

ALIBABA CLOUD

# 阿里云

## 快速入门

文档版本：20201223

 阿里云

## 法律声明

阿里云提醒您在使用或阅读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 <b>确定</b> 。
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.快速入门概述	05
2.开通消息队列RabbitMQ版	06
3.创建资源	07
4.调用SDK收发消息	11

# 1.快速入门概述

本文旨在以简单明了的方式引导您快速上手消息队列RabbitMQ版。

## 操作流程

1. 开通消息队列RabbitMQ版
2. 创建资源
3. 调用SDK收发消息

## 2. 开通消息队列RabbitMQ版

本文介绍如何开通消息队列RabbitMQ版。

### 前提条件

您已注册阿里云账号并完成实名认证。更多信息，请参见[阿里云账号注册流程](#)。

### 开通消息队列RabbitMQ版

1. 登录[消息队列RabbitMQ版控制台](#)。页面自动跳转至消息队列RabbitMQ版开通页面。
2. 在消息队列RabbitMQ版开通页面，选中服务协议，然后单击立即开通。

### 后续步骤

[创建资源](#)

## 3. 创建资源

一个新的应用接入消息队列RabbitMQ版需要创建实例、用户名密码、Vhost、Exchange、Queue、绑定关系、AccessKey等资源。

### 前提条件

开通消息队列RabbitMQ版

### 创建实例

#### 注意

- 消息队列RabbitMQ版不再支持创建按量付费实例，仅支持创建包年包月实例。已创建的按量付费实例可以继续正常使用。
- 包年包月实例创建后不支持手动释放。包年包月实例到期后自动释放，届时您可以删除该包年包月实例。

1. 登录[消息队列RabbitMQ版控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击**概览**。
4. 在我的资源页面，单击**增加实例**。
5. 在消息队列RabbitMQ包年包月页面，完成基本配置，选择购买时长，然后单击**立即购买**。

消息队列 RabbitMQ 包年包月

规格类型 **专业版** 铂金版  
 专业版：高可用，共享，逻辑隔离，数据多备份，多节点支持  
 铂金版：专享，物理隔离，多节点，多备份，高级特性支持

地域和可用区 **中国**  
 华东1 (深圳) 华北1 (青岛) 华北2 (北京) 华东2 (上海) 华北3 (呼和浩特) 华北3 (张家口) 华东1 (杭州) 中国 (香港)

公网支持 **不支持公网** 支持公网

TPS流量峰值

1000条/秒	1500条/秒	2000条/秒	2500条/秒	3000条/秒	4000条/秒	5000条/秒	6000条/秒
8000条/秒	1万条/秒	1.5万条/秒	2万条/秒	2.5万条/秒	3万条/秒	3.5万条/秒	4万条/秒
4.5万条/秒	5万条/秒	5.5万条/秒	6万条/秒	6.5万条/秒	7万条/秒	7.5万条/秒	8万条/秒
8.5万条/秒	9万条/秒	9.5万条/秒	10万条/秒				

Queue数量

购买数量

购买时长 **1个月** 2个月 3个月 6个月 1年 2年 **更多时长**

到期自动续费

包配置费用

**立即购买** **加入购物车**

6. 在确认订单页面，选中我已阅读并同意消息队列AMQP包年包月服务协议，然后单击去支付。
7. 在支付页面，单击**订购**。

### 创建用户名密码

1. 登录[消息队列RabbitMQ版控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击**用户名密码管理**。
4. 在用户名密码管理页面，单击**创建用户名密码**。

5. 在创建用户名密码对话框的AccessKey ID文本框，输入AccessKey ID；在AccessKey Secret 文本框，输入AccessKey Secret；在实例ID文本框，输入实例ID，然后单击确定。

## 创建Vhost

1. 登录消息队列RabbitMQ版控制台。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击Vhost详情。
4. 在Vhost详情页面，选择实例，单击创建Vhost。
5. 在创建Vhost对话框的Vhost名称文本框，输入Vhost名称，然后单击创建。

### 说明

- Vhost名称只能包含字母、数字、短划线 (-)、下划线 (\_)、英文句号 (.)、at符号 (@)。
- Vhost名称长度限制在1~255字符，长度超过255字符将被自动截取。
- Vhost创建成功后，Vhost名称不可修改。

## 创建Exchange

1. 登录消息队列RabbitMQ版控制台。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击Exchange管理。
4. 在Exchange管理页面，选择实例，选择Vhost，单击创建Exchange。
5. 在创建Exchange对话框，输入Exchange名称，选择Type，选择Internal，然后单击创建。

参数	描述
Exchange Name	Exchange名称。以amq开头的为保留字段，因此不能使用。例如：amq.test。
Type	Exchange类型。取值： <ul style="list-style-type: none"><li>○ fanout：该类型路由规则非常简单，会把所有发送到该Exchange的消息路由到所有与它绑定的Queue中，相当于广播功能。</li><li>○ direct：该类型路由规则会将消息路由到Binding Key与Routing Key完全匹配的Queue中。</li><li>○ topic：该类型与direct类型相似，只是规则没有那么严格，可以模糊匹配和多条件匹配，即该类型Exchange使用Routing Key模式匹配和字符串比较的方式将消息路由至绑定的Queue。</li></ul>
Internal	是否为Internal类型。默认值为false。取值： <ul style="list-style-type: none"><li>○ true：内建类型，用于Exchange和Exchange之间的绑定。</li><li>○ false：非内建类型，用于Exchange和Queue之间的绑定。</li></ul>



## 创建Queue

1. 登录消息队列RabbitMQ版控制台。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击Queue管理。
4. 在Queue管理页面，选择实例，选择Vhost，单击创建Queue。
5. 在创建Queue对话框，输入Queue的名称，选择是否为Auto Delete类型，设置Arguments，然后单击确定。

创建Queue
✕

**\* Queue名称**  4/255

1. Queue名称只能包含字母、数字、短划线 (-)、下划线 (\_)、英文句号 (.)、at符号 (@)，长度限制1~255字符。  
2. Queue一旦创建，名称无法再修改，只能删除后重建。

**Auto Delete**

**Arguments:** ⓘ + 新增

确定
取消

参数	描述	说明
Queue	Queue名称	<ul style="list-style-type: none"> <li>◦ 只能包含字母、数字、短划线 (-)、下划线 (_)、英文句号 (.)、at符号 (@)。</li> <li>◦ 长度限制在1~255字符。</li> <li>◦ 创建后无法修改，只能删除重建。</li> <li>◦ 以amq开头的为保留字段，因此不能使用。例如：amq.test。</li> </ul>
Auto Delete	最后一个Consumer取消订阅后，Queue是否自动删除。	<ul style="list-style-type: none"> <li>◦ true：在订阅该Queue消息的最后一个Consumer取消订阅该Queue的消息后，自动删除该Queue。</li> <li>◦ false：在订阅该Queue消息的最后一个Consumer取消订阅该Queue的消息后，不自动删除该Queue。</li> </ul>

参数	描述	说明
Arguments	Queue的参数设置，可用于设置死信Exchange、死信Routing Key和消息过期时间。	<ul style="list-style-type: none"> <li>◦ DeadLetterExchange: 指定死信消息发往的Exchange。</li> <li>◦ DeadLetterRoutingKey: 指定死信消息的Routing Key，即死信Exchange会将消息发往至匹配该死信Routing Key的Binding Key所对应的Queue。</li> <li>◦ MessageTTL: 消息过期时间，单位毫秒（ms）。超过指定时间段内还未被消费的消息是死信消息，该消息将会被发往死信Exchange。</li> </ul>

## 创建绑定关系

1. 登录[消息队列RabbitMQ版控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击[Queue管理](#)。
4. 在Queue管理页面，选择实例，选择Vhost，在对应Queue的右侧操作列，单击[绑定关系](#)。
5. 在Queue被绑定情况对话框右侧，单击[添加绑定](#)，在源Exchange对应的文本框中，选择Exchange；在Binding key对应的文本框中，输入Binding Key，然后单击[确定](#)。

## 创建AccessKey

在调用SDK发送或订阅消息时，除了需要指定实例、Vhost、Exchange、Queue以外，还需输入您在RAM控制台创建的AccessKey。AccessKey的信息包含AccessKey ID和AccessKey Secret。详情请参见[创建AccessKey](#)。

## 后续步骤

[调用SDK收发消息](#)

## 4.调用SDK收发消息

本文以Java SDK为例说明如何调用SDK收发消息。更多语言或框架的SDK，请参见[SDK概述](#)。

### 前提条件

[创建资源](#)

### 背景信息

客户端接入消息队列RabbitMQ版时，消息队列RabbitMQ版会通过用户名密码进行权限认证。消息队列RabbitMQ版支持通过控制台为客户端生成静态用户名密码，该方式和开源RabbitMQ保持一致。

**注意** 您的客户端在调用SDK收发消息时，请尽可能使用长期存活的Connection，以免每次收发消息时都需要创建新的Connection，消耗大量的网络资源和服务端资源，甚至引起服务端SYN Flood防护。更多信息，请参见[Connection](#)。

### 收发消息流程



### 获取接入点

您需要在消息队列RabbitMQ版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，以此接入消息队列RabbitMQ版实例。

1. 登录[消息队列RabbitMQ版控制台](#)。
2. 在顶部菜单栏，选择地域。
3. 在左侧导航栏，单击[实例详情](#)。
4. 在[实例详情](#)页面，选择实例，在[基本信息](#)区域，将鼠标指针移动到目标类型的接入点，然后单击该接入点以复制该接入点。

类型	说明	示例值
公网接入点	公网环境可读写。按量付费实例默认支持，预付费实例需在购买时选择才支持。	XXX.mq-amqp.cn-hangzhou-a.aliyuncs.com
VPC接入点	VPC环境可读写。按量付费实例和预付费实例默认都支持。	XXX.mq-amqp.cn-hangzhou-a-internal.aliyuncs.com

### 安装Java依赖库

在pom.xml中添加以下依赖。

```
<dependency>
  <groupId>com.rabbitmq</groupId>
  <artifactId>amqp-client</artifactId>
  <version>5.5.0</version>
</dependency>
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.6</version>
</dependency>
```

## 生产消息

创建并编译运行ProducerTest.java。

```
import com.rabbitmq.client.*;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.concurrent.TimeoutException;
import java.util.UUID;
public class ProducerTest {
    public static void main(String[] args) throws IOException, TimeoutException {
        ConnectionFactory factory = new ConnectionFactory();
        // 设置接入点，在RabbitMQ版控制台实例详情页面查看。
        factory.setHost("xxx.xxx.aliyuncs.com");
        // 用户名，在RabbitMQ版控制台用户名密码管理页面查看。
        factory.setUsername("${UserName}");
        // 密码，在RabbitMQ版控制台用户名密码管理页面查看。
        factory.setPassword("${PassWord}");
        //一定要这个才能自动恢复。
        factory.setAutomaticRecoveryEnabled(true);
        factory.setNetworkRecoveryInterval(5000);
        // 设置Vhost名称，请确保已在RabbitMQ版控制台上创建完成。
        factory.setVirtualHost("${VhostName}");
        // 默认端口，非加密端口5672，加密端口5671。
        factory.setPort(5672);
        // 基于网络环境合理设置超时时间。
        factory.setConnectionTimeout(30 * 1000);
        factory.setHandshakeTimeout(30 * 1000);
        factory.setShutdownTimeout(0);
        Connection connection = factory.newConnection();
        Channel channel = connection.createChannel();
        // 开始发送消息。
        for (int i = 0; i < 100; i++ ) {
            // ${ExchangeName}必须在RabbitMQ版控制台上已存在，并且Exchange的类型与控制台上的类型一致。
            // BindingKey根据业务需求填入相应的BindingKey。
            AMQP.BasicProperties props = new AMQP.BasicProperties.Builder().messageId(UUID.randomUUID().toString()).build();
            channel.basicPublish("${ExchangeName}", "BindingKey", true, props,
                ("消息发送Body" + i).getBytes(StandardCharsets.UTF_8));
        }
        connection.close();
    }
}
```

## 消费消息

创建并编译运行ConsumerTest.java。

```
import com.rabbitmq.client.*;
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.HashMap;
import java.util.concurrent.TimeoutException;
public class ConsumerTest {
    public static void main(String[] args) throws IOException, TimeoutException {
        ConnectionFactory factory = new ConnectionFactory();
        // 设置接入点，在RabbitMQ版控制台实例详情页面查看。
        factory.setHost("xxx.xxx.aliyuncs.com");
        // 用户名，在RabbitMQ版控制台用户名密码管理页面查看。
        factory.setUsername("${Username}");
        // 密码，在RabbitMQ版控制台用户名密码管理页面查看。
        factory.setPassword("${Password}");
        //一定要这个才能自动恢复。
        factory.setAutomaticRecoveryEnabled(true);
        factory.setNetworkRecoveryInterval(5000);
        // 设置Vhost名称，请确保已在RabbitMQ版控制台上创建完成。
        factory.setVirtualHost("${VhostName}");
        // 默认端口，非加密端口5672，加密端口5671。
        factory.setPort(5672);
        factory.setConnectionTimeout(300 * 1000);
        factory.setHandshakeTimeout(300 * 1000);
        factory.setShutdownTimeout(0);
        Connection connection = factory.newConnection();
        final Channel channel = connection.createChannel();
        // 开始消费消息。
        channel.basicConsume("${QueueName}", 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.UTF_8) + ", deliveryTag: " +
                    envelope.getDeliveryTag() + ", messageId: " + properties.getMessageId());
                channel.basicAck(envelope.getDeliveryTag(), false);
            }
        });
    }
};
```

```
connection.close();  
}  
}
```

可参考[消息日志](#)查看消息的更多消费信息。