消息队列 RocketMQ

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服务、创建资源，到使用 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消息队列 RocketMQ，为进一步使用和熟悉产品功能提供入门。

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 for IoT。

消息队列 RocketMQ 快速接入流程图（主账号）：

步骤一：开通服务

请按照以下步骤开通消息队列 RocketMQ 服务：

登录阿里云主页，将鼠标依次移动到**产品** > **企业应用** > **消息队列 MQ** ，单击**消息队列 RocketMQ** 进入消息队列 RocketMQ 的产品主页。

在消息队列 RocketMQ 的产品主页上，单击**立即开通**进入消息队列 RocketMQ 服务开通页面，根据提示完成开通服务。

如果您已经开通消息队列 RocketMQ 服务，请直接登录消息队列 RocketMQ 控制台。

步骤二：创建资源

资源类型说明

一个新的应用接入消息队列 RocketMQ 需要先创建相关的消息队列 RocketMQ 资源，包括：

实例：用于消息队列 RocketMQ 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。

消息主题（Topic）：在消息队列 RocketMQ 的消息系统中，消息生产者将消息发送到某个指定的 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。

Group ID：用于消息消费者（或生产者）的标识

阿里云 AccessKey：用于收发消息时进行账户鉴权

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。

网络访问说明

在使用消息队列 RocketMQ 时，请注意以下网络访问限制：

只有在同一个地域下的同一个实例中的 Topic 和 Group ID 才能互通，即某 Topic 是在哪个地域的哪个实例中创建的，它就只能被同样在该地域下的该实例中的创建的 Group ID 对应的生产端和消费端访问。

例如，当某 Topic 是创建在**华北 2**下的**实例 A** 中，那么该 Topic 只能被在**华北 2**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 RocketMQ 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。但注意遵循上一条原则，Topic 不能跨实例使用。

有关地域的详细介绍请参见 ECS 文档中的地域和可用区。

选择地域

登录消息队列 RocketMQ 控制台。

在页面上方选择地域（比如**公网**地域），即您要将资源创建在哪个地域。

创建实例

在控制台左侧导航栏选择**实例管理**。

在**实例管理**页面，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**确定**。

创建消息主题（Topic）

消息主题（Topic）是消息队列 RocketMQ 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。

在控制台左侧导航栏选择**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。

注意：Topic 名称必须在同一实例中是唯一的。

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 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。

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选择**Group 管理**。

在**Group 管理**页面上方选择刚创建的实例。

单击**创建 Group ID**。

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

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

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 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 AcessKeySecret。

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生产者和消费者的接入域名。

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，您可以在**TCP 接入点**一栏看到接入点域名。

在**TCP接入点**一栏，单击**复制**。

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 RocketMQ 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消息队列 RocketMQ。

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/API 发送消息

在生产环境使用消息队列 RocketMQ，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：

```
 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 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 MQ 服务器过滤
"TagA",
// Message Body
// 任何二进制形式的数据， 消息队列 RocketMQ 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中选择**消息查询**。

在**消息查询**页面，选择**按 Message ID 查询**页签。

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 RocketMQ 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 RocketMQ 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**。

步骤五：订阅消息

消息发送成功后，需要启动消费者进行消息订阅。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。目前控制台提供了 Java，C++, .NET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.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();
// 您在控制台创建的 Group ID
properties.put(PropertyKeyConst.GROUP_ID, "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 = 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选择**Group 管理**。

找到要查看的消费者的 Group ID，单击右侧**操作**列里的**消费者状态**。

如果状态为**在线**，则说明消费者已成功启动。如果状态为**离线**，说明消费者没有启动或者启动失败。

完成以上所有步骤后，您就成功接入了消息队列 RocketMQ 服务，可以用消息队列 RocketMQ 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，在被主账号授权某些实例中的 Topic 后，不可直接使用主账号创建的 Group ID。请先登录消息队列 RocketMQ 控制台查看被授权的实例和 Topic，并另外创建相应的 Group ID，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 for IoT。

消息队列 RocketMQ 快速接入流程图（子账号）：

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 用户中的相关内容进行创建。

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：查看被授权的实例和 Topic

请按照以下步骤查看您的 RAM 子账号所被授权的实例 和 Topic。您收发消息时就可以使用这些被授权的实例和 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** ，点击即可跳转到消息队列 RocketMQ 控制台。

若您的 RAM 子账号已在登录状态，可直接点击消息队列 RocketMQ 控制台 进入。

在左侧导航栏选择 **实例管理**，查看 RAM 子账号被授权的实例及其基本信息。

在左侧导航栏选择 **Topic 管理**，查看 RAM 子账号被授权的 Topic。

步骤二：创建 Group ID

查看了您的 RAM 子账号对哪些实例和 Topic 后拥有权限后，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。

请按以下步骤创建 Group ID：

在控制台左侧导航栏选择**Group 管理**。

在**Group 管理**页面上方选择刚创建的实例。

单击**创建 Group ID**。

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

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。

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生产者和消费者的接入域名。

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，您可以在**TCP 接入点**一栏看到接入点域名。

在**TCP接入点**一栏，单击**复制**。

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 RocketMQ 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消息队列 RocketMQ。

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/API 发送消息

在生产环境使用消息队列 RocketMQ，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：

```
 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 RocketMQ 服务器过滤
"TagA",
// Message Body
// 任何二进制形式的数据， 消息队列 RocketMQ 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 RocketMQ 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中选择**消息查询**。

在**消息查询**页面，选择**按 Message ID 查询**页签。

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 RocketMQ 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 RocketMQ 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**。

步骤五：订阅消息

消息发送成功后，需要启动消费者进行消息订阅。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。目前控制台提供了 Java，C++, .NET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.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 = 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选择**Group 管理**。

找到要查看的消费者的 Group ID，单击右侧**操作**列里的**消费者状态**。

如果状态为**在线**，则说明消费者已成功启动。如果状态为**离线**，说明消费者没有启动或者启动失败。

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

使用限制

消息队列 RocketMQ 对某些具体指标进行了约束和规范，您在使用消息队列 RocketMQ 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。