDK 发送和订阅消息的时候，除了需要指定创建的 Topic 和 Group 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。

创建 AccessKey 的具体步骤请参见[创建AccessKey](#)。

步骤三：获取接入点

在控制台创建好资源后，您需通过控制台获取实例的接入点。在收发消息时，您需要为生产端和消费端配置该接入点，以此接入某个具体实例或地域的服务。

在控制台左侧导航栏，单击**实例管理**。

在**实例管理**页面上方选择刚创建的实例。

在默认显示的**实例信息**页签的**获取接入点信息**区域，您可以分别看到新创建实例的 TCP 和 HTTP 协

议接入点。接入点性质因协议而异，具体说明如下：

TCP 协议：您在控制台看到的 TCP 协议接入点是地域下某个具体实例的接入点。同一地域下的不同实例的接入点各不相同。

HTTP 协议：您在控制台看到的 HTTP 协议接入点是某个地域的接入点，跟具体实例无关。您在收发消息时还需另外设置实例 ID。

在 TCP 协议的接入点区域，单击**复制**。

对于 TCP 协议的接入点，您还可以单击**示例代码**，查看在各种开发语言的程序中如何设置接入点。

完成以上准备工作后，您就可以运行示例代码，用消息队列 MQ 进行消息发送和订阅了。

步骤四：发送消息

您可以通过以下方式发送消息：

控制台发送消息：用于快速验证 Topic 资源的可用性，主要用作测试。

调用 SDK 发送消息：用于生产环境下使用消息队列 MQ。

通过控制台发送消息

在控制台左侧导航栏，单击 **Topic 管理**。

在 **Topic 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**对话框中的 **Message Body** 一栏，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 发送消息

下文以调用 TCP Java SDK 为例进行说明。

调用 TCP Java SDK 发送消息

通过以下任一方式引入依赖：

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>
```

Java SDK 的最新版本号，请参见 Java SDK 版本说明。

下载依赖 JAR 包：

Java SDK 最新版本的下载链接，请参见 Java SDK 版本说明。

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.ONSTFactory;
import com.aliyun.openservices.ons.api.PropertyKeyConst;

import java.util.Properties;

public class ProducerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // 您在控制台创建的 Group ID
        properties.put(PropertyKeyConst.GROUP_ID, "XXX");
        // 鉴权用 AccessKeyId，在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用 AccessKeySecret，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，进入控制台的实例管理页面，在页面上方选择实例后，在实例信息中的“获取接入点信息”区域查看
        properties.put(PropertyKeyConst.NAMESRV_ADDR, "XXX");

        Producer producer = ONSTFactory.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 指定过滤条件在消息队列 MQ 服务器过滤
                "TagA",
                // Message Body
```

```
// 任何二进制形式的数据，消息队列 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：

在控制台左侧导航栏，选择**消息查询** > **按 Message ID 查询**。

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。

储存时间表示消息队列 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消息队列 MQ 的场景，此时消费者从未启动过，所以消息状态显示暂无消费数据。要启动消费者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**和**查询消息轨迹**。

步骤五：调用 SDK 订阅消息

消息发送成功后，需要启动消费者来订阅消息。下文以调用 TCP Java SDK 为例说明如何订阅消息。

调用 TCP Java SDK 订阅消息

您可以运行以下示例代码来启动消费者，并测试订阅消息的功能。请按照说明正确设置相关参数。

```
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.ONSTactory;
import com.aliyun.openservices.ons.api.PropertyKeyConst;

import java.util.Properties;

public class ConsumerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // 您在控制台创建的 Group ID
        properties.put(PropertyKeyConst.GROUP_ID, "XXX");
        // 鉴权用 AccessKeyId, 在阿里云服务器管理控制台创建
        properties.put(PropertyKeyConst.AccessKey, "XXX");
        // 鉴权用 AccessKeySecret, 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, 进入控制台的实例管理页面, 在页面上方选择实例后, 在实例信息中的“获取接入点信息”区域查看
        properties.put(PropertyKeyConst.NAMESRV_ADDR, "XXX");

        Consumer consumer = ONSTactory.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，您可以在控制台查看消费者是否启动成功，即消息订阅是否成功。

在控制台左侧导航栏，单击 **Group 管理**。

找到要查看的 Group ID，单击该 Group ID 所在行操作列的**订阅关系**。

如果**是否在线**显示为**是**，且订阅关系一致，则说明订阅成功。否则说明订阅失败。

更多信息

子账号的操作步骤请参见子账号快速入门。

如需了解文中所涉及的名词概念，请参见名词解释。

若您需要使用 TCP 协议下的其他语言收发消息，请参见以下文档：

C/C++

收发普通消息

订阅消息

.NET

收发普通消息

订阅消息

若您需要使用 HTTP 协议收发消息，请参见 HTTP 协议了解相关注意事项、获取多语言（包含 Java、C++、Python 等七种语言）的 SDK 和示例代码。

子账号快速入门

如果您使用的是 RAM 子账号，在被主账号授予某些实例中的 Topic 的权限后，不可直接使用主账号创建的 Group ID。请先登录消息队列 MQ 控制台查看被授权的实例和 Topic，并另外创建 Group ID，即可调用 SDK 收发消息。

RAM 主子账号的详细说明请参见 RAM 主子账号授权。

前提条件

由于您调用 SDK 收发消息时需使用 AccessKey（包含 AccessKeyId 和 AccessKeySecret）进行身份验证，请确保主账号已为您的 RAM 子账号授予了编程访问权限，允许该 RAM 子账号启用 AccessKey。详情请参见创建 RAM 用户。

该 RAM 子账号已被授予了指定 Topic 的操作权限。

若还未获取授权，请联系主账号参见 RAM 主子账号授权进行授权。

步骤一：查看已被授权的实例和 Topic

在浏览器中打开 RAM 用户登录入口，并按提示完成登录。

在产品列表中找到**消息队列 MQ**，并单击**消息队列 MQ**。

页面跳转至消息队列 MQ 控制台。

在控制台左侧导航栏，单击**实例管理**查看 RAM 子账号被授权的实例及其基本信息。

在控制台左侧导航栏，单击**Topic 管理**查看 RAM 子账号被授权的 Topic。

步骤二：创建资源

创建 Group ID

查看了您的 RAM 子账号拥有哪些实例和 Topic 的操作权限后，您需要为消息的消费者（或生产者）创建客户端 ID，即 Group ID。

说明：消费者必须有对应的 Group ID，生产者不做强制要求。

在控制台左侧导航栏，单击**Group 管理**。

在**Group 管理**页面上方选择已被授权的实例，然后选择**TCP 协议 > 创建 Group ID**。本文以 TCP 协议为例。

说明：TCP 协议和 HTTP 协议下的 Group ID 不可以共用，因此需分别创建。

在**创建 Group ID**对话框中，输入 Group ID 和描述，然后单击**确认**。

注意：

Group ID 必须在同一实例中是唯一的。

Group ID 和 Topic 的关系是 N : N，即一个消费者可以订阅多个 Topic，同一个 Topic 也可以被多个消费者订阅；一个生产者可以向多个 Topic 发送消息，同一个 Topic 也可以接收来自多个生产者的消息。

创建阿里云 AccessKey

阿里云 AccessKey 用于收发消息时进行账户鉴权。

在调用 SDK 发送和订阅消息的时候，除了需要指定创建的 Topic 和 Group ID 以外，还需输入您在 RAM 控制

台创建的身份验证信息，即 AccessKey。

创建 AccessKey 的具体步骤请参见[创建AccessKey](#)。

步骤三：获取接入点

在控制台创建好资源后，您还需要通过控制台获取实例或地域的接入点。在收发消息时，您需要为生产端和消费端配置该接入点，以此接入某个具体实例或地域的服务。

在控制台左侧导航栏选择**实例管理**。

在**实例管理**页面上方选择刚创建的实例。

在默认显示的**实例信息**页签的**获取接入点信息**区域，您可以分别看到新创建实例的 TCP 和 HTTP 协议接入点。接入点性质因协议不同而不同，具体说明如下：

TCP 协议：您在控制台看到的 TCP 协议接入点是地域下某个具体实例的接入点。同一地域下的不同实例的接入点各不相同。

HTTP 协议：您在控制台看到 HTTP 协议接入点是某个地域的接入点，跟具体实例无关。您在收发消息时还需另外设置实例 ID。

在 TCP 协议的接入点区域，单击**复制**。

对于 TCP 协议的接入点，您还可以单击**示例代码**，查看在各种开发语言的程序中如何设置接入点。

完成以上准备工作后，您就可以运行示例代码，用消息队列 MQ 进行消息发送和订阅了。

步骤四：发送消息

您可以通过以下方式发送消息：

控制台发送消息：用于快速验证 Topic 资源的可用性，主要用作测试。

调用 SDK 发送消息：用于生产环境下使用消息队列 MQ。

通过控制台发送消息

在控制台左侧导航栏，选择 **Topic 管理**。

在 **Topic 管理** 页面，找到您刚刚创建的 Topic，单击右侧**操作列**的**发送**。

在**发送消息**对话框中的 **Message Body** 一栏，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 发送消息

下文以调用 TCP Java SDK 为例进行说明。

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>
```

Java SDK 的最新版本号，请参见 Java SDK 版本说明。

下载依赖 JAR 包：

Java SDK 最新版本的下载链接，请参见 Java SDK 版本说明。

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();
        // 您在控制台创建的 Group ID
        properties.put(PropertyKeyConst.GROUP_ID, "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，进入控制台的实例管理页面，在页面上方选择实例后，在实例信息中的“获取接入点信息”区域查看
properties.put(PropertyKeyConst.NAMESRV_ADDR, "XXX");

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 指定过滤条件在消息队列 MQ 服务器过滤
    "TagA",
    // Message Body
    // 任何二进制形式的数据，消息队列 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，可通过消息队列 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：

在控制台左侧导航栏，选择**消息查询** > **按 Message ID 查询**。

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。

储存时间表示消息队列 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消息队列 MQ 的场景，此时消费者从未启动过，所以消息状态显示暂无消费

数据。要启动消费者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](#)和[查询消息轨迹](#)。

步骤五：调用 SDK 订阅消息

消息发送成功后，需要启动消费者来订阅消息。下文以调用 TCP Java SDK 为例说明如何订阅消息。

调用 TCP Java SDK 订阅消息

您可以运行以下示例代码来启动消费者，并测试订阅消息的功能。请按照说明正确设置相关参数。

```
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.ONSTFactory;
import com.aliyun.openservices.ons.api.PropertyKeyConst;

import java.util.Properties;

public class ConsumerTest {
    public static void main(String[] args) {
        Properties properties = new Properties();
        // 您在控制台创建的 Group ID
        properties.put(PropertyKeyConst.GROUP_ID, "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, 进入控制台的实例管理页面, 在页面上方选择实例后, 在实例信息中的“获取接入点信息”区域查看
        properties.put(PropertyKeyConst.NAMESRV_ADDR, "XXX");

        Consumer consumer = ONSTFactory.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，您可以在控制台查看消费者是否启动成功，即消息订阅是否成功。

在控制台左侧导航栏，单击 **Group 管理**。

找到要查看的 Group ID，单击该 Group ID 所在行**操作**列的**订阅关系**。

如果**是否在线**显示为**是**，且订阅关系一致，则说明订阅成功。否则说明订阅失败。

更多信息

主账号的操作步骤请参见主账号快速入门。

如需了解文中所涉及的名词概念，请参见名词解释。

若您需要使用 TCP 协议下的其他语言收发消息，请参见以下文档：

C/C++

收发普通消息

订阅消息

.NET

收发普通消息

订阅消息

若您需要使用 HTTP 协议收发消息，请参见 HTTP 协议了解相关注意事项、获取多语言（包含 Java、C++、Python 等七种语言）的 SDK 和示例代码。