

# RabbitMQ 文档



# 目录

目录	2
产品描述	3
介绍	3
核心概念	3
优势	4
应用场景	5
操作指南	6
开通服务	6
创建RabbitMQ实例	6
多用户访问控制	8
产品定价	12
产品定价	12
余额不足和欠费提醒	13
API参考	13
API简介	13
API调用须知	13
计费相关接口	17
对象说明	21
附录	23
典型实践	23
访问实例	23

# 产品描述

## 介绍

百度消息服务 for RabbitMQ是百度智能云基于开源 RabbitMQ 推出的全托管的消息队列产品，提供完全兼容AMQP（Advanced Message Queuing Protocol 高级消息队列协议）协议、稳定可靠、高性能的消息队列服务。RabbitMQ采用全托管模式，用户即开即用，免去部署、维护的工作量和费用，最大化提升业务效率。

## 核心概念

本文主要对消息服务RabbitMQ涉及的专有名词和术语进行定义和解释，方便您更好地理解相关概念并使用该产品。

### RabbitMQ

RabbitMQ是一款用Erlang语言实现，完全支持AMQP协议的开源分布式消息队列系统，具备强大的分布式、可扩展和高可用能力。RabbitMQ在TCP协议层面提供了丰富的多语言SDK，便于用户使用。

### AMQP（高级消息队列协议）

AMQP（Advanced Message Queue Protocol）是一个提供统一消息服务的应用层标准高级消息队列协议，是应用层协议的一个开放标准，为面向消息的中间件设计。基于此协议的客户端与消息中间件可传递消息，不受客户端/中间件不同产品，不同的开发语言等条件的限制。

### 消息（Message）

消息队列服务对外服务的基本实体单元，消息由Header（标签）和Body（消息体）组成。Header是消息生产者添加的各种属性的集合，包括消息是否被持久化，由哪个队列接受，优先级是多少等等。Body是消息体，是真正需要被传输的数据。

### 队列（Queue）

消息的容器，存储被生产者投递过来但尚未被消费的消息。生产者把消息发送到队列，消费者从队列中获取并消费消息。每个消息会被投放到一个或多个队列里。

### 生产者（Producer）

发送消息的应用，消息数据的来源。

### 消费者（Consumer）

接收消息的应用，消息数据的去处。

### Exchange（交换机）

用于决定消息与队列之间投递关系的机制，本质上为消息路由。路由的基础是Routing Key（路由键）和Binding Key（绑定键）。生产者投递消息时，在Header中指定消息的Routing Key。每个队列被指定一个或多个Key（通配符），并将这些Key与Exchange绑定，称为Binding Key。Exchange通过Routing Key与Binding Key的匹配规则来决定消息应该投递给哪个/哪些队列。具体而言，Exchange支持如下几种类型：

- Fanout（扇出）：忽略Routing Key（认为匹配所有的Binding Key），将消息传递给Exchange所绑定的所有队列。
- Direct（直接/点对点）：精确匹配Routing Key与Binding Key，只将消息投递给能够匹配的队列。
- Topic（主题/分发-订阅）：模糊匹配Routing Key与Binding Key，其中“#”通配任意0个或多个单词，“\*”通配任意1个单词。
- Headers：该类型的Exchange不依赖于Routing Key与Binding Key的匹配规则来路由消息，而是根据发送的消息内容中的Headers属性进行匹配。
- RPC：MQ本身是基于异步的消息处理，生产者将消息投递到消息代理之后不关心也不会知道消费者后续处理成功或者失败。但在一些应用场景中，很可能需要同步处理机制，生产者需要同步等待服务端将消息处理完成后再进行下一步处理。

## Broker Server (代理服务器)

消息代理的服务端程序。Exchange机制、队列都位于其中。从生产者客户端连接中接受消息，存储消息，将消息路由到相应的队列，并将队列中的消息发送给消费者客户端。

## Vhost (虚拟主机)

不同的用户通过使用不同的虚拟主机，能够做到权限的隔离。在虚拟主机内，用户可以创建Exchange，Queue等。

## Connection (连接)

代理服务器与客户端之间建立的TCP连接，代理服务器保证不会主动关闭连接，除非出现网络错误或者服务出现问题。

## Channel (通道)

Connection的逻辑概念，只有通过Channel才能执行AMQP命令。业务程序通过Channel与Broker Server交互，Channel之间互不影响。建议客户端线程之间不公用Channel，或者至少要保证多个线程在一个Channel上发送消息的行为是串行的。

## Command (命令)

AMQP的命令。客户端通过Command执行与AMQP服务器的交互。例如在RabbitMQ中，客户端可以通过publish命令发送消息，通过txSelect开启一个事务，通过txCommit提交一个事务。

## 优先级队列

RabbitMQ支持按照消息优先级的高低来消费消息的队列。生产者在发送消息时，需要设置该消息的优先级，RabbitMQ按照队列中优先级高低的顺序来执行消费。

## 死信队列

消息在消费失败、TTL过期、队列长度超限之后会被重新发送到一个特殊的Exchange，即dead letter exchange (DLX)，进入到DLX的消息不会被重新发送到原来的队列重新消费。

## 延时队列

采用延时队列插件 (rabbitmq-delayed-message-exchange) 来实现延时队列功能 (延时队列本身依赖18.0及以上版本的Erlang/OTP)。

# 优势

## 高可用

调度服务实时监控RabbitMQ服务状态，并提供RabbitMQ服务的创建、查询等服务。一个调度服务故障后可以自动切换，不影响对外提供服务，保证服务的高可用。

## 高可靠

- 服务隔离：RabbitMQ服务创建在用户的VPC内，网络上是一个逻辑隔离区，保证不同RabbitMQ实例数据的安全性。
- 租户隔离：RabbitMQ的各租户间资源强隔离，互不影响，保证数据的安全性和稳定性。
- 权限控制：多用户访问控制提供权限管理功能，不同用户具有不同权限，实现细粒度权限控制，对每次消息请求进行有效的安全访问控制。

## 易用性

- 快速部署分钟级创建：RabbitMQ无需考虑后端部署工作，只需在控制界面选择需要创建的实例类型及规格，后台将自动创建部署完成整套RabbitMQ实例。
- 图形化界面：面向用户的Web控制台可帮助用户便捷、高效地进行资源管理，对RabbitMQ实例进行详情查看及相关操作。
- 自动化监控报警：提供队列数量、连接数量、消息发布速度等实例信息，实时监控及报警，随时随地了解实例动态。

## 高效运维

提供实例级别的数据采集监控及报警策略配置，7\*24小时实时监控在线连接、消费情况、消息堆积等方面指标，帮助您快速发现问题并处理问题。

## 高级功能

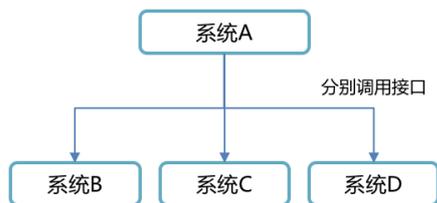
- 优先级队列：RabbitMQ支持按照消息优先级的高低来消费消息的队列。
- 死信队列：RabbitMQ支持消费失败消息、TTL过期消息、队列长度超限消息进入死信队列，且不重新发送到原来的队列消费，保证消息不丢失。
- 延时队列：RabbitMQ支持消息延时/定时发送，满足用户多个场景下的需求。

## 应用场景

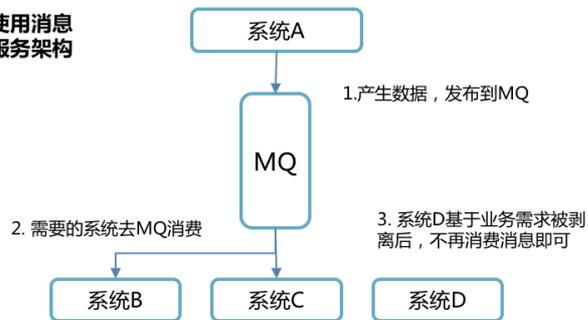
RabbitMQ作为一款热门的消息队列中间件，具备高效可靠的消息异步传递机制，主要用于不同系统间的数据交流和传递，在企业解决方案、金融支付、电信、电子商务、社交、即时通信、视频、物联网、车联网等众多领域都有广泛应用。

### 业务系统解耦

#### 传统架构



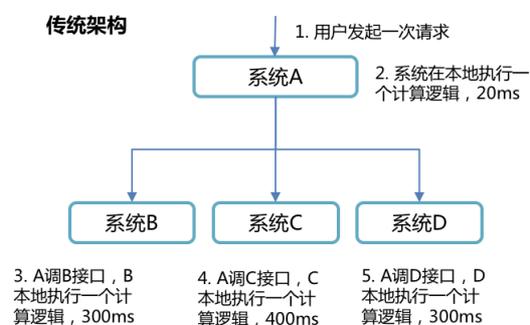
#### 使用消息服务架构



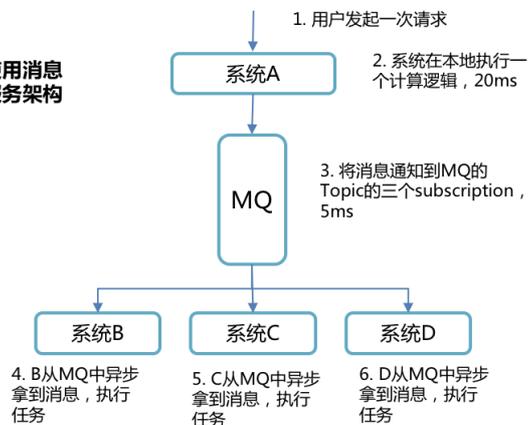
- 传统的事务处理中，多个系统之间的交互会耦合到一个大事务中，响应时间长，降低了系统的整体可用性。引入消息队列服务，将大事务进行上、下游拆分，上游核心交易系统和消息队列服务之间组成一个小事务；下游业务系统（购物车、积分等）相互异步化并完成解耦，从而实现最佳的架构设计，处理效率高，可用性强。
- 适用的领域：基于云搭建的复杂SaaS业务系统、各类互联网应用及游戏等

### 业务异步化

#### 传统架构



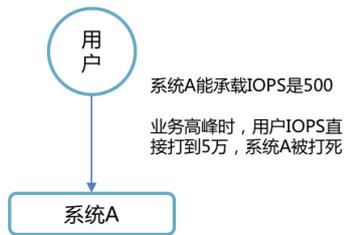
#### 使用消息服务架构



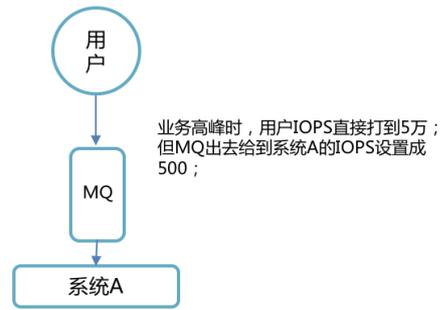
- 在上述的场景中，传统架构中一个任务返回的延时由1020ms降低到25ms，使用消息服务架构后将业务异步化，缩短任务处理时间、提升用户体验。
- 适用的领域：基在各类BS和CS架构的业务系统中几乎都适用，比如web网站、APP等。

### 业务突发削峰

## 传统架构



## 使用消息服务架构



- 在类似电商秒杀、春节活动等业务突发时，传统架构下系统的压力即为用户的请求压力，使用消息队列架构变化队列长度对用户的请求压力实现削峰，有效保障业务系统不被打死，业务可持续。
- 适用的领域：电商、新闻门户等存在周期性尖峰的业务。

## 操作指南

## 开通服务

控制台为用户提供了界面化管理RabbitMQ服务的能力，用户第一次使用百度智能云消息服务 for RabbitMQ产品时，首先需要开通服务。

在开通页面中，点击“开通服务”按钮，即可开通消息服务 for RabbitMQ产品服务。



## 创建RabbitMQ实例

使用百度智能云消息服务 for RabbitMQ 产品，需要先创建实例。

## 1. 登录百度智能云官网

登录[百度智能云官网](#)。

- 若您未注册账号，须先[注册账户](#)。
- 若您已注册，可直接[登录](#)。

## 2. 创建 RabbitMQ 实例

进入RabbitMQ服务管理控制台点击“创建实例”，进入“创建实例”页面。



当前每个用户可在单个区域创建50个实例，如配额用尽请手动删除。

### 3. 选择实例配置信息

#### 基本信息

填写需要创建实例的基本信息，包括实例名称、用户名、密码等。

- 实例名称：设置实例名称；
- 用户名：设置连接RabbitMQ实例的用户名
- 密码：设置连接RabbitMQ实例的密码
- 描述：填写实例的备注信息

#### 基本信息

实例名称

支持大小写字母、数字、中文以及\_./等特殊字符，必须以字母开头，不超过65个字符

\* 用户名:

由英文字母、数字、下划线组成，长度为4~64的字符，且不为root

\* 密码:

8-32位字符，英文，数字和符号必须同时存在，符号仅限!#\$%^&\*()\_

描述:

0/1024

#### 付费及地域

- 付费方式：消息服务 for RabbitMQ 支持预付费及后付费两种付费方式。
- 当前地域：地域为实例所属的地理位置，不同地域内的实例内网不互通，实例创建成功后无法更换地域，请谨慎选择。
- 可用区：百度智能云区域建设其他相关信息，请参见[区域选择说明](#)。

#### 付费及地域

付费方式:

当前地域:  !

可用区:  ▾ !

#### 配置

配置项说明如下：

配置项	说明
消息引擎	目前仅支持3.7版本的RabbitMQ服务
架构类型	支持单机版和集群版两种架构类型，用户可根据业务需求自行选择
实例规格	1) 单机版实例用户可选4种实例规格（CPU核数和内存大小），分别为：2核4GB、4核8GB、8核16GB、16核32GB； 2) 集群版实例用户可选3种实例规格（CPU核数和内存大小），分别为：4核8GB、8核16GB、16核32GB，可选3种规格节点数量，分别为3、5、7。
消息存储磁盘	用户可根据预估的实例存储空间选择磁盘（通用型SSD云磁盘）的存储容量，可选范围为100GB – 2000GB
网络类型	百度智能云为每个用户在每个地域提供了一个默认私有网络VPC和子网，用户也可选择自定义的VPC和子网
SSL	用户可根据业务需求开启SSL传输加密，开启后RabbitMQ实例数据将加密传输，安全性更高，但性能会有一定程度的下降；开启SSL加密后，用户可选是否开启客户端认证，若开启即提供双向验证模式，数据安全性更高。

### 配置

消息引擎: RabbitMQ 3.7

架构类型: 单机版 集群版

实例规格: 2核4G 4核8G 8核16G 16核32G

消息存储磁盘: 通用型SSD云磁盘 100GB 1185GB 2000GB 100 GB

网络类型: 默认私有网络 (192.168.0.0/...) 系统预定义子网 (192.168.0....)

SSL:  关  开  
开启/关闭SSL，创建完成实例后不支持更改

#### 4. 确认订单

请在页面下方确认所选配置及价格，点击"下一步"跳转至【确认订单】页面，进行付款。

支付完成后，若显示信息为“开通成功”，表示完成消息服务 for RabbitMQ 的创建，用户将收到消息服务 for RabbitMQ 实例创建成功的短信提示。

## 多用户访问控制

### 简介

多用户访问控制，主要用于帮助用户管理云账户下资源的访问权限，适用于企业内的不同角色，可以对不同的工作人员赋予使用产品的不同权限，当您的企业存在多用户协同操作资源时，推荐您使用多用户访问控制。

适用于下列使用场景：

- 中大型企业客户：对公司内多个员工授权管理；
- 偏技术型vendor或SAAS的平台商：对代理客户进行资源和权限管理；
- 中小开发者或小企业：添加项目成员或协作者，进行资源管理。

多用户访问控制包含以下几个流程：

1. 创建子用户
2. 配置策略

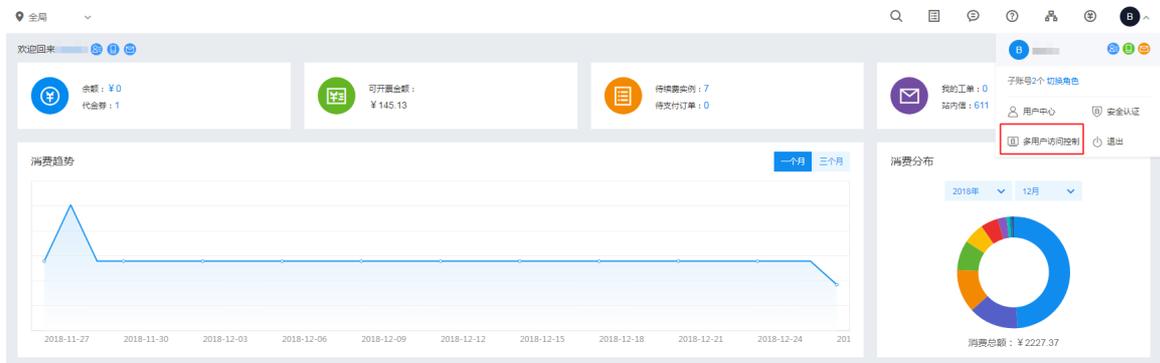
3. 为子用户授权系统权限

4. 子用户登录控制台

详细的操作步骤请参见[多用户访问控制一般操作步骤](#)

## 创建子用户

1. 主账号用户登录后在控制台选择“多用户访问控制”进入用户管理页面。



2. 在左侧导航栏点击“用户管理”，在“子用户管理列表”页，点击“新建用户”。

3. 在弹出的“新建用户”对话框中，完成填写“用户名”和确认，返回“子用户管理列表”区可以查看到刚刚创建的子用户。

## 权限策略配置

权限策略表示的是“允许/拒绝对某资源执行某种操作”，通过将策略授予用户可以使用户拥有相应权限。目前百度智能云支持系统策略和自定义策略两种权限策略，两者主要区别在于资源的配置。

- 系统策略对主用户的所有消息服务 for RabbitMQ 资源生效，用户不能随意修改。
- 自定义策略中，用户可以指定任一实例级别的资源，实现更细粒度的资源配置。

### 1. 操作权限说明

目前消息服务 for RabbitMQ 提供三种操作类型，对应权限如下表所示：

操作类型	实例操作权限
管理操作	<ul style="list-style-type: none"> <li>- 创建实例</li> <li>- 重启实例</li> <li>- 删除实例</li> <li>- 修改实例信息</li> <li>- 重置实例密码</li> <li>- 获取当前用户实例列表</li> <li>- 查看实例详情/监控</li> <li>- 添加白名单</li> <li>- 删除白名单</li> <li>- 查询白名单</li> </ul>
运维操作	<ul style="list-style-type: none"> <li>- 重启实例</li> <li>- 修改实例信息</li> <li>- 重置实例密码</li> <li>- 获取当前用户实例列表</li> <li>- 查看实例详情/监控</li> <li>- 添加白名单</li> <li>- 删除白名单</li> <li>- 查询白名单</li> </ul>
只读操作	<ul style="list-style-type: none"> <li>- 获取当前用户实例列表</li> <li>- 查看实例详情/监控</li> <li>- 查询白名单</li> </ul>

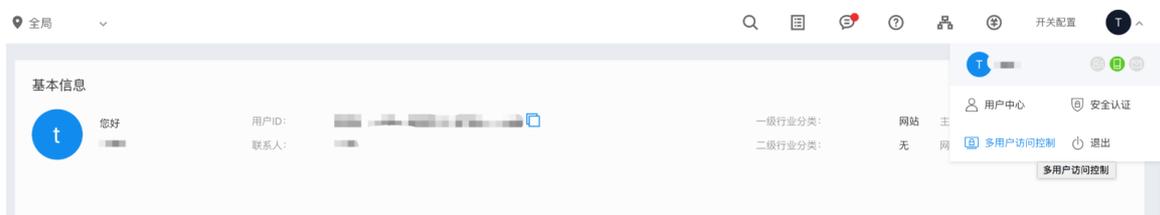
## 2. 系统策略

系统策略是消息服务 for RabbitMQ 定义好的策略，可以指定对主用户拥有的全部实例的操作权限。RabbitMQ 目前提供了三种系统策略，分别对应三种操作类型：

策略名称	操作类型
RABBITMQFullControlPolicy	管理操作
RABBITMQOperatePolicy	运维操作
RABBITMQReadPolicy	只读操作

消息服务 for RabbitMQ 的配置查看方法如下：

1. 进入“管理控制台”，选择“多用户访问控制”。



2. 选择“策略管理”，搜索RABBITMQ，可以看到消息服务 for RabbitMQ 相关策略。



### 3. 自定义策略

自定义策略由主用户自己定义，可以精确指定对主用户拥有的任一实例的操作权限。目前IAM支持2种方式创建自定义策略：按策略生成器创建和按标签创建。按策略生成器创建的方式，通过选择服务、权限以及区域下实例，以生成策略；按标签创建，根据你为服务实例创建的标签筛选资源，生成策略。

配置自定义策略的详细操作步骤请参见[自定义策略操作步骤](#)。

### 子用户授权

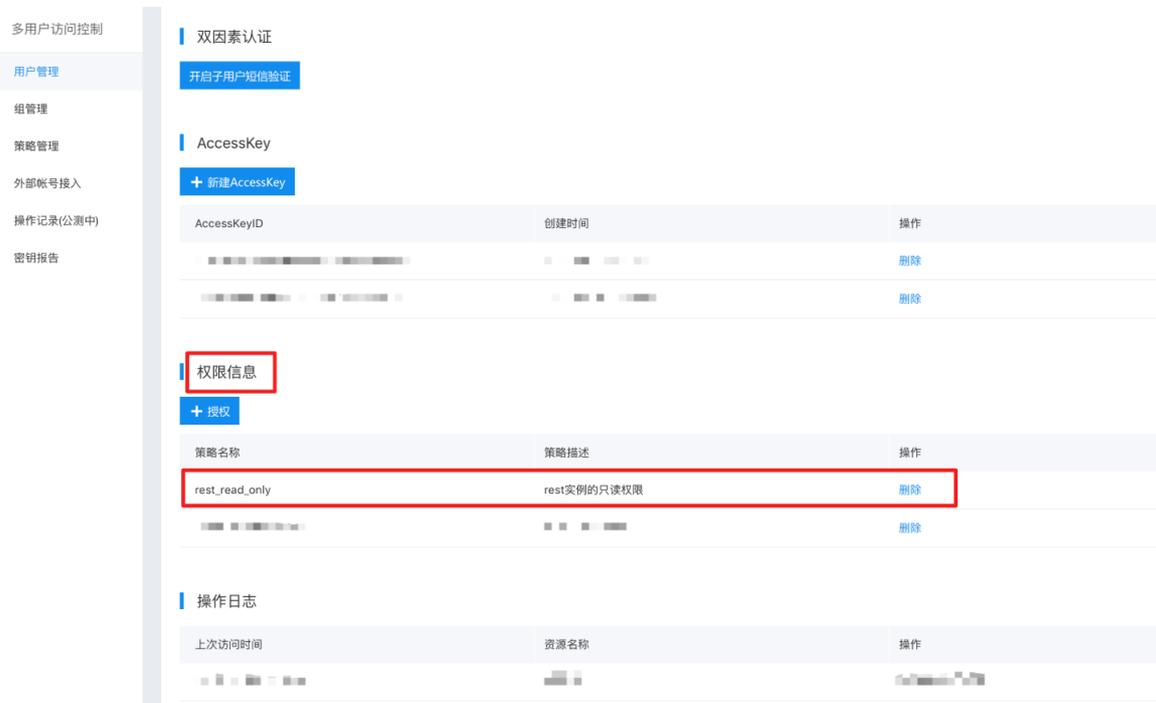
1. 在“用户管理”页签为子用户添加权限。



2. 筛选“全部策略”、“系统策略”或“自定义策略”，获取相应策略列表，选择希望授予子用户的权限，点击“确定”即可将权限赋予子用户。



3. 删除子用户授权：点击用户名，进入“子用户详情页面”，“权限信息”处可以看到此用户拥有的权限，点击“删除”可以将此权限从用户权限中去掉。



### 子用户访问

主账号完成对子用户的授权后，可以将链接发送给子用户；子用户可以通过IAM用户登录链接登录主账号的管理控制台，根据被授权的策略对主账号资源进行操作和查看。



## 产品定价

### 产品定价

#### 计费概述

RabbitMQ以实例为单位进行计价，在创建实例时显示实例对应配置的价格，目前支持单机版及集群版，按需计费详情请参考：[RabbitMQ价格详情](#)。

#### 计费规则

实例价格 = (实例规格价格 + 磁盘存储价格) \* 时长

具体计费规则如下：

- 按分钟计费，不足1分钟按1分钟计；
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 购买前需保证账户无欠款，且保证账户余额和可用代金券总和大于或等于100元。

## 余额不足和欠费提醒

### 余额不足提醒

- 根据您最近3天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来3天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近1天的账单金额来判断您的账户余额（含可用代金券）是否足以支付未来1天的费用，若不足以支付，系统发送续费提醒。

### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次RabbitMQ账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后立即停服，系统会发送欠费停服通知。

## API参考

### API简介

#### 概述

欢迎使用百度智能云消息服务 for RabbitMQ，您可以使用API对RabbitMQ做灵活的操作。如果您对RabbitMQ还不太了解，请参考[核心概念](#)和[应用场景](#)。

如果您是初次调用百度智能云产品的API，可以观看[API入门视频指南](#)，快速掌握调用API的方法。

消息队列服务RabbitMQ API提供下列接口类型。

接口类型	描述
实例管理	实例Instance是一个消息队列服务RabbitMQ实例，通过实例管理API提供的接口可以进行的操作有： 创建实例、查询指定实例详情、查询实例列表、修改实例信息、重置实例密码、删除指定实例、批量删除实例、重启实例、添加白名单、删除白名单、查询白名单、申请公网访问连接地址、释放公网访问连接地址、获取实例证书包

#### API服务域名

百度智能云消息队列服务RabbitMQ的服务域名参见下表：

Region	EndPoint	Protocol
北京	rabbitmq.bj.baidubce.com	HTTP and HTTPS
广州	rabbitmq.gz.baidubce.com	HTTP and HTTPS
苏州	rabbitmq.su.baidubce.com	HTTP and HTTPS
保定	rabbitmq.bd.baidubce.com	HTTP and HTTPS

### API调用须知

#### API认证机制

用户与百度智能云进行交互时，需要通过使用Access Key Id / Secret Access Key加密的方法来验证某个请求的发送者身份。Access Key Id (AK) 用于标示用户，Secret Access Key (SK) 是用户用于加密认证字符串和百度智能云用来验证认证字符串的密钥，其中SK必须保密，只有用户和百度智能云知道。

当百度智能云接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。鉴权认证机制的详细内容请参见[鉴权认证机制](#)。

## 🔗 排版约定

排版格式	含义
< >	变量
[]	可选项
{}	必选项
等宽字体Courier New	屏幕输出

## 🔗 时间与日期

日期与时间的表示有多种方式。为统一起见，除非是约定俗成或者有相应规范的，凡需要日期时间表示的地方一律采用UTC时间，遵循ISO 8601，并做以下约束：

- 表示日期一律采用YYYY-MM-DD方式，例如2016-06-01表示2016年6月1日。
- 表示时间一律采用hh:mm:ss方式，并在最后加一个大写字母Z表示UTC时间。例如23:00:10Z表示UTC时间23点0分10秒。
- 凡涉及日期和时间合并表示时，在两者中间加大写字母T，例如2016-06-01T23:00:10Z表示UTC时间2016年6月1日23点0分10秒。

## 🔗 通用说明

API调用遵循HTTP/HTTPS协议，各Region采用不同的域名，具体域名为rabbitmq.{region}.baidubce.com。数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

## 🔗 请求参数

请求参数包括如下4种：

参数类型	说明
URI	通常用于指明操作实体，如：POST /v{version}/instance/{instanceId}
Query	URL中携带的请求参数
HEADER	通过HTTP头域传入，如：x-bce-date
RequestBody	通过JSON格式组织的请求数据体

## 🔗 返回值

返回值分为两种形式：

返回内容	说明
HTTP STATUS CODE	如200,400,403,404等
RequestBody	JSON格式组织的响应数据体

## 🔗 API版本号

参数	类型	参数位置	描述	是否必须
version	string	URL参数	API版本号，当前API版本为v1	是

## 🔗 认证机制

所有API的安全认证一律采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串。对于每个HTTP请求，使用下面所描述的算法生成一个认证字符串。提交认证字符串放在Authorization头域里。服务端根据生成算法验证认证字符串的正确性。认证字符串的格式为：

```
bce-auth-v{version}/{accessKeyId}/{timestamp}/{expirationPeriodInSeconds}/{signedHeaders}/{signature}
```

#### 说明

- version是正整数。
- timestamp是生成签名时的UTC时间。
- expirationPeriodInSeconds表示签名有效期限。
- signedHeaders是签名算法中涉及到的头域列表。头域名之间用分号（;）分隔，如host;x-bce-date。列表按照字典序排列。（本API签名仅使用host和x-bce-date两个header）
- signature是256位签名的十六进制表示，由64个小写字母组成。

#### 🔗 密码加密传输规范定义

所有涉及密码的接口参数都需要加密，禁止明文传输。密码一律采用AES 128位加密算法进行加密，用SK的前16位作为密钥，加密后生成的二进制字节流需要转成十六进制，并以字符串的形式传到服务端。具体步骤如下：

- byte[] bCiphertext= AES(明文,SK)
- String strHex = HexStr(bCiphertext)

#### 🔗 幂等性

当调用创建接口时如果遇到了请求超时或服务器内部错误，用户可能会尝试重发请求，这时用户通过clientToken参数避免创建出比预期要多的资源，即保证请求的幂等性。

幂等性基于clientToken，clientToken是一个长度不超过64位的ASCII字符串，通常放在query string里，如<http://rabbitmq.bj.baidubce.com/v1/instance?clientToken=be31b98c-5e41-4838-9830-9be700de5a20>。

如果用户使用同一个clientToken值调用创建接口，则服务端会返回相同的请求结果。因此用户在遇到错误进行重试的时候，可以通过提供相同的clientToken值，来确保只创建一个资源；如果用户提供了一个已经使用过的clientToken，但其他请求参数（包括queryString和requestBody）不同甚至url Path不同，则会返回IdempotentParameterMismatch的错误代码。

clientToken的有效期为24小时，以服务端最后一次收到该clientToken为准。也就是说，如果客户端不断发送同一个clientToken，那么该clientToken将长期有效。

#### 🔗 错误返回

##### 🔗 错误返回格式

请求发生错误时通过response body返回详细错误信息，遵循如下格式。

参数名	类型	说明
requestId	String	本次请求的requestId
code	String	错误码
message	String	错误描述

示例：

```
{
  "requestId": "ae2225f7-1c2e-427a-a1ad-5413b762957d",
  "code": "NoSuchKey",
  "message": "The resource you requested does not exist"
}
```

## 错误返回码

错误码	错误描述	HTTP状态码	中文解释
InternalFailure	We encountered an Internal error. Please try again.	400	内部错误
ValidationError	具体错误原因	400	校验失败
InvalidVersion	The API version specified was invalid.	400	非法版本号
ServiceInternalError	Internal service occurs error.	500	内部（未知）服务器错误
PaymentFailed	payment failed, please check remaining balance	403	支付失败
InsufficientBalance	Insufficient balance, please check remaining balance	403	支付失败，余额不足
NoSuchObject	The specified object is not found or resource do not exist.	404	资源不存在
OperationDenied	No permission to accessor or permission error.	403	没有权限做此操作
BadRequest	Bad request parameters or illegal request.	400	参数输入不合法
InstanceNotFound	The specified instance does not exist.	404	指定的RabbitMQ实例不存在
QualifyNotPass	The User has not pass qualify.	403	用户未通过实名认证
AuthValidateFailed	Fail to validate authorization	401	鉴权失败
AccessDenied	Access denied.	403	禁止操作该实例
InvalidAction	Invalid Action.	400	非法操作
MissingParameter	A required parameter '' is not supplied.	400	缺少必传参数''
MissingDateHeader	Request must have a "date" or "x-bce-date" header.	400	Header中缺少"date" 或 "x-bce-date"其中之一
MissingAuthToken	Request must have a "authorization" header.	400	Header中缺少"authorization"
RequestExpired	Request has expired.	400	请求中authorization过期
QuotaExceeded	The amount of instance exceed limit.	400	实例数量超出用户配额
InstanceCreationFailed	Fail to create instance.	400	创建实例失败，通常是由于资源不足，用户金额不足等
NotSupportOperation	Internal server does not support such operation.	501	服务器不支持该操作
InternalServerError	Internal Server Error.	>500	服务器内部错误

## 公共头

## 公共请求头

下表列出了所有RabbitMQ OpenAPI所携带的公共头域。HTTP协议的标准头域不在此处列出。

公共头部	描述
Authorization	包含 Access Key与请求签名。
Host	表示请求API的域名。
Content-Type	application/json; charset=utf-8。
Content-Length	实际请求body大小。
x-bce-date	表示日期的字符串，如果用户使用了标准的Date域，该头域可以不填。当两者同时存在时，以x-bce-date为准。统一使用UTC时间，日期和时间之间加字母T，结尾加字母Z表示UTC时间，如：2014-06-01T23:00:10Z。服务端由到请求会判断本机时间与该时间差值，若大于30分钟，则抛弃本次请求，响应HTTP 400。
Date	同x-bce-date头域，二者必须存在其一。

## 公共响应头

下表列出了所有RabbitMQ OpenAPI的公共响应头域。HTTP协议的标准响应头域不在此处列出。

公共头部	描述
Content-Type	application/json; charset=utf-8。
Content-Length	返回body大小。
x-bce-request-id	对应请求的requestId，由服务端生成并自动设置到响应头域中。

## 计费相关接口

### 实例续费

为一个指定的预付费实例续费，延长该实例的使用时间。

#### 说明

- 后付费实例不能续费。
- 处于变配、创建期间的实例不能续费。
- 已开通到期转后付费的实例不能续费。

#### URI格式

Verb	URI
PUT	/v{version}/instance/{instanceId}?renew

#### 请求头域

除公共头域外，无其它特殊头域。

#### 请求参数

参数名称	类型	是否必须	参数位置	描述
version	string	是	URL参数	API版本号
instanceId	string	是	URL参数	指定实例Id
clientToken	string	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串。
billing	Billing	是	RequestBody参数	续费信息（只传续费时长）。

### 返回头域

除公共头域，无其它特殊头域。

### 返回参数

无特殊返回参数

### 请求示例

```
PUT /v{version}/instance/rmq-xF9tIG?renew&clientToken=be31b98c-5e41-4838-9830-9be700de5a20 HTTP/1.1
HOST: rabbitmq.bj.baidubce.com
Authorization: bce-auth-v1/rdsak/2016-07-03T06:13:13Z/1800/x-bce-date/{signature}

{
  "billing":{
    "reservation": {
      "reservationLength": 1
    }
  }
}
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 0
```

## ☞ 预付费转后付费

为一个指定的预付费实例开通转后付费，预付费到期后生效。

### 说明

- 只有预付费方式的实例才支持转后付费操作。
- 处于变配、创建期间的实例不能转后付费。
- 只有预付费到期后，开通转后付费才生效。

### URI格式

Verb	URI
POST	/v{version}/instance/{instanceId}?toPostpay

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必须	参数位置	描述
version	string	是	URL参数	API版本号
instanceId	string	是	URL参数	指定实例Id
clientToken	string	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串。

### 返回头域

除公共头域，无其它特殊头域。

### 返回参数

参数名称	类型	描述
orderId	string	订单号

### 请求示例

```
POST /v{version}/instance/rmq-xF9tIG?toPostpay&clientToken=be31b98c-5e41-4838-9830-9be700de5a20 HTTP/1.1
HOST: rabbitmq.bj.baidubce.com
Authorization: bce-auth-v1/rdsak/2016-07-03T06:13:13Z/1800/x-bce-date/{signature}
{
}
}
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 0

{
  "result" : {
    "orderId": "134-134-1234-134"
  }
}
```

## 取消预付费转后付费

为已开通过预付费转后付费的实例关闭转后付费功能。

### 说明

- 只有开通过预付费转后付费的实例才支持取消预付费转后付费操作。
- 处于变配、创建期间的实例不能取消预付费转后付费。
- 取消预付费转后付费成功后立即生效。

### URI格式

Verb	URI
POST	/v{version}/instance/{instanceId}?cancelToPostpay

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必须	参数位置	描述
version	string	是	URL参数	API版本号
instanceId	string	是	URL参数	指定实例Id
clientToken	string	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串。

### 返回头域

除公共头域，无其它特殊头域。

### 返回参数

参数名称	类型	描述
orderId	string	订单号

### 请求示例

```
POST /v{version}/instance/rmq-xF9tIG?cancelToPostpay&clientToken=be31b98c-5e41-4838-9830-9be700de5a20
HTTP/1.1
HOST: rabbitmq.bj.baidubce.com
Authorization: bce-auth-v1/rdsak/2016-07-03T06:13:13Z/1800/x-bce-date/{signature}
{
}
}
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 0
{
  "result" : {
    "orderId": "134-134-1234-134"
  }
}
```

## 后付费转预付费

为一个指定的后付费实例开通转预付费，立即生效。

### 说明

- 只有后付费方式的实例才支持转预付费操作。
- 处于变配、创建期间的实例不能转预付费。
- 后付费转预付费立即生效。

### URI格式

Verb	URI
POST	/v{version}/instance/{instanceId}?toPrepay

### 请求头域

除公共头域外，无其它特殊头域。

### 请求参数

参数名称	类型	是否必须	参数位置	描述
version	string	是	URL参数	API版本号
instanceId	string	是	URL参数	指定实例Id
duration	int	是	RequestBody参数	购买时长
clientToken	string	是	Query参数	幂等性Token，是一个长度不超过64位的ASCII字符串。

### 返回头域

除公共头域，无其它特殊头域。

### 返回参数

参数名称	类型	描述
orderId	string	订单号

### 请求示例

```
POST /v{version}/instance/rmq-xF9tlG?toPrepay&clientToken=be31b98c-5e41-4838-9830-9be700de5a20 HTTP/1.1
HOST: rabbitmq.bj.baidubce.com
Authorization: bce-auth-v1/rdsak/2016-07-03T06:13:13Z/1800/x-bce-date/{signature}
{
  "duration": 1
}
```

### 返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 7869616F-7A68-6977-656E-406261696475
Content-Type: application/json
Content-Length: 0
{
  "result" : {
    "orderId": "134-134-1234-134"
  }
}
```

## 对象说明

[Instance Object](#)详情

参数名称	类型	描述
instanceId	String	实例ID
instanceName	String	实例名称
instanceStatus	String	实例状态
instanceType	String	实例类型：单机（Single），集群（Cluster）
description	String	实例描述信息
engine	String	消息引擎
engineVersion	String	消息引擎版本
specification	Struct	实例规格：单机返回实例规格（CPU核数和内存，内存单位GB），集群返回实例规格和节点数
storageInGB	Int	节点存储空间，单位GB
accountName	String	实例用户名
createTime	DateTime	实例创建时间，0时区时间，样式：2016-04-05T10:20:35Z
connectionAddresses	String	实例连接地址
port	Int	实例连接端口
vpId	String	VPC短ID
subnets	List SubnetDetail	实例所在可用区和子网

**Node参数**

参数	类型	描述
nodeList	List RabbitMQNode	实例各节点信息

**RabbitMQNode**

参数	类型	描述
nodeId	String	节点ID
nodeName	String	节点名称
nodeType	String	节点类型（RAM、DISC）
nodeStatus	String	节点状态
bcId	String	BCC ID

**SubnetDetail Object详情**

参数名称	类型	描述
zoneName	String	子网所在可用区
subnetId	String	子网ID，填短ID

**Specification**

参数名称	类型	描述
cpuCount	int	CPU核数
memoryCapacityInGB	int	内存大小，单位GB
nodeCount	Int	集群节点数量

## 附录

### 🔗 订单相关

#### Billing相关

状态	类型	描述
paymentTiming	String	付费方式。预付费：Prepaid，后付费：Postpaid
reservation	Reservation	保留信息，支付方式为后支付时不需要设置，预支付时必须设置

#### Reservation

状态	类型	描述
reservationLength	int	时长
reservationTimeUnit	String	时间单位，Month，当前仅支持按月

### 🔗 实例可选套餐

实例类型	节点数量	单节点CPU&MEM规格	单节点存储规格
单节点	1	2C4G, 4C8G, 8C16G, 16C32G	100G~2000G
集群	3, 5, 7	4C8G, 8C16G, 16C32G	100G~2000G

### 🔗 实例相关

#### 实例状态

状态值	描述
CREATING	创建中，即申请实例后，在实例状态进入运行中之前的状态
RUNNING	运行中
RESTARTING	实例重启中
CLASS_CHANGING	实例升降级中
EXPIRED	实例已过期或已欠费
STARTING	(充值/续费) 启动中，从已关闭到运行中的中间状态
STOPPING	(过期/欠费) 停机中

## 典型实践

### 访问实例

本文档通过一个简单的例子来展示如何使用代码来访问RabbitMQ实例。

### 🔗 前置条件

1. 参考操作指南，创建RabbitMQ实例，并且记录下创建时配置的账号和密码。

### 基本信息

实例名称

未填写将自动生成

支持大小写字母、数字、中文以及\_./等特殊字符，必须以字母开头，不超过65个字符

外传

\* 用户名:

rmq\_123



由英文字母、数字、下划线组成，长度为4~64的字符，且不为root

\* 密码:

.....



8-32位字符，英文，数字和符号必须同时存在，符号仅限!#\$%^&()

.....

描述:

请输入实例描述信息

0/1024

2. RabbitMQ实例创建完毕后，在实例详情页面获取实例的地址和端口，以及所在的VPC。

**实例信息**

实例名称: rabbitmq-demo	实例ID: rmq-sBdKtT	状态: ● 运行中
引擎版本: RabbitMQ 3.7	架构类型: 单机版	实例规格: 8核16G
付费方式: 后付费	连接地址: sBdKTtPGO.bce.qa.sandbox.baidu.com	连接端口: 5672
用户名: rmq_123 <a href="#">重置密码</a>	创建时间: 2019-12-18 10:45:07	描述信息: - <a href="#">+</a>

---

**存储信息**

磁盘类型: 本地SSD磁盘	可用存储空间: 100GB
---------------	---------------

---

**网络信息**

可用区: 可用区B	VPC: vpc1 (192.168.64.0/18)	子网: subnet1 (192.168.72.0/21)
-----------	-----------------------------	-------------------------------

3. 用户创建可用的BCC实例，且BCC实例的VPC与RabbitMQ实例的VPC相同。

### 配置信息

CPU: 1核

内存: 4GB

公网IP: 未分配

所在网络: vpc1 (192.168.64.0/18)

VPC需要与RabbitMQ的相同

内网IP: 192.168.72.65 [10.107.233.122] [变更内网IP](#)

操作系统: Ubuntu / 18.04 LTS amd64 (64bit) [重装操作系统](#)

4. 将BCC的IP地址加入RabbitMQ实例的白名单中。



### 🔗 详细步骤

1. 登录BCC，若没有EIP进行公网绑定，可使用网页版进行登录
2. 下载安装JDK环境，确保JDK版本在1.8及以上。

```
javac -version # javac正确安装且版本在1.8以上
```

```
root@instance-939nn03m:~/demo_dir/rabbitmq-java-deom# javac -version
javac 1.8.0_222
```

3. 下载并解压实例代码。

```
wget 'https://doc-attachment.bj.bcebos.com/rabbitmq-java-deom.zip'
unzip rabbitmq-java-deom.zip
```

4. 打开conf目录，修改目录下的Connection.properties文件，配置实例信息。

```
host=sBdKTtPG0.bce.qa.sandbox.baidu.com # 实例链接地址
port=5672 # 实例连接端口
virtual_host= # virtual host配置
username=rmq_123 # 创建时设定的用户名
password=password!123 # 创建时设定的密码
```

5. 进入代码根目录，编译示例代码。

```
sh recompile.sh
```

6. 运行代码。这里仅演示示例1-HelloWorld，首先，需要配置类搜索路径。

```
source set_env.sh # 设置类搜索路径在变量CP中
```

接下来，运行1-HelloWorld中的Send类，向队列发送信息，可以发送多次。

```
java -cp $CP Send
```

最后运行1-HelloWorld中的Recv类，从队列读取消息，按Ctrl+c退出

```
java -cp $CP Recv
```

具体过程如下图所示：

```
root@instance-939nn03m:~/demo_dir/rabbitmq-java-deom# source set_env.sh
root@instance-939nn03m:~/demo_dir/rabbitmq-java-deom# java -cp $CP Send
[x] Sent 'Hello World!'
root@instance-939nn03m:~/demo_dir/rabbitmq-java-deom# java -cp $CP Send
[x] Sent 'Hello World!'
root@instance-939nn03m:~/demo_dir/rabbitmq-java-deom# java -cp $CP Recv
[*] Waiting for messages. To exit press CTRL+C
[x] Received 'Hello World!'
[x] Received 'Hello World!'
^Croot@instance-939nn03m:~/demo_dir/rabbitmq-java-deom#
```

🔗 参考

<https://github.com/rabbitmq/rabbitmq-tutorials/tree/master/java>