

消息队列RabbitMQ版 快速入门

文档版本: 20220221



ALIBABA CLOUD

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。

- 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。
- 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。
- 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。		
▲ 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。	大意 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 <i>,</i> 不是 用户必须了解的内容。	⑦ 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。	
粗体	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 cd /d C:/window 命令,进入 Windows系统文件夹。	
斜体	表示参数、变量。	bae log listinstanceid	
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]	
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}	

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	12

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Rabbit MQ版。

操作流程

- 1. 开通消息队列Rabbit MQ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Rabbit MQ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,请参见注册阿里云账号。

开通消息队列RabbitMQ版

1. 登录消息队列Rabbit MQ版控制台。

页面自动跳转至消息队列Rabbit MQ版开通页面。

2. 在消息队列Rabbit MQ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Rabbit MQ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Rabbit MQ版

创建实例

- ◯ 注意
 - 消息队列Rabbit MQ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Rabbit MQ版控制台。
- 2. 在左侧导航栏,单击实例列表。
- 3. 在实例列表页面,单击创建实例。
- 4. 购买实例,请选择如下任意一种方式购买:
 - a. 在购买面板,完成基本配置,选择购买时长并勾选消息队列 Rabbit MQ 版包年包月服务协议,然后单击立即购买。
 - b. 在订单支付面板,确认订单信息,单击**支付**。
 - a. 单击前往购买页,在消息队列 Rabbit MQ 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。
 - b. 在确认订单页面,选中我已阅读并同意消息队列Rabbit MQ包年包月服务协议,然后单击去支付。
 - c. 在订单支付页面, 单击**支付**。

创建用户名密码

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在实例列表页面,单击目标实例名称。
- 4. 在左侧导航栏,单击静态用户名密码。
- 5. 在静态用户名密码页面,单击创建用户名密码。
- 6. 在创建用户名密码面板,输入AccessKey ID,输入AccessKey Secret,单击确定。 静态用户名密码页面,显示创建的静态用户名与密码,密码处于隐藏状态。

7. 在创建的静态用户名密码的密码列,单击显示密码,可查看用户名的密码。

创建Vhost

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在实例列表页面,单击目标实例名称。
- 4. 在左侧导航栏,单击Vhost 列表。
- 5. 在Vhost 列表页面,单击创建 Vhost。
- 6. 在创建 Vhost 面板的Vhost 名称文本框,输入Vhost 名称,然后单击确定。

? 说明

您设置Vhost名称时,请注意有如下要求:

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

创建Exchange

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在实例列表页面,单击目标实例名称。
- 4. 在左侧导航栏,单击Exchange 列表。
- 5. 在Exchange 列表页面,在当前 Vhost 右侧的切换下拉列表中,选择Vhost,然后单击创建 Exchange。
- 在创建 Exchange面板,输入Exchange名称,选择Exchange类型,设置是否为Internal类型,然后单击确定。

参数	描述
Exchange 名称	Exchange名称。以amq.开头的为保留字段,因此不能 使用。例如:amq.test。

参数	描述
类型	 Exchange类型。取值: 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。 topic:该类型与direct类型相似。Topic Exchange路由规则没有Direct Exchange那么严格,支持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中。 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。 headers:该类型与direct类型相似。Headers Exchange使用Headers属性代替Routing Key进行路由匹配,在绑定Headers Exchange和Queue时,设置绑定属性的键值对;在向Headers Exchange发送消息时,设置消息的Headers属性键值对比较的方式将消息路由至绑定的Queue。 x-jms-topic:适用于通过消息队列RabbitMQ版提供的JMS接口接入消息队列RabbitMQ版提供的JMS接口接入消息队列RabbitMQ版版提供的JMS成用,该类型路由规则会将消息路由到Binding Key与Routing Key通配符匹配的Queue中。更多信息,请参见JMS概述。 x-delayed-message:通过声明该类Exchange,您可以自定义消息的Header属性x-delay来指定消息延时投递的时间段,单位为毫秒。消息将在x-delay中定义的时间段后,根据路由规则 减投递到对应的Queue。路由规则取决于x-
x-delayed-type	当Exchange类型为 x-delayed-message 时,需要 配置此参数,以指定Exchange的路由类型。
Internal	是否为Internal类型,默认值为否。取值: • 是:内建类型,用于Exchange和Exchange之间的 绑定。 • 否:非内建类型,用于Exchange和Queue之间的绑 定。

创建Queue

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在实例列表页面,单击目标实例名称。
- 4. 在左侧导航栏,单击Queue 列表。
- 5. 在Queue 列表页面,在当前 Vhost 右侧的切换下拉列表中,选择 Vhost,单击创建 Queue。
- 6. 在创建 Queue面板, 在Queue 名称文本框输入Queue的名称, 选择是否为Auto Delete类型, 单

击高级选项,设置Queue的参数,然后单击确定。

配置参数

参数	描述	说明
Queue 名称	Queue的名称	 只能包含字母、数字、短划线 (-)、下划线(_)、半角句号 (.)、井号(#)、正斜线 (/)、at符号(@)。 长度限制在1~255字符。 创建后无法修改,只能删除重 建。 以amq.开头的为保留字段,因 此不能使用。例如: amq.test。
Auto Delete	最后一个Consumer取消订阅 后,Queue是否自动删除。	 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。 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。
高级选项	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。	 DeadLetterExchange:指定死信消息发往的Exchange。 DeadLetterRoutingKey:指定死信消息的RoutingKey,即死信Exchange会将消息发往至匹配该死信RoutingKey的BindingKey所对应的Queue。 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。

创建绑定关系

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在**实例列表**页面,单击目标实例名称。
- 4. 在左侧导航栏,单击Queue 列表。
- 5. 在Queue 列表页面, 在当前 Vhost 右侧的切换下拉列表中, 选择Vhost, 找要被绑定的Queue, 在其 右侧操作列, 选择更多 > 被绑定信息。
- 6. 在Queue 详情页面的被绑定信息页签,单击添加被绑定。

7. 在添加被绑定面板,选择源Exchange,在Binding Key文本框输入Binding Key,然后单击确定。

? 说明

您设置Binding Key时,请注意有以下要求:

- 不支持中文的Binding Key。
- 普通类型的Binding Key只能包含字母、数字、下划线(_)、短划线(-)、半角句号
 (.)、正斜线(/)、at符号(@),长度限制在1~255字节之间。
- Topic类型的Binding Key只能包含字母、数字、下划线(_)、短划线(-)、半角句号
 (.)、井号(#)、正斜线(/)、at符号(@)、星号(*),长度限制在1~255字符之
 间,支持以井号(#)开头或结尾。

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AcessKey Secret。具体操作,请参见获取 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。

前提条件

创建资源

背景信息

客户端接入消息队列Rabbit MQ版时,消息队列Rabbit MQ版会通过用户名密码进行权限认证。消息队列 Rabbit MQ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Rabbit MQ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(以Java语言为例)



获取接入点

您需要在消息队列Rabbit MQ版控制台获取实例的接入点。在收发消息时,您需要为发布端和订阅端配置该 接入点,通过接入点接入消息队列Rabbit MQ版实例。

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在**实例列表**页面,单击目标实例名称。
- 4. 在**实例详情**页面的接入点信息页签,将鼠标指针移动到目标类型的接入点,单击该接入点右侧的

标,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添加以下依赖。

<dependency>

```
<proupId>com.rabbitmq</proupId>
<artifactId>amqp-client</artifactId>
<version>5.5.0</version> <!-- 支持开源所有版本 -->
</dependency>
```

生成用户名密码

- 1. 登录消息队列Rabbit MQ版控制台。
- 2. 在概览页面的资源分布区域,选择地域。
- 3. 在实例列表页面,单击目标实例名称。
- 4. 在左侧导航栏,单击静态用户名密码。
- 5. 在静态用户名密码页面,单击创建用户名密码。
- 6. 在**创建用户名密码**面板,输入AccessKey ID,输入AccessKey Secret,单击**确定**。 静态用户名密码页面,显示创建的静态用户名与密码,密码处于隐藏状态。

宾例 ID	Access Key	用户名	密码	创建时间	操作
and the second se	Consequences of the Consequences	spine and an end of the second	****** 显示密码	今天11:29:38	删除

7. 在创建的静态用户名密码的密码列,单击显示密码,可查看用户名的密码。

生产消息

创建并编译运行ProducerTest.java。

↓ 注意 编译运行 ProducerTest.java生产消息之前,您需要根据代码提示信息配置参数列表中所列举的参数。

参数列表

参数	示例值	描述
hostName	1880770****.mq-amqp.cn- hangzhou-a.aliyuncs.com	消息队列RabbitMQ版实例接入点。 获取方式,请参见。
Port	5672	默认端口。非加密端口为5672 <i>,</i> 加 密端口为5671。
userName	MjoxODgwNzcwODY5MD****	在消息队列RabbitMQ版控制台将阿 里云账号或RAM用户的AccessKey ID、AccessKey Secret和消息队列 RabbitMQ版实例ID通过Base64编码 后生成的静态用户名。您可以在消息 队列RabbitMQ版控制台的静态用户 名密码页面获取。

参数	示例值	描述
passWord	NDAxREVDQzI2MjA0OT****	在 <mark>消息队列RabbitMQ版控制台</mark> 将阿 里云账号或RAM用户的AccessKey Secret和timestamp参数(系统当 前时间)通过HMAC-SHA1生成一个 签名后,再将这个签名 和timestamp参数(系统当前时 间)通过Base64编码后生成的静态 密码。您可以在消息队列RabbitMQ 版控制台的静态用户名密码获取。
virtualHost	Test	消息队列RabbitMQ版实例的 Vhost。您可以在 <u>消息队列</u> RabbitMQ版控制台的Vhost 详 情页面查看。如何查看Vhost,请参 见查看Vhost详情。
ExchangeName	ExchangeTest	消息队列Rabbit MQ版的Exchange。 您可以在 <mark>消息队列Rabbit MQ版控制</mark> <mark>台的Exchange 列表</mark> 页面,结合实 例ID与Vhost模糊搜索已创建的 Exchange。
BindingKey	Binding Key Test	消息队列Rabbit MQ版Exchange与 Queue的Binding Key。您可以在 <mark>消 息队列Rabbit MQ版控制</mark> 台的Exchange 列表页面查看 Exchange的绑定关系,获取Binding Key。
QueueName	QueueTest	消息队列Rabbit MQ版的Queue。仅 在订阅消息时候需要配置,您可以 在 <mark>消息队列Rabbit MQ版控制</mark> 合的Exchange 列表页面,查看 Exchange的绑定关系,获取 Exchange鄉定的Queue。

import com.rabbitmq.client.*; import java.io.IOException; import java.nio.charset.StandardCharsets; import java.util.concurrent.TimeoutException; import java.util.HashMap; 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,在消息队列RabbitMQ版控制台实例详情页面查看。 factory.setHost("xxx.xxx.aliyuncs.com"); // 用户名,在消息队列RabbitMQ版控制台静态用户名密码页面查看。 factory.setUsername("\${UserName}"); // 密码,在消息队列RabbitMQ版控制台静态用户名密码页面查看。 factory.setPassword("\${PassWord}"); //设置为true,开启Connection自动恢复功能;设置为false,关闭Connection自动恢复功能。 factory.setAutomaticRecoveryEnabled(true); factory.setNetworkRecoveryInterval(5000); // 设置Vhost名称,请确保已在消息队列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(); channel.exchangeDeclare("\${ExchangeName}", "\${ExchangeType}", true, false, false, n ull); channel.queueDeclare("\${QueueName}", true, false, false, new HashMap<String, Object >()); channel.queueBind("\${QueueName}", "\${ExchangeName}", "\${BindingKey}"); // 开始发送消息。 for (int i = 0; i < 100; i++) { // \${ExchangeName}必须在消息队列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(); } }

订阅消息

创建并编译运行ConsumerTest.java。

↓ 注意 编译运行 ConsumerTest.java订阅消息之前,您需要根据代码提示信息配置参数列表中所列举的参数。

```
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,在消息队列RabbitMQ版控制台实例详情页面查看。
       factory.setHost("xxx.xxx.aliyuncs.com");
       // 用户名,在消息队列RabbitMQ版控制台静态用户名密码页面查看。
       factory.setUsername("${Username}");
       // 密码,在消息队列RabbitMQ版控制台静态用户名密码页面查看。
       factory.setPassword("${Password}");
       //设置为true,开启Connection自动恢复功能;设置为false,关闭Connection自动恢复功能。
       factory.setAutomaticRecoveryEnabled(true);
       factory.setNetworkRecoveryInterval(5000);
       // 设置Vhost名称,请确保已在消息队列RabbitMQ版控制台上创建完成。
       factory.setVirtualHost("${VhostName}");
       // 默认端口,非加密端口5672,加密端口5671。
       factory.setPort(5672);
       factory.setConnectionTimeout(300 * 1000);
       factory.setHandshakeTimeout(300 * 1000);
       factory.setShutdownTimeout(0);
       Connection connection = factory.newConnection();
       final Channel channel = connection.createChannel();
       // 创建${ExchangeName},该Exchange必须在消息队列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。
       AMQP.Exchange.DeclareOk exchangeDeclareOk = channel.exchangeDeclare("${ExchangeName
}", "${ExchangeType}", true, false, false, null);
       // 创建${QueueName} ,该Queue必须在消息队列RabbitMQ版控制台上已存在。
       AMQP.Queue.DeclareOk queueDeclareOk = channel.queueDeclare("${QueueName}", true, fa
lse, false, new HashMap<String, Object>());
       // Queue与Exchange进行绑定,并确认绑定的BindingKeyTest。
       AMQP.Queue.BindOk bindOk = channel.queueBind("${QueueName}", "${ExchangeName}", "Bi
ndingKeyTest");
       // 开始消费消息。
       channel.basicConsume("${QueueName}", false, "ConsumerTag", new DefaultConsumer(chan
nel) {
           QOverride
           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.getMes
sageId());
              channel.basicAck(envelope.getDeliveryTag(), false);
           }
       });
   }
}
```

⑦ 说明 消息队列Rabbit MQ版与开源Rabbit MQ完全兼容。更多参数说明,请参见开源Rabbit MQ客户 端文档。