

ALIBABA CLOUD

# 阿里云

## 微消息队列MQTT版 快速入门

文档版本：20210203

 阿里云

## 法律声明

阿里云提醒您,在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击确定。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.快速使用MQTT的Java SDK收发消息（无跨产品数据互通）	05
2.快速使用MQTT的Java SDK收发消息（跨产品数据流入）	10
3.快速使用MQTT的Java SDK收发消息（跨产品数据流出）	16
4.快速使用MQTT的Java SDK收发消息（客户端上下线通知）	22

# 1.快速使用MQTT的Java SDK收发消息 (无跨产品数据互通)

本文介绍如何快速使用微消息队列MQTT版的Java SDK实现MQTT客户端消息的自发自收。本场景示例不涉及跨云产品的数据互通。

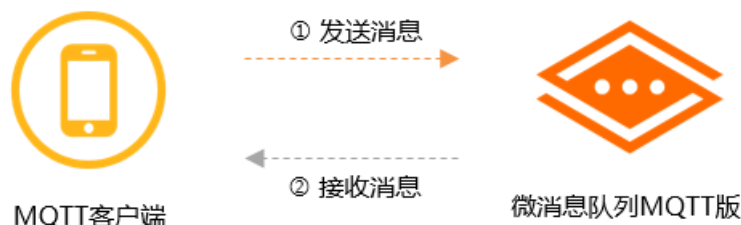
## 前提条件

- 安装IDE。您可以使用IntelliJ IDEA或者Eclipse，本文以IntelliJ IDEA为例。
- 下载安装JDK。

## 背景信息

微消息队列MQTT版最简单的使用场景即是MQTT客户端消息的自发自收。此场景下可使用多语言的第三方开源SDK来实现消息收发。目前支持的语言详情请参见[SDK下载](#)。

本文以公网环境中的Java SDK为例说明如何实现MQTT客户端消息的自发自收。



如上图所示，在物联网和移动互联网场景中，您部署在公网的MQTT客户端采用Java语言开发，需要与阿里云MQTT服务端实现消息收发。那么您需要在您的MQTT客户端代码中，嵌入Java SDK的相应代码，并在SDK中配置相应参数以与MQTT服务端通信。

## 网络访问

微消息队列MQTT版同时提供了公网接入点、经典网接入点和VPC接入点。接入点说明如下：

- 在物联网和移动互联网的场景中，客户端推荐使用公网接入点接入。
- 经典网接入点和VPC接入点仅供一些特殊场景使用。因为一般而言，涉及部署在云端服务器上的应用的场景，建议使用服务端消息产品例如

消息队列RocketMQ版

实现。

**注意** 客户端使用接入点连接服务时务必使用域名接入，不得直接使用域名背后的IP地址直接连接，因为IP地址随时会变化。在以下使用情况中出现的问题微消息队列MQTT版产品方概不负责：

- 客户端不使用域名接入而是使用IP地址接入，产品方更新了域名解析导致原有IP地址失效。
- 客户端网络对IP地址设置网络防火墙策略，产品方更新了域名解析后新IP地址被您的防火墙策略拦截。

本文以公网接入点为例。微消息队列MQTT版与

消息队列RocketMQ版

的应用场景对比和消息属性映射关系请参见以下文档：

- [MQTT与Rocket的应用场景对比](#)
- [MQTT与Rocket的消息结构映射](#)

## 使用流程

调用MQTT的Java SDK收发消息的流程如下图所示。



### 步骤一：创建实例并获取接入点

1. 登录[微消息队列MQTT版控制台](#)，在顶部菜单栏选择实例所在地域。
2. 在概览页面的实例列表区域，[点击这里](#)创建您的第一个实例。
3. 在售卖页面，按需选择您所需的实例规格，单击[立即购买](#)。您可购买包年包月或按量付费实例，两种实例的计费详情请参见[计费说明](#)。此处以按量付费实例为例。



4. 在[确认订单](#)页面，选中[微消息队列for IoT（按量付费）服务协议](#)，单击[立即开通](#)。此处以按量付费实例为例。



支付完成后，即可看到以下界面，说明已购买成功。



5. 回到[微消息队列MQTT版控制台](#)，在顶部菜单栏将地域切换为您所购买的实例所对应的地域。
6. 在左侧导航栏，选择实例详情 > 实例信息。
7. 在实例信息页签的获取接入点信息 区域，即可看到接入点信息，本文示例以公网接入点为例。

## 步骤二：创建父级Topic

MQTT协议支持多级Topic，父级Topic需在控制台或调用OpenAPI创建，子级Topic无需创建，Topic详情请参见[名词解释](#)。本文以在控制台创建父级Topic为例。

1. 在左侧导航栏，选择Topic 管理 > 创建 Topic。
2. 在创建 Topic对话框，输入Topic以及描述，单击确认。  
您可以在Topic 管理页面查看刚创建的Topic。

## 步骤三：创建Group ID

Group ID详情请参见[名词解释](#)。

1. 在左侧导航栏，选择Group 管理 > 创建 Group ID。
2. 在创建 Group ID对话框，输入Group ID，然后单击确认。  
您可以在Group ID列表中查看到刚创建的Group ID。

## 步骤四：调用Java SDK收发消息

1. 下载第三方的开源Java SDK。下载地址为[Eclipse Paho Java Client](#)。
2. 下载阿里云微消息队列MQTT版的Java SDK的Demo示例作为您代码开发的参考。下载地址为[mqtt-java-demo](#)。
3. 解压该Demo工程包至您指定的文件夹。
4. 在IntelliJ IDEA中，导入解压后的文件以创建相应的工程，并确认中已包含以下依赖。

```
<dependencies>
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.10</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.2</version>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.48</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-onsmqtt</artifactId>
    <version>1.0.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.0</version>
  </dependency>
</dependencies>
```

5. 在 `MQ4IoTSendMessageToMQ4IoTUseSignatureMode.java` 类中，按代码注释说明填写相应参数，主要涉及步骤一至步骤三所创建的MQTT资源，然后执行Main函数运行代码实现消息收发。示例代码详情请参见 [MQTTSendMessageToMQTT](#)。

## 结果验证

完成消息收发后，您可在微消息队列MQTT版控制台查询轨迹以验证消息是否发送并接收成功。详情请参见 [查询消息轨迹](#)。

## 更多信息



- 快速使用MQTT的Java SDK收发消息（跨产品数据流入）
- 快速使用MQTT的Java SDK收发消息（跨产品数据流出）
- 快速使用MQTT的Java SDK收发消息（客户端上下线通知）

## 2.快速使用MQTT的Java SDK收发消息 （跨产品数据流入）

本文介绍如何将其他阿里云产品的数据导入微消息队列MQTT版。本文以当前仅支持的消息队列RocketMQ版数据互通为例进行说明。

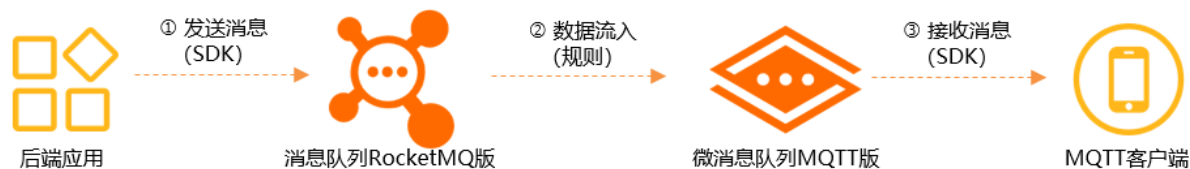
### 前提条件

- 安装IDE。您可以使用IntelliJ IDEA或者Eclipse，本文以IntelliJ IDEA为例。
- 下载安装JDK。
- 在公网地域，创建消息队列RocketMQ版实例、Topic以及Group ID。详情请参见[创建资源](#)。

### 背景信息

本文以公网环境中的Java SDK为例说明如何将消息队列RocketMQ版数据流入至微消息队列MQTT版。

此场景下可使用多语言的第三方开源SDK来实现消息收发。更多信息，请参见[SDK下载](#)。



如上图所示，您部署在公网地域的后端应用数据需要传送到公网环境中的MQTT客户端，后端应用和MQTT客户端均通过Java语言开发。数据从

消息队列RocketMQ版

流入至微消息队列MQTT版是通过创建相应的数据流入规则实现。两个产品的服务端通过各自产品提供的Java SDK分别与各自的客户端实现消息收发。

#### ② 说明

消息队列RocketMQ版

和微消息队列MQTT版的Topic不能跨地域使用，因此，本文中所有资源都应在公网地域创建。详情请参见[Topic地域化](#)。

### 网络访问

微消息队列MQTT版同时提供了公网接入点、经典网接入点和VPC接入点。接入点说明如下：

- 在物联网和移动互联网的场景中，客户端推荐使用公网接入点接入。
- 经典网接入点和VPC接入点仅供一些特殊场景使用。因为一般而言，涉及部署在云端服务器上的应用的

场景，建议使用服务端消息产品例如

消息队列RocketMQ版

实现。

**注意** 客户端使用接入点连接服务时务必使用域名接入，不得直接使用域名背后的IP地址直接连接，因为IP地址随时会变化。在以下使用情况中出现的问题微消息队列MQTT版产品方概不负责：

- 客户端不使用域名接入而是使用IP地址接入，产品方更新了域名解析导致原有IP地址失效。
- 客户端网络对IP地址设置网络防火墙策略，产品方更新了域名解析后新IP地址被您的防火墙策略拦截。

本文以公网接入点为例。微消息队列MQTT版与

消息队列RocketMQ版

的应用场景对比和消息属性映射关系请参见以下文档：

- [MQTT与Rocket的应用场景对比](#)
- [MQTT与Rocket的消息结构映射](#)

## 使用流程

后端应用消息发送至MQTT客户端的使用流程如下所示。



## 步骤一：创建MQTT实例并获取接入点

1. 登录[微消息队列MQTT版控制台](#)，在顶部菜单栏选择实例所在地域。
2. 在概览页面的实例列表区域，[点击这里](#)创建您的第一个实例。
3. 在售卖页面，按需选择您所需的实例规格，单击[立即购买](#)。您可购买包年包月或按量付费实例，两种实例的计费详情请参见[计费说明](#)。此处以按量付费实例为例。



4. 在[确认订单](#)页面，选中[微消息队列for IoT（按量付费）服务协议](#)，单击[立即开通](#)。此处以按量付费

实例为例。



支付完成后，即可看到以下界面，说明已购买成功。



5. 回到[微消息队列MQTT版控制台](#)，在顶部菜单栏将地域切换为您所购买的实例所对应的地域。
6. 在左侧导航栏，选择实例详情 > 实例信息。
7. 在实例信息页签的获取接入点信息 区域，即可看到接入点信息，本文示例以公网接入点为例。

## 步骤二：创建父级Topic

MQTT协议支持多级Topic，父级Topic需在控制台或调用OpenAPI创建，子级Topic无需创建，Topic详情请参见[名词解释](#)。本文以在控制台创建父级Topic为例。

1. 在左侧导航栏，选择Topic 管理 > 创建 Topic。
2. 在创建 Topic对话框，输入Topic以及描述，单击确认。  
您可以在Topic 管理页面查看刚创建的Topic。

## 步骤三：创建Group ID

Group ID详情请参见[名词解释](#)。

1. 在左侧导航栏，选择Group 管理 > 创建 Group ID。
2. 在创建 Group ID对话框，输入Group ID，然后单击确认。  
您可以在Group ID列表中查看到刚创建的Group ID。

## 步骤四：创建数据流入规则

规则中填写的参数需与您创建的资源保持一致。

1. 在左侧导航栏，选择规则管理 > 创建规则。
2. 在创建规则面板，单击数据流入页签。
3. 在数据流入页签，按提示填写以下参数，单击确定。

参数	取值示例	说明
规则ID	111111	规则的全局唯一标识，说明如下： <ul style="list-style-type: none"> <li>只能包含字母、数字、横划线（-）和下划线（_），至少包含一个字母或数字。</li> <li>名称长度限制在3~64字符之间，长于64字符将被自动截取。</li> <li>创建后无法更新。</li> </ul>
规则描述	migrate from rocketmq	对规则的描述。
状态	启用	是否启用当前规则，取值说明如下： <ul style="list-style-type: none"> <li>启用</li> <li>禁用</li> </ul>
数据源 <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <span style="color: #007bff; font-size: 1.2em;">?</span> <b>说明</b> 当前仅支持消息队列RocketMQ版。                     </div>		
云产品	消息队列RocketMQ	指定您需将哪个源云产品的数据流入至微消息队列MQTT版。
实例ID	MQ_INST_13801563067*****_BbyOD2jQ	指定的源云产品的实例ID。
Topic	Topic_test	指定的源云产品的资源键值，即消息队列RocketMQ版的Topic。Topic_test的数据将流转至目标微消息队列MQTT版的Topic。
流转目标		
Topic	TopicA	指定您需将其他源云产品的数据导入至微消息队列MQTT版的哪个目标Topic。


您可以在**规则管理**的规则列表查看到刚创建的数据流入规则。

### 步骤五：调用Java SDK收发消息

1. 下载第三方的开源Java SDK。下载地址为[Eclipse Paho Java Client](#)。
2. 下载阿里云微消息队列MQTT版的Java SDK的Demo示例作为您代码开发的参考。下载地址为[mqtt-java-demo](#)。
3. 解压该Demo工程包至您指定的文件夹。

4. 在IntelliJ IDEA中，导入解压后的文件以创建相应的工程，并确认中已包含以下依赖。

```
<dependencies>
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.10</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.2</version>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.48</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>ons-client</artifactId>
    <version>1.8.5.Final</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-onsmqtt</artifactId>
    <version>1.0.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.0</version>
  </dependency>
</dependencies>
```

 说明 ons-client的版本详情请参见[版本说明](#)。

5. 在 `RocketMQSendMessageToMQ4IoT.java` 类中，按代码注释说明填写相应参数，主要涉及步骤一至步骤三所创建的MQTT资源以及您在

消息队列RocketMQ版

创建的相应资源，然后执行Main函数运行代码实现消息收发。

示例代码详情请参见[RocketMQSendMessageToMQ4IoT.java](#)。

## 结果验证

完成消息收发后，您可在微消息队列MQTT版控制台查询轨迹以验证消息是否发送并接收成功。详情请参见[查询消息轨迹](#)。

## 更多信息

- [快速使用MQTT的Java SDK收发消息（无跨产品数据互通）](#)
- [快速使用MQTT的Java SDK收发消息（跨产品数据流出）](#)
- [快速使用MQTT的Java SDK收发消息（客户端上下线通知）](#)

## 3.快速使用MQTT的Java SDK收发消息 (跨产品数据流出)

本文介绍如何将微消息队列MQTT版的数据导出至其他阿里云产品。本文以当前仅支持的消息队列RocketMQ版数据互通为例进行说明。

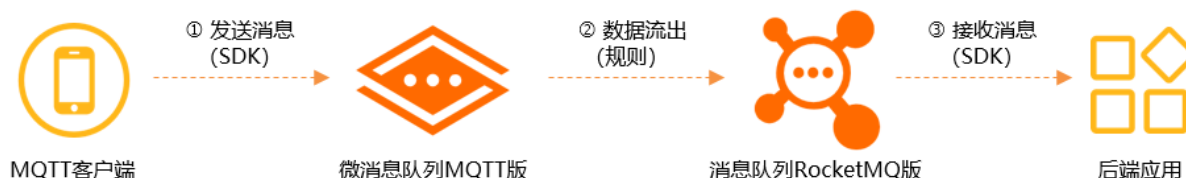
### 前提条件

- 安装IDE。您可以使用IntelliJ IDEA或者Eclipse，本文以IntelliJ IDEA为例。
- 下载安装JDK。
- 在公网地域，创建消息队列RocketMQ版实例、Topic以及Group ID。详情请参见[创建资源](#)。

### 背景信息

本文以公网环境中的Java SDK为例说明如何将微消息队列MQTT版的数据导出至消息队列RocketMQ版。

此场景下可使用多语言的第三方开源SDK来实现消息收发。更多信息，请参见[SDK下载](#)。



如上图所示，您部署在公网环境的MQTT客户端数据需要传送至公网地域的后端应用，后端应用和MQTT客户端均通过Java语言开发。数据从微消息队列MQTT版导出至消息队列RocketMQ版

是通过配置数据流出规则实现。两个产品的服务端通过各自产品提供的Java SDK分别与各自的客户端实现消息收发。

#### ② 说明

消息队列RocketMQ版

和微消息队列MQTT版的Topic不能跨地域使用，因此，本文中所有资源都应在公网地域创建。详情请参见[Topic地域化](#)。

### 网络访问

微消息队列MQTT版同时提供了公网接入点、经典网接入点和VPC接入点。接入点说明如下：

- 在物联网和移动互联网的场景中，客户端推荐使用公网接入点接入。
- 经典网接入点和VPC接入点仅供一些特殊场景使用。因为一般而言，涉及部署在云端服务器上的应用的



场景，建议使用服务端消息产品例如

消息队列RocketMQ版

实现。

**注意** 客户端使用接入点连接服务时务必使用域名接入，不得直接使用域名背后的IP地址直接连接，因为IP地址随时会变化。在以下使用情况中出现的问题微消息队列MQTT版产品方概不负责：

- 客户端不使用域名接入而是使用IP地址接入，产品方更新了域名解析导致原有IP地址失效。
- 客户端网络对IP地址设置网络防火墙策略，产品方更新了域名解析后新IP地址被您的防火墙策略拦截。

本文以公网接入点为例。微消息队列MQTT版与

消息队列RocketMQ版

的应用场景对比和消息属性映射关系请参见以下文档：

- [MQTT与Rocket的应用场景对比](#)
- [MQTT与Rocket的消息结构映射](#)

## 使用流程

MQTT客户端消息发送至后端应用的流程如下图所示。



## 步骤一：创建MQTT实例并获取接入点

1. 登录[微消息队列MQTT版控制台](#)，在顶部菜单栏选择实例所在地域。
2. 在概览页面的实例列表区域，[点击这里](#)创建您的第一个实例。
3. 在售卖页面，按需选择您所需的实例规格，单击[立即购买](#)。您可购买包年包月或按量付费实例，两种实例的计费详情请参见[计费说明](#)。此处以按量付费实例为例。



4. 在确认订单页面，选中[微消息队列for IoT（按量付费）服务协议](#)，单击[立即开通](#)。此处以按量付费

实例为例。



支付完成后，即可看到以下界面，说明已购买成功。



5. 回到[微消息队列MQTT版控制台](#)，在顶部菜单栏将地域切换为您所购买的实例所对应的地域。
6. 在左侧导航栏，选择实例详情 > 实例信息。
7. 在实例信息页签的获取接入点信息 区域，即可看到接入点信息，本文示例以公网接入点为例。

## 步骤二：创建父级Topic

MQTT协议支持多级Topic，父级Topic需在控制台或调用OpenAPI创建，子级Topic无需创建，Topic详情请参见[名词解释](#)。本文以在控制台创建父级Topic为例。

1. 在左侧导航栏，选择Topic 管理 > 创建 Topic。
2. 在创建 Topic对话框，输入Topic以及描述，单击确认。  
您可以在Topic 管理页面查看刚创建的Topic。

## 步骤三：创建Group ID

Group ID详情请参见[名词解释](#)。

1. 在左侧导航栏，选择Group 管理 > 创建 Group ID。
2. 在创建 Group ID对话框，输入Group ID，然后单击确认。  
您可以在Group ID列表中查看到刚创建的Group ID。

## 步骤四：创建数据流出规则

规则中填写的参数需与您创建的资源保持一致。

1. 在左侧导航栏，选择规则管理 > 创建规则。
2. 在创建规则面板，单击数据流出页签。
3. 在数据流出页签，按提示填写以下参数，单击确定。

参数	取值示例	说明
规则ID	111111	规则的全局唯一标识，说明如下： <ul style="list-style-type: none"> <li>只能包含字母、数字、横划线（-）和下划线（_），至少包含一个字母或数字。</li> <li>名称长度限制在3~64字符之间，长于64字符将被自动截取。</li> <li>创建后无法更新。</li> </ul>
规则描述	migrate to rocketmq	对规则的描述。
状态	启用	是否启用当前规则，取值说明如下： <ul style="list-style-type: none"> <li>启用</li> <li>禁用</li> </ul>
数据源		
Topic	TopicA	指定您需导出数据的源Topic。
流转目标		
<div style="background-color: #e6f2ff; padding: 10px;"> <p> <b>说明</b> 当前仅支持消息队列RocketMQ版。</p> <ul style="list-style-type: none"> <li>。</li> </ul> </div>		
云产品	消息队列RocketMQ	指定您需将源Topic的数据转发至的目标云产品。
实例ID	MQ_INST_13801563067*****_BbyOD2jQ	指定的目标云产品的实例ID。
Topic	Topic_test	指定的目标云产品的资源键值，即消息队列RocketMQ版的Topic。源Topic的数据将流转至Topic_test。

您可以在**规则管理**的规则列表查看到刚创建的数据流出规则。

## 步骤五：调用Java SDK收发消息

1. 下载第三方的开源Java SDK。下载地址为[Eclipse Paho Java Client](#)。
2. 下载阿里云微消息队列MQTT版的Java SDK的Demo示例作为您代码开发的参考。下载地址为[mqtt-java-demo](#)。
3. 解压该Demo工程包至您指定的文件夹。
4. 在IntelliJ IDEA中，导入解压后的文件以创建相应的工程，并确认中已包含以下依赖。

```
<dependencies>
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.10</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.2</version>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.48</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>ons-client</artifactId>
    <version>1.8.5.Final</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-onsmqtt</artifactId>
    <version>1.0.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.0</version>
  </dependency>
</dependencies>
```

 说明 ons-client的版本详情请参见[版本说明](#)。

5. 在*MQ4IoTSendMessageToRocketMQ.java*类中，按代码注释说明填写相应参数，主要涉及步骤一至步骤三所创建的MQTT资源以及您在消息队列RocketMQ版创建的相应资源，然后执行Main函数运行代码实现消息收发。代码示例详情请参见[MQ4IoTSendMessageToRocketMQ.java](#)。

## 结果验证

完成消息收发后，您可在微消息队列MQTT版控制台查询轨迹以验证消息是否发送并接收成功。详情请参见[查询消息轨迹](#)。

## 更多信息

- [快速使用MQTT的Java SDK收发消息（无跨产品数据互通）](#)
- [快速使用MQTT的Java SDK收发消息（跨产品数据流入）](#)
- [快速使用MQTT的Java SDK收发消息（客户端上下线通知）](#)

## 4.快速使用MQTT的Java SDK收发消息 (客户端上下线通知)

MQTT客户端的上下线事件将会触发MQTT服务端生成一条通知消息，微消息队列MQTT版支持将该条消息数据导出至其他阿里云产品，并使用MQTT的Java SDK实现MQTT客户端与后端应用收发消息。本文以当前仅支持的

消息队列RocketMQ版

数据互通为例进行说明。

### 前提条件

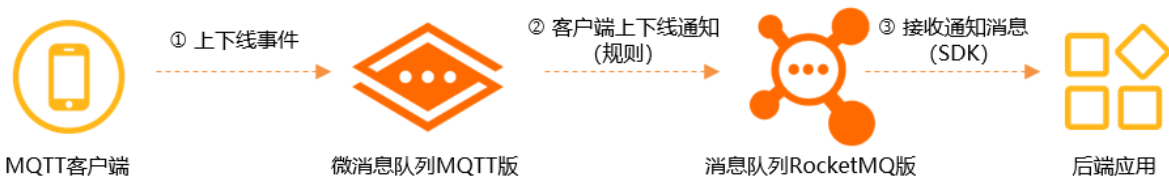
- 安装IDE。您可以使用IntelliJ IDEA或者Eclipse，本文以IntelliJ IDEA为例。
- 下载安装JDK。
- 在公网地域，创建消息队列RocketMQ版实例、Topic以及Group ID。详情请参见[创建资源](#)。

### 背景信息

您可以通过配置客户端上下线通知规则来将MQTT客户端的上下线事件通知的数据导出至其他阿里云产品。

本文以公网环境中的Java SDK为例说明微消息队列MQTT版如何将MQTT客户端上下线事件通知的消息发送至后端应用。

此场景下可使用多语言的第三方开源SDK来实现消息收发。更多信息，请参见[SDK下载](#)。



如上图所示，您的后端应用和MQTT客户端均通过Java语言开发。您部署在公网环境中的MQTT客户端上线或下线时，会触发MQTT服务生成一条事件消息，您可以通过配置客户端上下线通知规则将该通知消息经过消息队列RocketMQ版

发送至后端应用。两个产品的服务端通过各自产品提供的Java SDK分别与各自的客户端实现消息收发。

客户端上下线通知功能的更多信息，请参见[MQTT 客户端上下线事件数据流出](#)。

#### ④ 说明

消息队列RocketMQ版

和微消息队列MQTT版的Topic不能跨地域使用，因此，本文中所有资源都应在公网地域创建。详情请参见[Topic地域化](#)。

### 网络访问

微消息队列MQTT版同时提供了公网接入点、经典网接入点和VPC接入点。接入点说明如下：

- 在物联网和移动互联网的场景中，客户端推荐使用公网接入点接入。

- 经典网接入点和VPC接入点 仅供一些特殊场景使用。因为一般而言，涉及部署在云端服务器上的应用的场景，建议使用服务端消息产品例如

消息队列RocketMQ版  
实现。

**注意** 客户端使用接入点连接服务时务必使用域名接入，不得直接使用域名背后的IP地址直接连接，因为IP地址随时会变化。在以下使用情况中出现的问题微消息队列MQTT版产品方概不负责：

- 客户端不使用域名接入而是使用IP地址接入，产品方更新了域名解析导致原有IP地址失效。
- 客户端网络对IP地址设置网络防火墙策略，产品方更新了域名解析后新IP地址被您的防火墙策略拦截。

本文以公网接入点为例。微消息队列MQTT版与

消息队列RocketMQ版

的应用场景对比和消息属性映射关系请参见以下文档：

- [MQTT与Rocket的应用场景对比](#)
- [MQTT与Rocket的消息结构映射](#)

## 使用流程

MQTT客户端上下线通知的消息收发流程如下图所示。



## 步骤一：创建MQTT实例并获取接入点

1. 登录[微消息队列MQTT版控制台](#)，在顶部菜单栏选择实例所在地域。
2. 在概览页面的实例列表区域，[点击这里](#)创建您的第一个实例。
3. 在售卖页面，按需选择您所需的实例规格，单击[立即购买](#)。您可购买包年包月或按量付费实例，两种实例的计费详情请参见[计费说明](#)。此处以按量付费实例为例。



4. 在**确认订单**页面，选中**微消息队列for IoT（按量付费）**服务协议，单击**立即开通**。此处以按量付费实例为例。



支付完成后，即可看到以下界面，说明已购买成功。



5. 回到**微消息队列MQTT版控制台**，在顶部菜单栏将地域切换为您所购买的实例所对应的地域。
6. 在左侧导航栏，选择**实例详情 > 实例信息**。
7. 在**实例信息**页签的**获取接入点信息**区域，即可看到接入点信息，本文示例以**公网接入点**为例。

## 步骤二：创建父级Topic

MQTT协议支持多级Topic，父级Topic需在控制台或调用OpenAPI创建，子级Topic无需创建，Topic详情请参见**名词解释**。本文以在控制台创建父级Topic为例。

1. 在左侧导航栏，选择**Topic 管理 > 创建 Topic**。
2. 在**创建 Topic**对话框，输入Topic以及描述，单击**确认**。  
您可以在**Topic 管理**页面查看刚创建的Topic。

## 步骤三：创建Group ID

Group ID详情请参见**名词解释**。

1. 在左侧导航栏，选择**Group 管理 > 创建 Group ID**。
2. 在**创建 Group ID**对话框，输入Group ID，然后单击**确认**。  
您可以在Group ID列表中查看到刚创建的Group ID。

## 步骤四：创建客户端上下线通知规则

规则中填写的参数需与您创建的资源保持一致。

1. 在左侧导航栏，选择**规则管理 > 创建规则**。
2. 在**创建规则**面板，单击**上下线通知**页签。
3. 在**上下线通知**页签，按提示填写以下参数，单击**确定**。



参数	取值示例	说明
规则ID	111111	规则的全局唯一标识，说明如下： <ul style="list-style-type: none"> <li>只能包含字母、数字、横划线（-）和下划线（_），至少包含一个字母或数字。</li> <li>名称长度限制在3~64字符之间，长于64字符将被自动截取。</li> <li>创建后无法更新。</li> </ul>
规则描述	migrate from rocketmq	对规则的描述。
状态	启用	是否启用当前规则，取值说明如下： <ul style="list-style-type: none"> <li>启用</li> <li>禁用</li> </ul>
设备组		
Group ID	GID_Client_Status	指定需导出数据的设备组。Group ID详情请参见 <a href="#">名词解释</a> 。
通知 <div style="background-color: #e6f2ff; padding: 10px; margin-top: 10px;"> <p> <b>说明</b> 当前仅支持消息队列RocketMQ版。</p> <ul style="list-style-type: none"> <li>。</li> </ul> </div>		
云产品	消息队列RocketMQ	指定您需将MQTT客户端上下线通知流转至哪个目标云产品，即消息队列RocketMQ版。
实例ID	MQ_INST_13801563067*****_BbyOD2jQ	指定的目标云产品的实例ID，即消息队列RocketMQ版的实例ID。
Topic	Topic_test	指定的目标云产品的资源键值，即消息队列RocketMQ版的Topic。MQTT客户端上下线通知信息将流转至Topic_test。


您可以在[规则管理](#)的规则列表查看到刚创建的上下线通知规则。

### 步骤五：调用Java SDK收发消息

1. 下载第三方的开源Java SDK。下载地址为[Eclipse Paho Java Client](#)。
2. 下载阿里云微消息队列MQTT版的Java SDK的Demo示例作为您代码开发的参考。下载地址为[mqtt-java-demo](#)。
3. 解压该Demo工程包至您指定的文件夹。

4. 在IntelliJ IDEA中，导入解压后的文件以创建相应的工程，并确认中已包含以下依赖。

```
<dependencies>
  <dependency>
    <groupId>commons-codec</groupId>
    <artifactId>commons-codec</artifactId>
    <version>1.10</version>
  </dependency>
  <dependency>
    <groupId>org.eclipse.paho</groupId>
    <artifactId>org.eclipse.paho.client.mqttv3</artifactId>
    <version>1.2.2</version>
  </dependency>
  <dependency>
    <groupId>org.apache.httpcomponents</groupId>
    <artifactId>httpclient</artifactId>
    <version>4.5.2</version>
  </dependency>
  <dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>fastjson</artifactId>
    <version>1.2.48</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun.openservices</groupId>
    <artifactId>ons-client</artifactId>
    <version>1.8.5.Final</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-onsmqtt</artifactId>
    <version>1.0.3</version>
  </dependency>
  <dependency>
    <groupId>com.aliyun</groupId>
    <artifactId>aliyun-java-sdk-core</artifactId>
    <version>4.5.0</version>
  </dependency>
</dependencies>
```

 说明 ons-client的版本详情请参见[版本说明](#)。

5. 在 `MQTTClientStatusNoticeProcessDemo.java` 类中，按代码注释说明填写相应参数，主要涉及步骤一至步骤三所创建的MQTT资源以及您在

消息队列RocketMQ版

创建的相应资源，然后执行Main函数运行代码实现消息收发。

示例代码详情请参见[MQTTClientStatusNoticeProcessDemo.java](#)。

## 结果验证

完成消息收发后，您可在微消息队列MQTT版控制台查询轨迹以验证消息是否发送并接收成功。详情请参见[查询消息轨迹](#)。

## 更多信息

- [快速使用MQTT的Java SDK收发消息（无跨产品数据互通）](#)
- [快速使用MQTT的Java SDK收发消息（跨产品数据流入）](#)
- [快速使用MQTT的Java SDK收发消息（跨产品数据流出）](#)