

# 消息队列 RabbitMQ 版

## 快速入门

## 产品文档



腾讯云

---

**【版权声明】**

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

---

## 文档目录

快速入门

资源创建与准备

使用 SDK 收发消息

# 快速入门

## 资源创建与准备

最近更新时间：2024-01-03 11:39:16

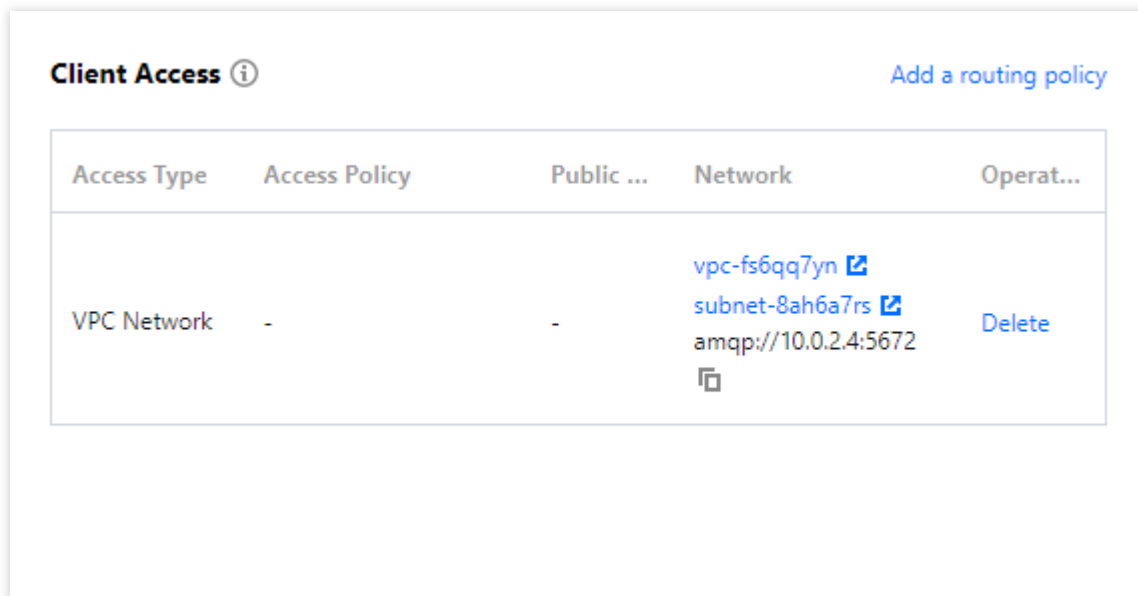
### 操作场景

本文介绍通过 TDMQ 控制台创建一个专享集群，并通过开源 RabbitMQ 控制台创建 Vhost、Exchange 和 Queue 等资源的操作步骤，了解运行一个客户端之前所需要进行的资源准备。

### 操作步骤

#### 步骤1：新建集群

1. 登录 [TDMQ 控制台](#)。
2. 在左侧导航栏选择 **rabbitmq** 下的集群管理，单击**新建集群**，进入购买页面。
3. 在购买页面，选择要购买的实例规格，单击**立即购买**，完成集群创建。
4. 单击集群的“ID”，进入基本信息页面，在**客户端接入**模块，得到服务端的连接信息。



#### 步骤2：新建 Vhost

1. 单击刚刚创建好的集群的“ID”，进入基本信息页面。

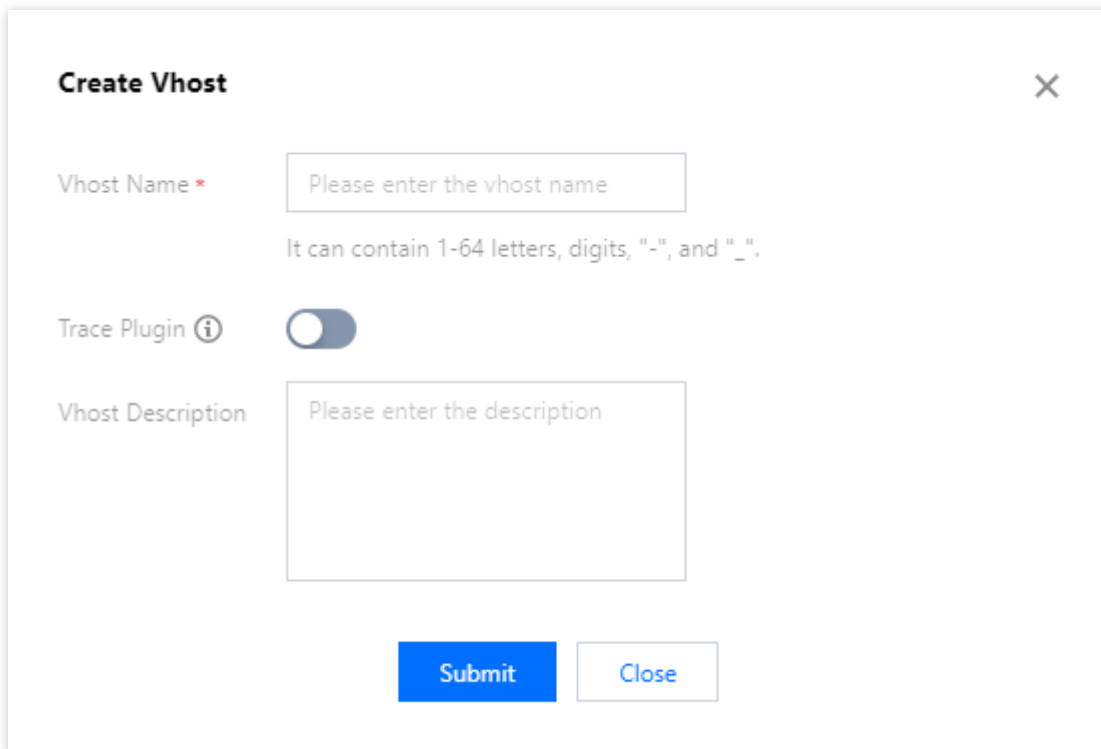
2. 选择顶部 **Vhost** 页签，单击**新建**进入创建 Vhost 页面。

3. 在新建 Vhost 对话框，设置 Vhost 的相关属性配置。

Vhost 名称：设置 Vhost 的名称（创建后不可修改），1-64个字符，只能包含字母、数字、'-'及'\_'

说明：Vhost 的备注说明

4. 单击**提交**完成所在集群 Vhost 的创建。



**Create Vhost** [X]

Vhost Name \*   
It can contain 1-64 letters, digits, "-", and "\_".

Trace Plugin ⓘ

Vhost Description

### 步骤3：创建用户并授权


每个集群下都默认有一个名为“admin”的用户，您可以为这个默认用户配置权限，也可以重新新建用户。


1. 在页面顶部选择用户与权限页签，在用户管理页面，单击**新建用户**。



2. 填写好用户名和密码，单击**提交**，完成用户创建。

### Create User ✕

Username \*   
This field is required. Please enter 1-64 letters, digits, or symbols ("-" or "\_").

Password \*    
This field is required and must contain 8-64 characters in at least two of the following types: lowercase letters, uppercase letters, digits, and symbols ([!@#\$%^&\*\_{}|~`'";,./]).  
**Please keep your password properly and remember it.**

Confirm Password \*    
This field is required and must contain 8-64 characters in at least two of the following types: lowercase letters, uppercase letters, digits, and symbols ([!@#\$%^&\*\_{}|~`'";,./]).

Role    
For permission description for different roles, see [Documentation](#) .

Description

3. 在**用户与权限**页面，选择**权限列表**页签，进入权限列表，单击**配置权限**。

4. 在权限配置页面，选择好需要配置权限的 **Vhost** 和用户，并设置好权限规则。

权限规则设置支持使用**正则表达式**匹配资源。例如，勾选**配置**且在输入框内输入 `test.-*`，则表示授权给用户当前 **Vhost** 下，所有名称以 `test-` 开头的资源的配置权限。

5. 单击**提交**，完成权限配置。

### Configure Permission ✕

Vhost \*

No vhost is available? Please go to the [Vhost](#) tab to create one.

Username \*

Permission ⓘ

<input checked="" type="checkbox"/> Configuration	If this option is selected, the defa
<input checked="" type="checkbox"/> read	If this option is selected, the defa
<input type="checkbox"/> write	If this option is selected, the defa

For more permission type information, see [here](#)

#### 步骤4：创建Exchange

1. 在 Exchange 列表页，选择顶部的 **Exchange** 页签，进入 Exchange 列表页。
2. 选择刚刚创建好的 Vhost，单击**新建**，填写 Exchange 名称，选择路由类型，可选填其他高级参数。
3. 单击**提交**，完成 Exchange 创建。

### Create Exchange ✕

Current Vhost (AMQP default vhost)

Exchange Name \*   
This field is required. Please enter 1-64 letters, digits, or symbols (".", "-", or "\_").

Route Type \*   
For route type descriptions, see [Route Type](#)

Durable   
If this option is enabled, the exchange will still exist after the service is restarted; if it is disabled, the exchange will disappear after the service restart and needs to be created again.

AutoDelete   
If this option is enabled, the exchange will be automatically deleted when the last queue bound to it is deleted.

Internal   
If this option is enabled, this exchange cannot be directly used by producers but bound with other exchanges.

Exchange Description   
Up to 128 characters

Advanced Settings ▶

## 步骤5：创建 Queue

1. 选择顶部的 **Queue** 页签，进入 Queue 列表页。
2. 选择刚刚创建好的 Vhost，单击**新建**，填写 Queue 名称，选择 Queue 类型，所在节点，可选填常用参数和其他高级选项。
3. 在最后一步，单击**提交**，完成 Queue 创建。



### Create Queue ✕

1 Basic Info > 2 Common Parameters > 3 Other Advanced Options

Current Vhost (AMQP default vhost)

Queue Name \*   
This field is required. Please enter 1-64 letters, digits, or symbols (".", "-", or "\_").

Type

Durable

Node

AutoDelete   
The queue will be immediately deleted after the last consumer unsubscribes from it.

Queue Description   
Up to 128 characters

## 步骤6：绑定路由关系

1. 在 Queue 列表页，选择顶部的**路由关系**页签，进入路由关系列表页。
2. 选择刚刚创建好的 Vhost，单击**新建**。  
源 Exchange 选择刚刚创建的 Exchange，填写绑定key，绑定类型选择 Queue，绑定目标选择刚刚创建好的 Queue。
3. 单击**提交**，完成路由关系绑定。

### Create Binding ✕

Current Vhost (AMQP default vhost)

Source Exchange \*

Binding Key \*   
It can only contain 1-255 letters, digits, and symbols (-\_@#\*).

Binding Target Type  Exchange  Queue

Binding Target \*

# 使用 SDK 收发消息

最近更新时间：2024-01-03 11:39:16

## 操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解消息收发的完整过程。

## 前提条件

[完成资源创建与准备](#)

[安装1.8或以上版本 JDK](#)

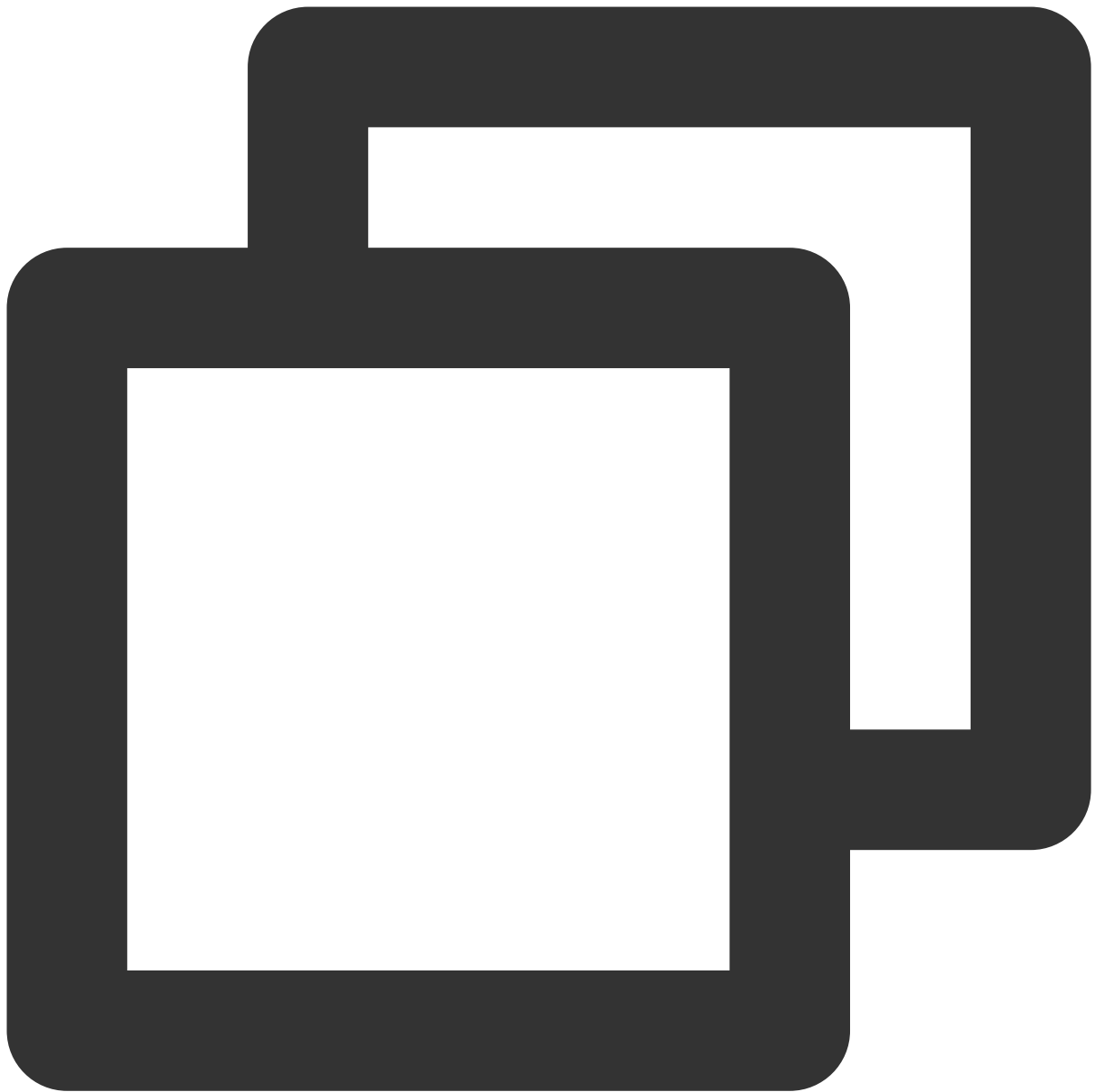
[安装2.5或以上版本 Maven](#)

[下载 Demo](#)

## 操作步骤

### 步骤1：安装 Java 依赖库

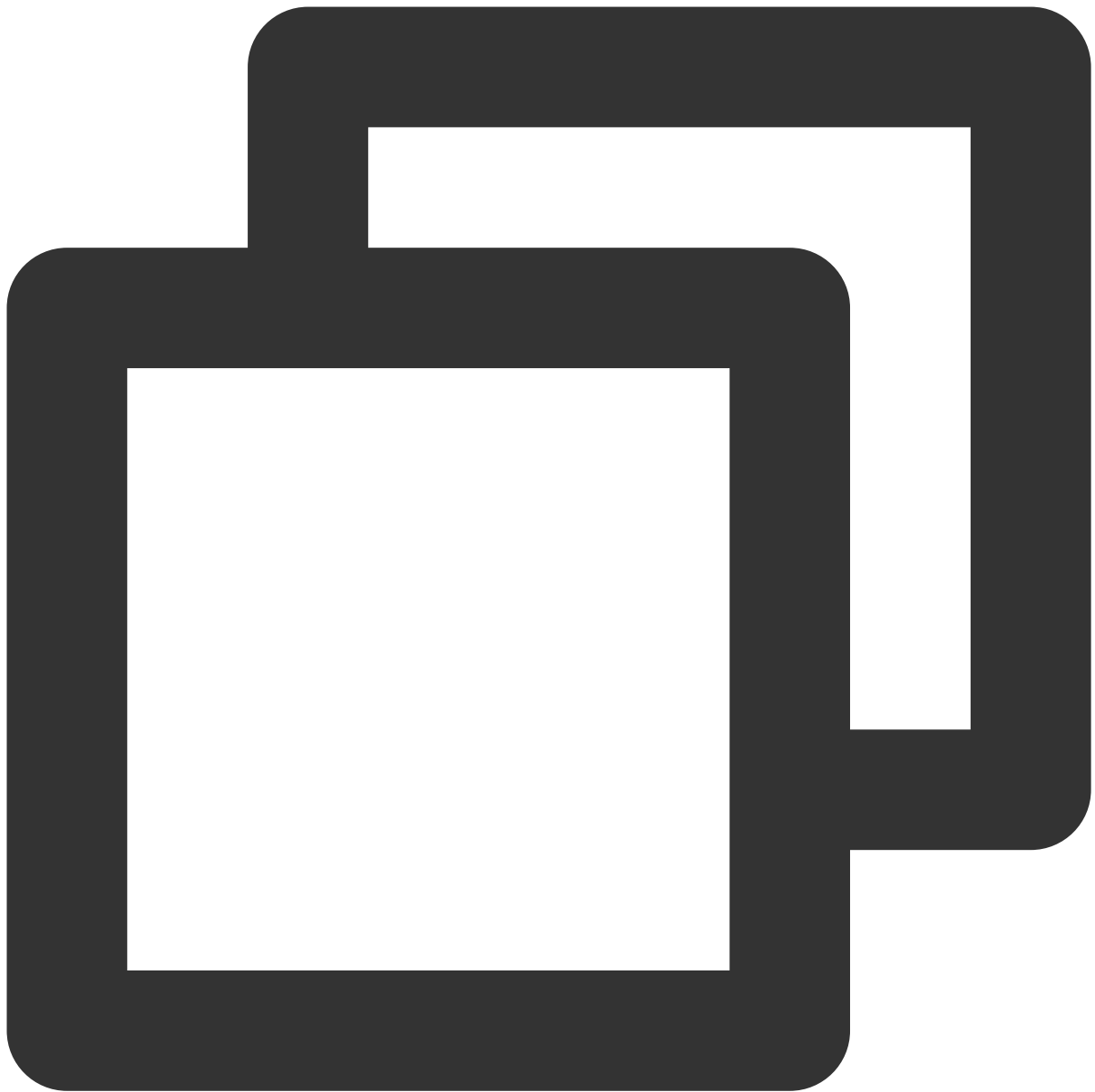
在 pom.xml 添加以下依赖：



```
<!-- in your <dependencies> block -->
<dependency>
  <groupId>com.rabbitmq</groupId>
  <artifactId>amqp-client</artifactId>
  <version>5.13.0</version>
</dependency>
```

## 步骤2：生产消息

编译并运行 MessageProducer.java。



```
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.tencent.tdmq.demo.cloud.Constant;

/**
 * 消息生产者
 */
public class MessageProducer {
```

```

/**
 * 交换机名称
 */
private static final String EXCHANGE_NAME = "exchange_name";

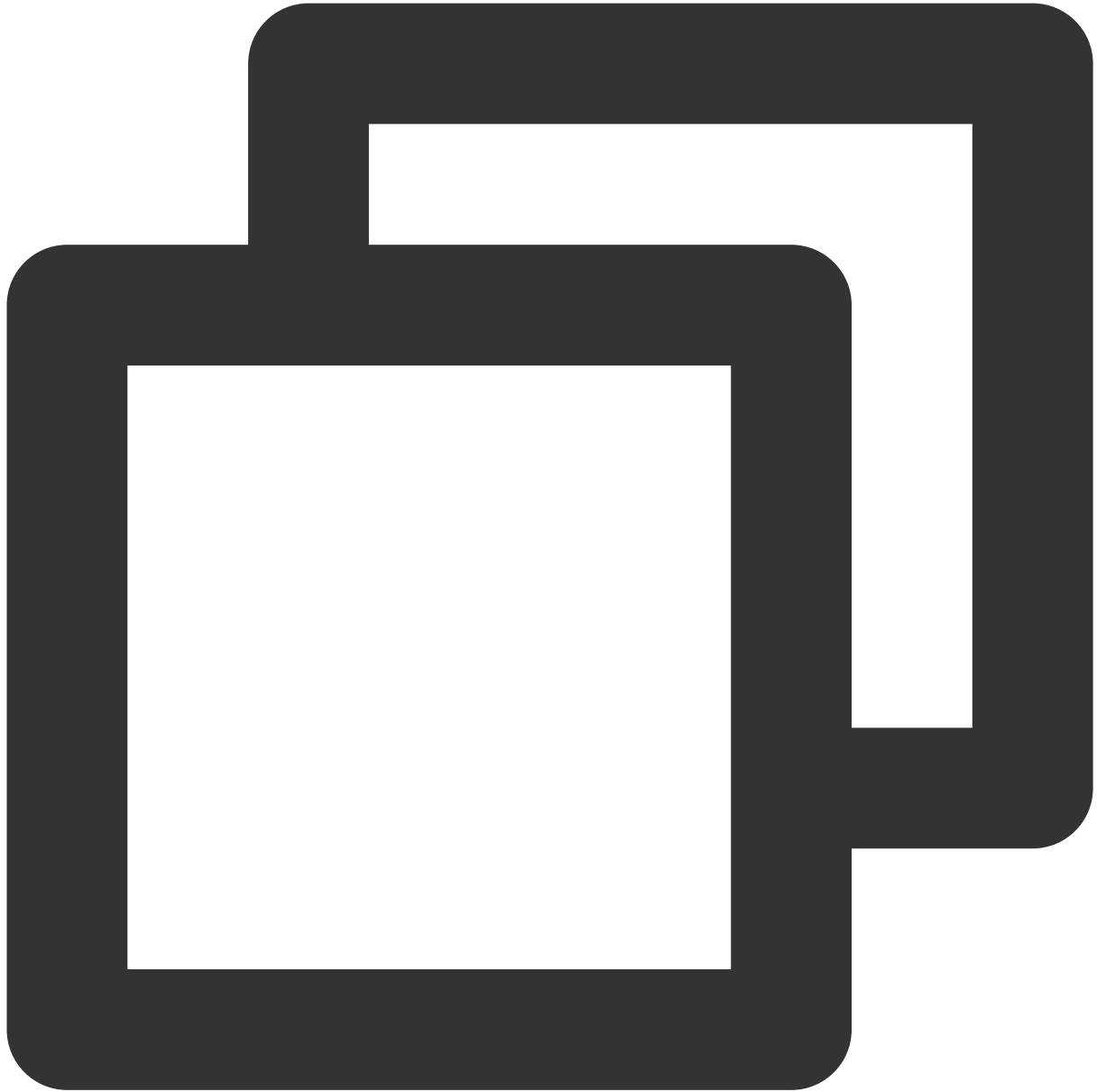
public static void main(String[] args) throws Exception {
    // 连接工厂
    ConnectionFactory factory = new ConnectionFactory();
    // 设置服务地址 (完整复制控制台接入点地址)
    factory.setUri("amqp://***");
    // 设置Virtual Hosts (开源 RabbitMQ 控制台复制完整Vhost名称)
    factory.setVirtualHost(VHOST_NAME);
    // 设置用户名 (开源 RabbitMQ 控制台中Vhost的配置权限中的user名称)
    factory.setUsername(USERNAME);
    // 设置密码 (对应user的密钥)
    factory.setPassword("****");
    // 获取连接、建立通道
    try (Connection connection = factory.newConnection(); Channel channel = conn
        // 绑定消息交换机 (EXCHANGE_NAME必须在消息队列RabbitMQ版控制台上已存在, 并且Exch
        channel.exchangeDeclare(EXCHANGE_NAME, "fanout");
        for (int i = 0; i < 10; i++) {
            String message = "this is rabbitmq message " + i;
            // 发布消息到交换机, 交换机自动将消息投递到相应队列
            channel.basicPublish(EXCHANGE_NAME, "", null, message.getBytes());
            System.out.println(" [producer] Sent '" + message + "'");
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}
}

```

参数	说明
EXCHANGE_NAME	Exchange 名称, 在控制台 Exchange 列表获取。
factory.setUri	集群接入地址, 在集群基本信息页面的 <b>客户端接入</b> 模块获取。
factory.setVirtualHost	Vhost 名称, 在控制台 Vhost 列表获取。
factory.setUsername	用户名称, 填写在控制台创建的用户名称。
factory.setPassword	用户密码, 填写在控制台创建用户时填写的密码。

### 步骤3：消费消息

编译并运行 MessageConsumer.java。



```
import com.rabbitmq.client.AMQP;
import com.rabbitmq.client.Channel;
import com.rabbitmq.client.Connection;
import com.rabbitmq.client.ConnectionFactory;
import com.rabbitmq.client.DefaultConsumer;
import com.rabbitmq.client.Envelope;
import com.tencent.tdmq.demo.cloud.Constant;

import java.io.IOException;
```

```

import java.nio.charset.StandardCharsets;

/**
 * 消息消费者
 */
public class MessageConsumer1 {

    /**
     * 队列名称
     */
    public static final String QUEUE_NAME = "queue_name";

    /**
     * 交换机名称
     */
    private static final String EXCHANGE_NAME = "exchange_name";

    public static void main(String[] args) throws Exception {
        // 连接工厂
        ConnectionFactory factory = new ConnectionFactory();
        // 设置服务地址 (完整复制控制台接入点地址)
        factory.setUri("amqp://***");
        // 设置Virtual Hosts (开源 RabbitMQ 控制台中复制完整vhost名称)
        factory.setVirtualHost(VHOST_NAME);
        // 设置用户名 (开源 RabbitMQ 控制台中vhost的配置权限中的user名称)
        factory.setUsername(USERNAME);
        // 设置密码 (对应user的密钥)
        factory.setPassword("****");
        // 获取连接
        Connection connection = factory.newConnection();
        // 建立通道
        Channel channel = connection.createChannel();
        // 绑定消息交换机
        channel.exchangeDeclare(EXCHANGE_NAME, "fanout");
        // 声明队列信息
        channel.queueDeclare(QUEUE_NAME, true, false, false, null);
        // 绑定消息交换机 (EXCHANGE_NAME必须在消息队列RabbitMQ版控制台上已存在, 并且Exchange)
        channel.queueBind(QUEUE_NAME, EXCHANGE_NAME, "");
        System.out.println(" [Consumer1] Waiting for messages.");
        // 订阅消息
        channel.basicConsume(QUEUE_NAME, false, "ConsumerTag", new DefaultConsumer(channel) {
            @Override
            public void handleDelivery(String consumerTag, Envelope envelope,
                AMQP.BasicProperties properties, byte[] body)
    
```



```

        throws IOException {
        //接收到的消息，进行业务逻辑处理。
        System.out.println("Received: " + new String(body, StandardCharsets
        channel.basicAck(envelope.getDeliveryTag(), false);
    }
    });
}
}

```

参数	说明
QUEUE_NAME	Queue 名称，在控制台 Queue 列表获取。
EXCHANGE_NAME	Exchange 名称，在控制台 Exchange 列表获取。
factory.setUri	集群接入地址，在集群基本信息页面的 <b>客户端接入</b> 模块获取。
factory.setVirtualHost	Vhost 名称，在控制台 Vhost 列表获取。
factory.setUsername	用户名称，填写在控制台创建的用户名称。
factory.setPassword	用户密码，填写在控制台创建用户时填写的密码。