消息队列 RocektMQ 版

产品文档



目录

消息队列 RocektMQ 版 5
- 产品简介 5
  - 产品简介 5
  - 使用限制 6
  - 开源对比 8
  - 产品优势 9
  - 开源 RocketMQ 版本支持说明 11
- 购买指南 13
  - 购买指南 13
- 快速入门 14
  - 创建 RocketMQ 资源 14
  - 5.x 集群使用 5.x SDK 收发普通消息 19
  - 4.x 集群使用 4.x SDK 收发普通消息 24
- 操作指南 30
  - 集群管理 30
  - 命名空间管理 32
  - Topic 管理 34
  - Group 管理 38
  - 角色与鉴权 44
  - 监控大盘 47
  - 消息查询 54
    - 查询普通消息 54
    - 查看死信队列 60
    - 查询重试消息 63
    - 查询定时与延时消息 65
    - 消息轨迹与说明 67
  - 导入元数据 71
  - 单元管理 73
  - 单元化路由 76
  - 消息跨地域复制 79
- 开发指南 82
  - 消息类型 82
    - 普通消息 82
    - 定时与延时消息 83
    - 顺序消息 85
    - 事务消息 88
  - 消息过滤 90
  - 消费模式 93
  - 消息重试 94
  - 死信队列 96
  - 集群消费与广播消费 97
- SDK文档 99
  - SDK 概述 99
  - 4.x SDK 102
    - Spring Boot Starter 接入 102
      - 发送与接收普通消息 102
      - 发送与接收过滤消息 106
      - 发送与接收延迟消息 108
    - Java SDK 109
      - 发送与接收普通消息 109
      - 发送与接收延迟消息 114
      - 发送与接收顺序消息 118
      - 发送与接收事务消息 122
      - 发送与接收过滤消息 126
      - 发送与接收广播消息 129
    - C++ SDK 132
    - Go SDK 135
    - Python SDK 139
    - C# SDK 143
- 实践教程 146
  - RocketMQ 常见概念命名规范 146
  - RocketMQ 客户端实践 149
  - 客户端风险说明和更新指南 153
- 常见问题 155
  - 控制台常见问题 155
  - 客户端接入问题 156
  - 发送消息报错 157
  - 消费消息报错 159
  - 消息拉取频率介绍 160
- 术语 161
  - 基本概念 161
- 版本说明书 165
  - 版本说明书 165
- API文档 166
  - （tdmq） 166
    - 版本（2020-02-17） 166
      - API 概览 166
      - 调用方式 172
        - 接口签名v1 172
        - 接口签名v3 179
        - 请求结构 188
        - 返回结果 189
        - 公共参数 192
      - CMQ消息相关接口 194
        - 清空cmq消息队列中的消息 194
        - 清空cmq订阅者消息标签 196
        - 发送cmq主题消息 198
        - 发送cmq消息 200
      - CMQ管理相关接口 202

- 创建cmq队列接口 ····· 202
- 创建cmq订阅接口 ····· 205
- 创建cmq主题 ····· 208
- 删除cmq队列 ····· 210
- 删除cmq订阅 ····· 212
- 删除cmq主题 ····· 214
- 枚举cmq死信队列源队列 ····· 216
- 查询cmq队列详情 ····· 218
- 查询cmq全量队列 ····· 220
- 查询cmq订阅详情 ····· 222
- 查询cmq主题详情 ····· 224
- 枚举cmq全量主题 ····· 226
- 修改cmq队列属性 ····· 228
- 修改cmq订阅属性 ····· 231
- 修改cmq主题属性 ····· 234
- 回溯cmq队列 ····· 236
- 解绑cmq死信队列 ····· 238

- RocketMQ 主题相关接口 ····· 240
  - 创建RocketMQ主题 ····· 240
  - 删除RocketMQ主题 ····· 242
  - 获取Topic生产详情列表 ····· 244
  - 获取RocketMQ主题列表 ····· 246
  - 更新RocketMQ主题信息 ····· 249
- RocketMQ 命名空间相关接口 ····· 251
  - 创建RocketMQ命名空间 ····· 251
  - 删除RocketMQ命名空间 ····· 253
  - 获取RocketMQ命名空间列表 ····· 255
  - 更新RocketMQ命名空间 ····· 257
- RocketMQ 平滑迁移相关接口 ····· 259
  - 平滑迁移：查询Topic迁移状态列表 ····· 259
  - 获取RocketMQ平滑迁移任务详情 ····· 261
  - 获取RocketMQ平滑迁移任务列表 ····· 264
  - 平滑迁移：获取源集群的group列表 ····· 266
  - 平滑迁移：获取源集群的topic列表 ····· 268
  - 平滑迁移：导入消费者组列表 ····· 270
  - 平滑迁移：导入topic列表 ····· 272
- RocketMQ 消息相关接口 ····· 274
  - rocketmq消息详情 ····· 274
  - 查询RocketMQ消息轨迹 ····· 277
  - rocketmq 消息查询 ····· 279
  - 导出RocketMQ消息详情 ····· 282
  - 重置RocketMQ消费位点 ····· 285
  - 重发RocketMQ死信消息 ····· 287
  - 发送RocketMQ消息 ····· 289
  - Rocketmq消费验证 ····· 291
- RocketMQ 消费组相关接口 ····· 293
  - 创建RocketMQ消费组 ····· 293
  - 删除RocketMQ消费组 ····· 295
  - 获取消费详情列表 ····· 297
  - 获取在线消费端详情 ····· 299
  - 获取指定消费组下当前客户端的连接情况 ····· 301
  - 获取RocketMQ消费组列表 ····· 303
  - 获取RocketMQ消费组订阅关系 ····· 306
  - 获取指定消费组下订阅的主题列表 ····· 308
  - 更新RocketMQ消费组信息 ····· 310
- RocketMQ 角色授权相关接口 ····· 312
  - 创建RocketMQ环境角色授权 ····· 312
  - 创建RocketMQ角色 ····· 315
  - 批量删除RocketMQ环境角色授权 ····· 317
  - 批量删除RocketMQ角色 ····· 319
  - 获取RocketMQ命名空间角色列表 ····· 321
  - 获取RocketMQ角色列表 ····· 323
  - 修改RocketMQ环境角色授权 ····· 325
  - RocketMQ角色修改 ····· 327
- RocketMQ 集群相关接口 ····· 329
  - 创建RocketMQ集群 ····· 329
  - 创建RocketMQ专享实例 ····· 331
  - 删除RocketMQ集群 ····· 334
  - 删除RocketMQ专享实例 ····· 336
  - 获取单个RocketMQ集群信息 ····· 338
  - 获取RocketMQ的集群列表 ····· 340
  - 拉取公网指标监控数据 ····· 342
  - 查询RocketMQ实例公网接入点信息 ····· 344
  - 获取 RocketMQ 集群相关指标排序列表 ····· 346
  - 获取单个RocketMQ专享集群信息 ····· 348
  - 查询RocketMQ专享实例列表 ····· 350
  - 更新RocketMQ集群信息 ····· 352
  - 修改RocketMQ专享实例 ····· 354
  - 修改RocketMQ专享实例配置 ····· 356
  - 设置RocketMQ实例公网访问接入点 ····· 358
- 主题相关接口 ····· 360
  - 新增主题 ····· 360
  - 删除主题 ····· 363
  - 查询跨地域复制目标集群 ····· 365
  - 查询topic跨地域复制绑定关系 ····· 367
  - 查询主题列表 ····· 369
  - 修改主题 ····· 372
- 其他接口 ····· 374
  - 删除角色 ····· 374
  - 获取租户VPC绑定关系 ····· 376
  - 运营端获取命名空间bundle列表 ····· 378

- 运营端获节点健康状态 380
- 发送消息 382
- 命名空间相关接口 384
- 创建命名空间 384
- 删除命名空间 387
- 获取命名空间属性 389
- 获取命名空间角色列表 392
- 获取命名空间列表 394
- 修改命名空间属性 396
- 消息相关接口 398
- 确认消息 398
- 接收消息 400
- 批量发送消息 402
- 发送单条消息 405
- 环境角色授权相关接口 407
- 检验批量导入角色 407
- 批量导入角色 409
- 创建环境角色授权 411
- 创建角色 413
- 删除环境角色授权 415
- DescribeBatchRole 417
- 获取角色列表 419
- 修改环境角色授权 421
- 角色修改 423
- 获取用户项目权限列表 425
- 修改资源对应的项目（包含新增） 427
- 批量导入角色预校验 429
- 生产消费相关接口 431
- 创建订阅关系 431
- 删除订阅关系 434
- 生产者列表 436
- 消费订阅列表 438
- 消息回溯 441
- 集群相关接口 443
- 创建集群 443
- 删除集群 445
- 获取专享集群列表 447
- 获取集群详情 449
- 获取集群列表 451
- 获取用户可以使用物理集群列表 453
- 更新集群信息 455
- 数据结构 457
- 错误码 502
- 版本（2021-06-21） 509
- API 概览 509
- 调用方式 510
- 接口签名v1 510
- 接口签名v3 517
- 请求结构 526
- 返回结果 527
- 公共参数 530
- 集群相关接口 532
- TDMQ询价接口 532
- 数据结构 534
- 错误码 535

产品简介
产品简介

消息队列 RocketMQ 版（TDMQ for RocketMQ，简称 TDMQ RocketMQ 版）是基于 Apache RocketMQ 构建的分布式消息中间件，完全兼容 Apache RocketMQ 的各个组件与概念，支持开源社区版本的客户端零改造接入。
消息队列 RocketMQ 版具有低延迟、高性能、高可靠、万亿级消息容量和灵活可扩展等特点，可为分布式应用系统提供异步解耦和削峰填谷的能力，同时也具备互联网应用所需的海量消息堆积、高吞吐、可靠重试等特性。

使用限制

本文列举了 TDMQ RocketMQ 版中对一些指标和性能的限制，请您在使用中注意不要超出对应的限制值，避免出现异常。

集群

限制类型	限制
集群名称长度	3-64个字符

命名空间

限制类型	限制
单集群内命名空间数量上限	10个
命名空间名称长度	3-32个字符

Topic

限制类型	限制
单集群内 Topic 数量上限	150
Topic 名称长度	3-64个字符

Group

限制类型	限制
单集群内 Group 数量上限	1500
Group 名称长度	3-64个字符

消息

限制类型	限制
消息最大保留时间	3天
消息最大延时	40天
消息大小	4 MB

限制类型	限制
消费位点重置	3天

开源对比

TDMQ RocketMQ 版与开源 RocketMQ 的性能对比详情如下：

功能大类	功能项	TDMQ RocketMQ 版	开源 RocketMQ
基础功能	定时消息	优化（精准秒级），支持任意延时刻度	有限支持（指定延迟级别）
	可视化管理能力	支持对集群、topic、group 等进行可视化管理和详细信息浏览，包括订阅关系、消费者状态等	简单支持，易用性一般，控制台不区分 topic 类型
可用性	弹性伸缩	客户无需关注部署和扩容，无需人工配置，节点注册等操作完全自动化和白屏化；用户可以根据需要随时横向扩容节点数量、增加存储磁盘大小、垂直升级单节点配置	依赖自建运维团队，自动化、白屏化程度低
	高可靠	数据主从副本，容器化秒级自动重启，保证宕机时容量和数据都不受损	支持同步复制和异步复制，需要自行设计部署方案和参数，主从同步方案不支持自动切主
	跨 AZ 高可用部署	支持跨 AZ 高可用部署，避免机房级故障	支持但较为繁琐，需要自行设计部署方案和参数
可观测性	资源大盘	核心指标观测、生产消费报表和细粒度监控	简单支持，部分监控指标缺失
	报警管理	消息积压、延迟告警，云监控联动	不支持
安全管控	租户命名空间隔离	控制台可视化支持	不支持，命名空间 bug 较多，无法实现真正隔离
	主子账号管理	全面支持主子账号，实现 CAM 主子账号及企业间跨账号的授权服务	不支持
迁移工具	开源迁移工具	脚本化一键迁移，可以无缝从开源 RocketMQ 迁移	-

产品优势

产品优势

兼容开源

100% 兼容 Apache RocketMQ 的各个组件与概念，支持客户端零改造接入。

丰富的消息类型

支持普通消息、顺序消息、延时消息、分布式事务消息等多种消息类型，支持消息重试和死信机制，满足各类业务场景。

高性能

单机最高可支持上万级别的生产消费吞吐，分布式架构，无状态服务，可以横向扩容来增强整个集群的吞吐。

易用免运维

提供 API 访问接口，支持开源所有语言和版本的 SDK。提供云平台整套运维服务，实时监控告警，帮助用户快速发现并解决问题，保证服务的可用性。

5.x 系列独有优势

消息队列 RocketMQ 5.x 系列除了兼具低延迟、高性能、高可靠、万亿级消息容量等历史版本的特点之外，充分结合云原生大潮下的基础设施和生态技术，提高资源利用和弹性能力。
相比于自建的 RocketMQ 和 消息队列 TDMQ RocketMQ 版 4.x 系列， 5.x 系列还具有以下优势：

存储和计算弹性

TDMQ RocketMQ 版 5.x 系列使用了存算分离的架构，充分提高了资源利用率和弹性能力。

轻量化 SDK

TDMQ RocketMQ 版 5.x 系列兼容开源社区 SDK，享受开源社区迭代带来的红利。5.x 系列的客户端更加轻量，采用了全新的极简的 API 设计，更易被集成和使用，同时 5.x 系列提供了更多语言的 SDK，使得开发者在使用时有更多的技术栈选择。

基础功能增强

TDMQ RocketMQ 版 5.x 系列有了进一步的功能增强。消费者组有了更多的白屏化设置，如可以在服务端指定消息重试次数和自由绑定死信队列等功能。

可观测性

TDMQ RocketMQ 版 5.x 系列增加了更加丰富的指标，如消息堆积场景相关指标、关键接口的耗时指标、错误分布

指标、存储读写流量等指标。这些指标都得以对接监控和告警功能，同时提供完善的云 API，支持集成自助运维系统。

开源 RocketMQ 版本支持说明

Apache RocketMQ 作为一款成熟的消息队列产品，历经多个主要版本的迭代。TDMQ RocketMQ 版在开源版本的基础上，持续迭代特性、修复 Bug 及稳定性风险，在保持完全兼容开源的基础上，针对企业实际应用场景进行了深度优化，为用户提供更稳定可靠的消息服务体验。

开源版本支持说明

TDMQ RocketMQ 版高度兼容开源 RocketMQ，其高版本和低版本在客户端协议上是向下兼容的。以下是具体的兼容性说明：

TDMQ RocketMQ 版本	可兼容社区版本	兼容性说明
4.x	≤ 4.9.x	100% 协议兼容。基于社区长期支持版本，功能稳定，是企业用户的首选。
5.x	≤ 5.3.x	100% 协议兼容。支持云原生架构下的新特性，如 gRPC 协议、轻量级客户端等。

> 说明：
>
> - 关于 4.x 版本：该版本是开源社区的长期支持版本，经过了大量生产环境的验证。TDMQ RocketMQ 版在此基础上提供了深度稳定性优化，是当前最稳定、最推荐的版本。
> - 关于 5.x 版本：该版本是面向云原生未来的重要版本，引入了新的通信协议和架构。TDMQ RocketMQ 版积极跟进，为用户提供前沿能力，同时保证服务稳定性。

版本选择建议

使用场景	选择建议
自建/其他云上 RocketMQ 迁移上云	- 有匹配版本：选择与自建 RocketMQ 或其他云上 RocketMQ 大版本一致的 TDMQ RocketMQ 版本。例如，自建版本为 4.9.4，则选择 TDMQ RocketMQ 的 4.x 系列集群。 - 无匹配版本： - 向上兼容：建议选择更高版本的 TDMQ RocketMQ 版。例如，自建是 4.6.x 版本，可以使用 4.x 版本或者 5.x 版本的 TDMQ RocketMQ 集群。TDMQ RocketMQ 的每个版本都向下兼容客户端协议。 - 不建议降级：自建为高版本时，避免选择低版本的 TDMQ RocketMQ 版，以免存在特性不兼容的风险。例如，自建是 5.0.x 版本，建议使用 5.x 版本集群（使用 5.x 版本的客户端 SDK 连接 4.x 系列集群可能会出现部分功能不兼容）。

使用场景	选择建议
第一次使用 RocketMQ	若无特殊需求，建议选择 TDMQ RocketMQ 5.x 版本。5.x 版本有较好的兼容性，同时提供了更丰富的功能，更高的性价比，想要享受云原生的技术红利，可评估选择 5.x 版本。

版本升级说明

TDMQ RocketMQ 版暂不产品化支持 4.x 系列集群到 5.x 系列集群的升级，但是客户可以通过导出元数据，切换接入点的方式进行手动迁移，详细操作见 导入元数据。

购买指南
购买指南

计费方式

TDMQ RocketMQ 集群计费方式为按量付费（后付费），按量计费是一种根据您所购买的资源规格的实际使用量计费的付费方式，主要适用于测试或者流量峰值不确定的场景。您可以先使用资源后再付费，费用按照小时整点结算。

> 说明：
>
> 以下计费说明仅适用于 4.x 版本集群，5.x版本集群的计费能力待上线。

计费项目

TDMQ RocketMQ 版以集群的形式售卖，在按量计费模式下的计费公式如下：
总费用 = API 调用次数费用 + Topic 资源占用费用 =（发送消息 API 调用次数 + 消费消息 API 调用次数）× API 调用次数单价 + Topic 个数 × 天数 × Topic 单价。

计费项	计费规则
API 调用次数费用	API 调用次数的计算规则分为两个维度：消息类型和消息大小。 - 消息类型：消息队列 RocketMQ 版有4种消息类型：普通消息、定时和延时消息、事务消息以及顺序消息，其中定时和延时消息、事务消息和顺序消息都为高级特性消息。 - 普通消息：发送或者消费1条普通消息都计算为1次 API 调用，无论消息是否发送成功或者消费成功，只要发起 API 调用都会进行计费。 - 高级特性消息：发送1条消费1条高级特性消息都计算为5次 API 调用。例如1个 Topic 发送2条事务消息1次，消费1条事务消息，则 API 调用次数为2×5+1×5=15次。 - 消息大小： - 单条消息大小的上限为4MB，以4 KB为计量单位，不满4KB按4KB计算。例如，一条18 KB的消息请求，将以⌈18/4⌉=5次 API 调用次数计费。⌈ ⌉表示向上取整数。
Topic 资源占用费用	Topic 计费单价会根据每个 Topic 当天的产生 API 调用次数阶梯变化，每个 Topic 每天都会被收取一次资源占用费，每天的 Topic 资源占用费为所有 Topic 当天产生费用的总和。无论 Topic 当天是否有进行消息收发，都会产生一次计费。

快速入门
创建 RocketMQ 资源

操作场景

在使用客户端进行消息收发前，您需要在消息队列 RocketMQ 版控制台中创建集群、Topic、Group 等资源，运行客户端时需要配置相关的资源信息。

操作步骤

5.x 集群

步骤1：新建集群

1. 登录 RocketMQ 控制台，进入集群管理页面，选择好目标地域。
2. 单击新建集群，填写好集群名称和说明，单击确定，创建一个集群。



3. 在集群列表页面，单击创建好的集群 ID，在集群基本信息页面的接入信息模块，可以查看到集群的接入点信息。



步骤2：创建角色并授权

1. 在集群列表页面，单击步骤1创建好的集群的“ID”，进入集群基本信息页。
2. 在页面上方选择集群权限，单击添加角色，创建一个角色并授予集群的生产消费权限。



步骤3：创建 Topic

1. 在集群权限列表页，选择顶部的 Topic 页签，进入 Topic 列表页。

2. 单击新建，填写好 Topic 名称，消息类型选择普通消息，其他参数保持默认，单击确定，创建一个 Topic。

> 说明：
>
> 本文以收发普通消息为例进行说明，因此，您参考上述步骤创建的普通消息的 Topic，不能用于收发其他类型的消息。



步骤5：创建 Group

1. 在 Topic 列表页，选择顶部的 Group 页签，进入 Group 列表页。
2. 单击新建，填写好 Group 名称，其他参数可保持默认，单击确定，创建一个 Group。



4.x 集群

步骤1：新建集群

1. 登录 RocketMQ 控制台，进入集群管理页面，选择好目标地域。
2. 单击新建集群，填写好集群名称和说明，单击确定，创建一个集群。



3. 在集群列表页面，单击创建好的集群ID，在集群基本信息页面的接入信息模块，可以查看到集群的接入点信息。



步骤2：创建命名空间

1. 在集群列表页面，单击步骤1创建好的集群的“ID”，进入集群基本信息页。
2. 选择顶部的命名空间页签，单击新建，设置好命名空间名称和描述信息，创建一个命名空间。



步骤3：创建角色并授权

1. 在命名空间列表页面选择角色管理，单击添加角色，创建一个角色。
2. 在页面上方选择命名空间页签，找到刚刚创建的命名空间，单击操作栏的配置权限。
3. 单击**配置权限，**选择刚刚创建好的角色，添加生产消费权限。

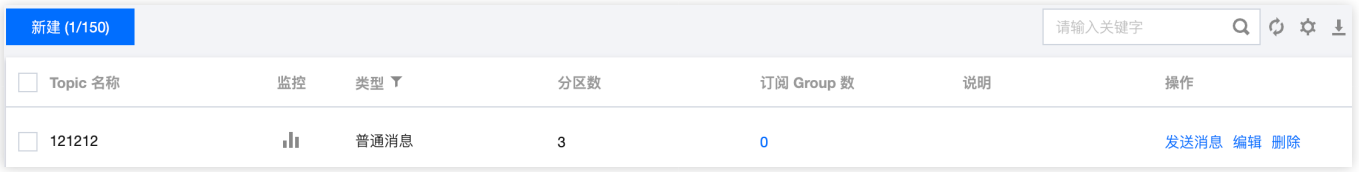


步骤4：创建 Topic

1. 在命名空间列表页，选择顶部的 Topic 页签，进入 Topic 列表页。

2. 选择 步骤2 创建好的命名空间，单击新建，填写好 Topic 名称，消息类型选择普通消息，单击确定，创建一个 Topic。

> 说明：
>
> 本文以收发普通消息为例进行说明，因此，您参考上述步骤创建的普通消息的Topic，不能用于收发其他类型的消息。



步骤5：创建 Group

1. 在 Topic 列表页，选择顶部的 Group 页签，进入 Group 列表页。
2. 选择刚刚创建好的命名空间，单击新建，填写好 Group 名称，协议类型选择 TCP，单击确定，创建一个 Group。



5.x 集群使用 5.x SDK 收发普通消息

操作场景

消息队列 RocketMQ 版支持多种语言的 SDK 收发不同类型的消息，本文以调用 Java SDK 为例介绍通过 5.x SDK 连接消息队列 RocketMQ 版服务端实现普通消息收发的操作过程。

前提条件

- 已创建 RocketMQ 资源
- 安装1.8或以上版本 JDK
- 安装2.5或以上版本 Maven

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 Maven 工程为例，在 pom.xml 添加以下依赖：

```
<dependencies>
    <dependency>
      <groupId>org.apache.rocketmq</groupId>
      <artifactId>rocketmq-client-java</artifactId>
      <version>5.0.6</version>
    </dependency>
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-api</artifactId>
      <version>1.7.7</version>
    </dependency>
</dependencies>
```

步骤2：生产消息

1. 在已创建的 Java 工程中，创建发送普通消息程序并运行。

```
import java.io.IOException;
import java.nio.charset.StandardCharsets;
import java.util.concurrent.CompletableFuture;
```

```
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import org.apache.rocketmq.client.apis.ClientConfiguration;
import org.apache.rocketmq.client.apis.ClientException;
import org.apache.rocketmq.client.apis.ClientServiceProvider;
import org.apache.rocketmq.client.apis.SessionCredentialsProvider;
import org.apache.rocketmq.client.apis.StaticSessionCredentialsProvider;
import org.apache.rocketmq.client.apis.message.Message;
import org.apache.rocketmq.client.apis.producer.Producer;
import org.apache.rocketmq.client.apis.producer.SendReceipt;
import org.apache.rocketmq.client.java.example.AsyncProducerExample;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class NormalMessageSyncProducer {
    private static final Logger log = LoggerFactory.getLogger(NormalMessageSyncProducer.class);

    private NormalMessageSyncProducer() {
    }

    public static void main(String[] args) throws ClientException, IOException {
        final ClientServiceProvider provider = ClientServiceProvider.loadService();

        // 添加配置的ak和sk
        String accessKey = "yourAccessKey"; //ak
        String secretKey = "yourSecretKey"; //sk
        SessionCredentialsProvider sessionCredentialsProvider =
            new StaticSessionCredentialsProvider(accessKey, secretKey);

        // 填写接入地址
        String endpoints = "rmq-xxx.rocketmq.xxxxxxtdmq.com:8081";
        ClientConfiguration clientConfiguration = ClientConfiguration.newBuilder()
            .setEndpoints(endpoints)
            .enableSsl(false)
            .setCredentialProvider(sessionCredentialsProvider)
            .build();
        String topic = "yourNormalTopic";
        // 通常在一个客户端内无需创建过多的生产者。
        final Producer producer = provider.newProducerBuilder()
            .setClientConfiguration(clientConfiguration)
            // 设置主题名，此处的设置非必须，但是推荐设置，以便生产者可以在正式发送消息前，预先抓
取消息路由。
            .setTopics(topic)
            // 如生产者未初始化可能会报 M {@link ClientException} 的错误。
            .build();
        // 此处定义消息主体。
        byte[] body = "This is a normal message for Apache RocketMQ".getBytes(StandardCharset
```

```
s.UTF_8);
        String tag = "yourMessageTagA";
        final Message message = provider.newMessageBuilder()
            // Set topic for the current message.
            .setTopic(topic)
            // 在 topic 下进行的消息二级分类，区别同一个主题内不同的消息。
            .setTag(tag)
            // 消息键，除消息 ID 外可以区别不同消息的其他途径。
            .setKeys("yourMessageKey-1c151062f96e")
            .setBody(body)
            .build();
        try {
            final SendReceipt sendReceipt = producer.send(message);
            log.info("Send message successfully, messageId={}", sendReceipt.getMessageId());
        } catch (Throwable t) {
            log.error("Failed to send message", t);
        }
        // 发送完成后，如无别的需要可以关闭生产者客户端。
        producer.close();
    }
}
```

参数	说明
accessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。
secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。
endpoints	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。
topic	Topic 的名称，在控制台 Topic 管理页面复制。

2. 运行结果如下：

```
INFO: Send message successfully, messageId={一个具体的消息ID}
```

步骤3：消费消息

1. 在已创建的 Java 工程中，创建订阅普通消息程序并运行。
   TDMQ RocketMQ 5.x 系列支持两种消费模式，分别为 Push Consumer 和 Simple Consumer，以下代码示例以 Push Consumer 为例。

```
import java.io.IOException;
import java.util.Collections;
```

```
import org.apache.rocketmq.client.apis.ClientConfiguration;
import org.apache.rocketmq.client.apis.ClientException;
import org.apache.rocketmq.client.apis.ClientServiceProvider;
import org.apache.rocketmq.client.apis.SessionCredentialsProvider;
import org.apache.rocketmq.client.apis.StaticSessionCredentialsProvider;
import org.apache.rocketmq.client.apis.consumer.ConsumeResult;
import org.apache.rocketmq.client.apis.consumer.FilterExpression;
import org.apache.rocketmq.client.apis.consumer.FilterExpressionType;
import org.apache.rocketmq.client.apis.consumer.PushConsumer;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class NormalPushConsumer {
    private static final Logger log = LoggerFactory.getLogger(NormalPushConsumer.class);

    private NormalPushConsumer() {
    }

    public static void main(String[] args) throws ClientException, IOException, InterruptedExceptio
n {
        final ClientServiceProvider provider = ClientServiceProvider.loadService();

        // 添加配置的 ak 和 sk
        String accessKey = "yourAccessKey"; //ak
        String secretKey = "yourSecretKey"; //sk
        SessionCredentialsProvider sessionCredentialsProvider =
            new StaticSessionCredentialsProvider(accessKey, secretKey);

        // 填写接入地址
        String endpoints = "rmq-xxx.rocketmq.xxxxxxxxxtdmq.com:8081";
        ClientConfiguration clientConfiguration = ClientConfiguration.newBuilder()
            .setEndpoints(endpoints)
            .enableSsl(false)
            .setCredentialProvider(sessionCredentialsProvider)
            .build();
        String tag = "*";
        FilterExpression filterExpression = new FilterExpression(tag, FilterExpressionType.TAG);
        String consumerGroup = "yourConsumerGroup";
        String topic = "yourTopic";
        // 通常在一个客户端内无需创建过多的消费者。
        PushConsumer pushConsumer = provider.newPushConsumerBuilder()
            .setClientConfiguration(clientConfiguration)
            // 设置消费者组名称。
            .setConsumerGroup(consumerGroup)
            // 设置消费者订阅名称
            .setSubscriptionExpressions(Collections.singletonMap(topic, filterExpression))
            .setMessageListener(messageView -> {
```

```
                // 处理消息并返回消息消费结果。
                log.info("Consume message={}", messageView);
                return ConsumeResult.SUCCESS;
            })
            .build();
        // 生产环境无需阻塞主线程。
        Thread.sleep(Long.MAX_VALUE);
        // 消费完成后，如无别的需要可以关闭消费者客户端。
        pushConsumer.close();
    }
}
```

参数	说明
accessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。
secretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。
endpoints	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。
consumerGroup	消费者组名称，在控制台 Group 管理页面复制。
topic	Topic 的名称，在控制台 Topic 管理页面复制。

2. 运行结果如下

```
INFO: Consume message=MessageView{messageId='...', topic='yourTopic', body=...}
```

4.x 集群使用 4.x SDK 收发普通消息

操作场景

消息队列 RocketMQ 版支持多种语言的 SDK 收发不同类型的消息，本文以调用 Java SDK 为例介绍通过 4.x SDK 连接消息队列 RocketMQ 版服务端实现普通消息收发的操作过程，

前提条件

- 已创建 RocketMQ 资源
- 安装1.8或以上版本 JDK
- 安装2.5或以上版本 Maven

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 Maven 工程为例，在 pom.xml 添加以下依赖：

> 说明：
>
> 依赖版本要求 ≥ 4.9.3。

```
<!-- in your <dependencies> block -->
<dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.9.3</version>
</dependency>

<dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-acl</artifactId>
    <version>4.9.3</version>
</dependency>
```

步骤2：生产消息

1. 创建消息生产者

```
// 实例化消息生产者Producer
DefaultMQProducer producer = new DefaultMQProducer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))
    // ACL权限
);
// 设置NameServer的地址
producer.setNamesrvAddr(nameserver);
// 启动Producer实例
producer.start();
```

参数	说明
namespace	命名空间的名称，在控制台命名空间页面复制。
groupName	生产者组名称，在控制台集群管理中 Group 页签中复制。
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。
secretKey	角色名称，在控制台的角色管理页面 SecretKey 列复制。
accessKey	角色密钥，在控制台的角色管理页面 AccessKey 列复制。

2. 发送消息

发送消息有多种方式：同步发送、异步发送、单向发送等。

同步发送

```
 for (int i = 0; i < 10; i++) {
    // 创建消息实例，设置topic和消息内容
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes(RemotingHelpe
r.DEFAULT_CHARSET));
    // 发送消息
    SendResult sendResult = producer.send(msg);
    System.out.printf("%s%n", sendResult);
}
```

参数	说明
topic_name	Topic 的名称，在控制台 Topic 管理页面复制。
TAG	用来设置消息的 TAG。

异步发送

```
 // 设置发送失败后不重试
producer.setRetryTimesWhenSendAsyncFailed(0);
// 设置发送消息的数量
int messageCount = 10;
final CountDownLatch countDownLatch = new CountDownLatch(messageCount);
for (int i = 0; i < messageCount; i++) {
    try {
        final int index = i;
        // 创建消息实例，设置topic和消息内容
        Message msg = new Message(topic_name, "TAG", ("Hello rocketMq " + index).getBytes(Remot
ingHelper.DEFAULT_CHARSET));
        producer.send(msg, new SendCallback() {
            @Override
            public void onSuccess(SendResult sendResult) {
                // 消息发送成功逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d OK %s %n", index, sendResult.getMsgId());
            }

            @Override
            public void onException(Throwable e) {
                // 消息发送失败逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d Exception %s %n", index, e);
                e.printStackTrace();
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }
}
countDownLatch.await(5, TimeUnit.SECONDS);
```

参数	说明
topic_name	Topic 的名称，在控制台 Topic 管理页面复制。
TAG	用来设置消息的 TAG。

单向发送

```
 for (int i = 0; i < 10; i++) {
    // 创建消息实例，设置topic和消息内容
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes(RemotingHelpe
r.DEFAULT_CHARSET));
```

```
    // 发送单向消息
    producer.sendOneway(msg);
}
```

参数	说明
topic_name	Topic 的名称，在控制台 Topic 管理页面复制。
TAG	用来设置消息的 TAG。

步骤3：消费消息

1. 创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费者模式。

push 消费者

```
  // 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
// 设置NameServer的地址
pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
namespace	命名空间的名称，在控制台命名空间页面复制。
groupName	生产者组名称，在控制台集群管理中 Group 页签中复制。
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。
secretKey	角色名称，在控制台的角色管理页面 SecretKey 列复制。
accessKey	角色密钥，在控制台的角色管理页面 AccessKey 列复制。

pull 消费者

```
  // 实例化消费者
DefaultLitePullConsumer pullConsumer = new DefaultLitePullConsumer(
    namespace,
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)));
// 设置NameServer的地址
pullConsumer.setNamesrvAddr(nameserver);
```

```
// 设置从第一个偏移量开始消费
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_FIRST_OFFSET);
```

参数	说明
namespace	命名空间的名称，在控制台命名空间页面复制。
groupName	生产者组名称，在控制台集群管理中 Group 页签中复制。
nameserver	集群接入地址，在控制台集群基本信息页面的接入信息模块获取。
secretKey	角色名称，在控制台的角色管理页面 SecretKey 列复制。
accessKey	角色密钥，在控制台的角色管理页面 AccessKey 列复制。

2. 订阅消息

根据消费模式不同，订阅方式也有所区别。

push 订阅

```
  // 订阅topic
pushConsumer.subscribe(topic_name, "*");
// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
    // 消息处理逻辑
    System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
    // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
    return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

参数	说明
topic_name	Topic 的名称，在控制台 Topic 管理页面复制。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

pull 订阅

```
// 订阅topic
pullConsumer.subscribe(topic_name, "*");
// 启动消费者实例
pullConsumer.start();
try {
```

```
        System.out.printf("Consumer Started.%n");
        while (true) {
            // 拉取消息
            List<MessageExt> messageExts = pullConsumer.poll();
            System.out.printf("%s%n", messageExts);
        }
    } finally {
        pullConsumer.shutdown();
    }
```

参数	说明
topic_name	Topic 的名称，在控制台 Topic 管理页面复制。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

操作指南
集群管理

操作场景

集群是 TDMQ RocketMQ 版中的一个资源维度，不同集群的 Topic、Group 等资源完全隔离。每个集群会有集群的资源限制例如 Topic 总数、消息保留时长等。常见的使用方式如：开发测试环境使用一个专门集群，生产环境使用一个专门的集群。
在完成集群资源评估后，您可以根据业务需求在 TDMQ RocketMQ 版控制台购买对应规格的集群。

操作步骤

创建集群

1. 登录 TDMQ RocketMQ 控制台，进入集群管理页面。
2. 在集群管理页面，选择地域后，单击新建集群进入新建集群对话框。
    - 计费模式：按量计费。
    - 地域：选择与您的业务最靠近的地域，处于不同地域的云产品内网不通，购买后不能更换，请您谨慎选择；例如，城市1地域的云服务器无法通过内网访问城市2地域的集群。
    - 集群规格：根据您的业务需求选择合适的集群规格。
    - 集群名称：填写集群名称， 3-64个字符，只能包含数字、字母、“-”和“_”。
    - 标签：标签用于从不同维度对资源分类管理。
3. 单击立即购买，完成集群创建。

编辑集群

1. 在集群管理列表页，单击操作列的编辑。
2. 在弹窗中填写集群名称和集群说明，单击提交。

删除集群

> 注意：
>
> 删除后，该集群下的所有配置都会被清空，且无法恢复，请谨慎操作。当集群内存在资源时，需清空后才可以删除集群。

如果想删掉创建的集群，可以通过以下步骤操作：

1. 在集群管理列表页，单击操作列的删除。

2. 在删除的确认弹框中，单击删除，即可删除集群。

命名空间管理

操作场景

命名空间是 TDMQ RocketMQ 4.x 集群中的一个资源管理概念。用户不同的业务场景一般都可以通过命名空间做隔离，并且针对不同的业务场景设置专门的配置。不同命名空间之间的 Topic 相互隔离，订阅相互隔离，角色权限相互隔离。
本文档指导您使用消息队列 TDMQ RocketMQ 版时，创建多个命名空间，以便在同一个集群下将 TDMQ RocketMQ 版应用于不同的场景。

约束与限制

- 仅 4.x 集群需要创建命名空间，5.x 集群无需创建。
- 每个集群下最多可以创建10个命名空间。
- 同一个命名空间下的 Topic 和 Group 的名称唯一。
- 消息成功生产后，默认在服务端保留3天，无论是否被消费，到时间则自动删除。

操作步骤

创建命名空间

1. 登录 RocketMQ 控制台，选择地域后，单击目标集群的 “ID”，进入基本信息页面。
2. 在页面上方选择命名空间页签，单击新建进入创建命名空间页面。
3. 在新建命名空间对话框，设置命名空间的相关属性配置。
    - 命名空间名称：设置命名空间的名称（创建后不可修改），3-32个字符，只能包含字母、数字及'_'
    - 命名空间描述：命名空间的备注说明
4. 单击保存完成所在集群命名空间的创建。
   后续步骤：接下来就可以在该命名空间中 创建 Topic 进行消息的生产和消费。

获取接入地址

在命名空间列表页，在 VPC 内网接入地址可获取命名空间的 TCP 协议接入地址。

修改命名空间

如果需要重新修改编辑，可以通过以下步骤操作：

1. 在命名空间列表页，单击操作列的编辑，进入编辑页面。
2. 修改命名空间说明，单击保存完成修改。

删除命名空间

如果想删掉创建的命名空间，可以通过以下步骤操作：

1. 在命名空间列表页，单击操作列的删除。
2. 在删除的确认弹框中，单击确认，即可删除命名空间。

Topic 管理

操作场景

Topic 是 TDMQ RocketMQ 版中的核心概念。Topic 通常用来对系统生产的各类消息做一个集中的分类和管理，例如和交易的相关消息可以放在一个名为 “trade” 的 Topic 中，供其他消费者订阅。
在实际应用场景中，一个 Topic 往往代表着一个业务聚合，由开发者根据自身系统设计、数据架构设计来决定如何设计不同的 Topic。
本文档可以指导您使用 TDMQ RocketMQ 版时，利用 Topic 对消息进行分类管理。

前提条件

已创建好对应的命名空间。

操作步骤

创建 Topic

1. 登录 RocketMQ 控制台，选择地域后，单击目标集群的 ID 进入集群基本信息页面。

2. 单击顶部 Topic 页签，选择命名空间后，单击新建进入创建 Topic 页面。

3. 在新建 Topic 对话框中，填写以下信息。
   5.x 集群：
   - Topic 名称：填写 Topic 名称（创建后不可修改），只能包含字母、数字、“%”、“-”及“_”，3-100 字符。
   - 类型：选择消息类型，包括：普通、顺序、延时消息和事务消息。不同消息类型的 Topic 不能混用，例如您创建的普通消息的 Topic，不能用于收发其他类型的消息。
   - 队列数：选择队列数量，最大支持16队列。多队列可以提高单 Topic 的生产消费性能，但是非顺序消息的场景下无法保证顺序性。
   - 消息保留时间：消息持久化后的保留时间，过期不论是否消费均删除。范围 1 天到 3 天。
   - 说明：填写 Topic 的说明信息，最长128字符。

   4.x 集群：
   - Topic 名称：填写 Topic 名称（创建后不可修改），3-64个字符，只能包含字母、数字、“-”及“_”
   - 类型：选择消息类型，包括：普通、顺序消息和事务消息。不同消息类型的 Topic 不能混用，例如您创建的普通消息的 Topic，不能用于收发其他类型的消息。
   - 队列数：选择队列数量，最大支持16队列。多队列可以提高单 Topic 的生产消费性能，但是非顺序消息的场景下无法保证顺序性。
   - 说明：填写 Topic 的说明信息，最长128个字符。

4. 单击提交，在 Topic 列表中即可看见创建好的 Topic。

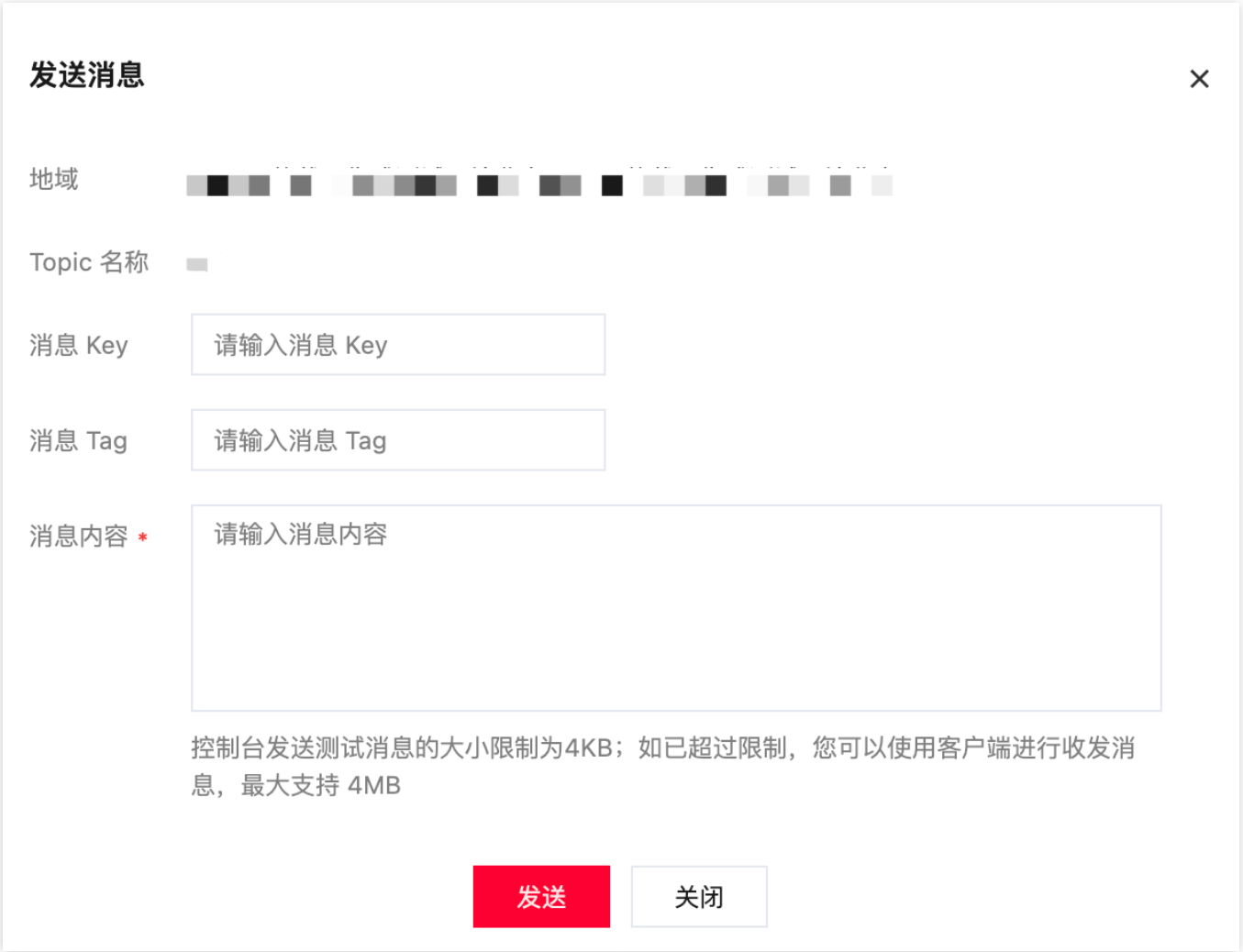
> 说明：
>
> 如当前集群为单元化环境，Topic 创建后会在其他单元创建相同的 Topic 及配置。

发送测试消息

TDMQ RocketMQ 版控制台支持手动发送消息，在控制台进行相应的操作即可实现消息发送给指定的 Topic。

> 说明：
>
> 控制台发送测试消息的大小限制为4KB；如已超过限制，您可以使用客户端进行收发消息，最大支持 4MB。

1. 在 Topic 列表中，单击目标Topic 操作栏的发送消息。
2. 在弹窗中输入消息 Key，消息 Tag 和消息内容，单击提交。



查看 Topic 生产端连接关系

在生产环境中，生产端（Producer）与 Broker 的连接状态直接影响消息发送的稳定性和性能。通过监控连接关系（如客户端 IP、版本、消息发送时间等），可以快速识别网络异常或者资源瓶颈，确保消息高效可靠投递。

> 说明：
>
> 4.x 集群不支持查看生产者信息。

1. 在 Topic 列表中，单击目标 Topic 的名称。
2. 页面跳转到 Group 列表，选择生产者管理页签，查看当前 Topic 下生产者的相关信息。

参数名称	含义与说明
生产者 ID	生产者实例的唯一标识符，通常由客户端在启动时自动生成，用于在 Broker 端区分不同的连接。
生产者地址	生产者客户端所在服务器的 IP 地址和端口，是定位生产者源机器的关键信息。
最近发送时间	该生产者最后一次向此 Topic 发送消息的时间戳，反映了生产者的活跃程度，如果时间与当前时间差距很大，说明该生产者可能已停止工作。
客户端类型	标识生产者是生产者（Producer）还是 回馈生产者（Reply Producer）等类型。
客户端语言	开发生产端所使用的编程语言，例如 Java、Go、C++、Python 等。
客户端版本	生产端所使用的 RocketMQ 客户端库的版本号。用于判断客户端是否过旧，或在出现 Bug 时确认是否属于特定版本的问题。 注意： 请避免使用存在已知风险的客户端版本，详见客户端风险说明和更新指南。

查看 Topic 订阅关系

通过查看 Topic 的订阅 Group，您可以了解订阅了该 Topic 的所有 Group 的订阅状态，包括 Group 列表、在线状态及消息堆积情况等。通过这些信息，您可以确认消息是否被正常消费、及时发现无 Group 订阅的异常 Topic，并评估业务变更的影响范围。

1. 在 Topic 列表中，单击目标 Topic 的订阅 Group 数栏下的数字。
2. 页面跳转到 Group 列表，展示订阅该 Topic 的 Group 信息。

参数名称	含义与说明
Group 名称	消费者组（Consumer Group）的唯一标识。组内所有消费者共同消费同一个 Topic 的消息，实现负载均衡。
状态	消费者组的当前状态，通常为在线或离线，离线状态表示该组内没有任何运行的消费者实例。
消费模式	分为集群消费和广播消费。 - 集群消费：一条消息只会被组内的一个消费者消费。 - 广播消费：一条消息会被组内的每一个消费者消费一次。
协议类型	（仅 4.x 集群展示）消费者使用的通信协议，如 TCP 和 HTTP 等，决定了客户端与 Broker 的交互方式。

参数名称	含义与说明
过滤类型	消息过滤的方式。主要为 TAG 或 SQL92。 - TAG：通过消息的标签进行简单匹配。 - SQL92：通过自定义的 SQL 表达式对消息属性进行复杂过滤。
过滤规则	具体执行的过滤条件。例如在 TAG 模式下可能是 TagA ，在 SQL92 模式下可能是 a > 5 。
消息堆积数量	该消费者组尚未消费的消息总量，是衡量消费健康度的关键指标，堆积持续增长意味着消费速度跟不上生产速度。
消费进度更新时间	该消费者组的消费进度（offset）最后一次被提交到 Broker 的时间，反映了消费端的活跃度，长时间未更新可能意味着消费者已僵死或停止。
操作	如果当前 Topic 下的 Group 处于离线状态（即没有在线客户端），则可以在操作列单击取消订阅以清理历史订阅关系数据，避免历史脏数据的影响。

编辑 Topic

Topic 创建完成后，您可根据业务需求修改 Topic 配置，例如调整队列数、Topic 说明等参数，以优化消息生产消费性能。

1. 在 Topic 列表中，找到需要编辑的 Topic ，单击操作栏中的编辑。
2. 在弹出的对话框中可以对 Topic 信息进行编辑。
3. 单击提交即完成对 Topic 的编辑。

删除 Topic

- 批量删除：在 Topic 列表中，勾选所有需要删除的 Topic，单击左上角的批量删除，在弹出的提示框中，单击提交，完成删除。
- 单个删除：在 Topic 列表中，找到需要删除的 Topic，单击操作列的删除，在弹出的提示框中，单击提交，完成删除。

> 注意：
>
> 删除了 Topic 之后也会清除该 Topic 下积累的未消费消息，请谨慎执行。

导入和导出元数据

TDMQ RocketMQ 版支持在控制台上批量导入和导出 Topic 元数据，以简化 Topic 在多环境间的迁移、备份与批量配置。



- **元数据导出：**通过 Topic 列表页右上角的 按钮直接导出元数据，元数据的导出格式为 .xlsx 的表格文件。
- **元数据导入：**如果您需要将一个集群的 Topic 信息载入到另一个集群内，在导出元数据后，可以单击 Topic 列表



页右上角的 按钮，将 Topic 数据导入到指定的集群内。

Group 管理

操作场景

Group 用于标识一类 Consumer，这类 Consumer 通常消费同一类消息，且消息订阅的逻辑一致。用户在启动消费者客户端时，需要先在控制台创建对应的 Group。

操作步骤

创建 Group

1. 登录 RocketMQ 控制台，选择地域后，单击目标集群的“ID”进入集群基本信息页面。

2. 单击顶部 Group 页签，选择命名空间后，单击新建进入创建 Group 页面。

3. 填写 Group 相关信息。
   5.x 集群
   - Group 名称：填写 Group 名称（创建后不可修改），只能包含字母、数字、“%”、“-”及“_”，3-100个字符
   - Group 说明：填写 Group 说明，最长128字符
   - 最大重试次数：表示消息可以重新被投递的最大次数，超过最大重试次数还没被成功消费，消息将被投递至死信队列或丢弃。
     - 如果您使用的是 RocketMQ 4.x 客户端，消息的重试次数以您在客户端内设置消息重试次数为准。
     - 如果您使用的是 RocketMQ 5.x 客户端，则消息重试次数以您在当前页面设置的为准。
   - 投递顺序性：服务端将消息投递给消费者消费的顺序，支持顺序投递和并发投递，默认投递方式为并发投递。
     - 顺序投递：将逻辑上相关联的消息路由至同一队列并串行化处理，从而严格保证消息按其发送顺序被消费的机制。
     - 并发投递：指消息队列通过将消息分发至多个队列并由多个消费者线程并行处理，旨在最大化系统吞吐量与资源利用率，不保障消息间的处理顺序。
   - 开启消费：关闭后 Group 下的所有消费会暂停，重新开启可继续消费。可用于系统运维，对消费者组进行流控与故障隔离，以实现集群级别的保护与平滑操作。

   4.x 集群
   - Group 名称：填写 Group 名称（创建后不可修改），3-87个字符，只能包含字母、数字、“-”及“_”
   - 协议类型：当前仅支持 TCP 协议

- Group 说明：填写 Group 说明
- 开启消费：关闭后 Group 下的所有消费会暂停，重新开启可继续消费。可用于系统运维，对消费者组进行流控与故障隔离，以实现集群级别的保护与平滑操作。
- 开启广播模式：关闭后 Group 下的所有声明为广播模式的消费者会暂停，重新开启可继续消费。
  - 集群消费：当使用集群消费模式时，任意一条消息只需要被集群内的任意一个消费者处理即可。
  - 广播消费：当使用广播消费模式时，每条消息会被推送给集群内所有注册过的消费者，保证消息至少被每个消费者消费一次。

4. 单击提交，完成 Group 创建。

> 说明：
>
> 如果当前集群为单元化环境，Group 创建后会在其他单元创建相同的 Group 及配置。

查看消费者详情

1. 在 Group 列表，单击 Group 的名称，可以查看 Group 详情。

- 基本信息

  5.x 集群

参数	说明
Group 名称	Group 的名称。
创建时间	Group 的创建时间。
投递顺序性	服务端将消息投递给消费者消费的顺序，支持顺序投递和并发投递，默认投递方式为并发投递。 - 顺序投递：将逻辑上相关联的消息路由至同一队列并串行化处理，从而严格保证消息按其发送顺序被消费的机制。 - 并发投递：指消息队列通过将消息分发至多个队列并由多个消费者线程并行处理，旨在最大化系统吞吐量与资源利用率，不保障消息间的处理顺序。
消费者类型	客户端获取消息的方式，支持 PUSH 或者 PULL。 - PUSH 方式由客户端内部封装的长轮询机制来模拟服务端主动推送。 - PULL 方式则由应用层主动调用客户端的拉取方法。
总消息堆积	消息堆积的总数量。

  4.x 集群

参数	说明

参数	说明
Group 名称	Group 的名称。
创建时间	Group 的创建时间。
消费模式	集群模式或者广播模式。 - 集群消费：当使用集群消费模式时，任意一条消息只需要被集群内的任意一个消费者处理即可。 - 广播消费：当使用广播消费模式时，每条消息会被推送给集群内所有注册过的消费者，保证消息至少被每个消费者消费一次。
客户端协议	仅支持 TCP 协议。
总消息堆积	消息堆积的总数量。
消费者类型	ACTIVELY（客户端主动拉消息，Pull 模式）或 PASSIVELY（客户端等待服务端推消息，Push 模式）。

- 客户端连接

参数名	说明
客户端地址	指客户端所在机器的 IP 地址和端口号。这个端口是客户端进程随机打开的本地端口，用于与 Broker 进行网络通信。
主机名	客户端机器在操作系统层面配置的主机名称。
客户端类型	仅5.x 集群展示。标识客户端的角色，主要是 Producer（生产者）或 Consumer（消费者）。
客户端语言	开发此客户端所使用的编程语言，如 JAVA、CPP、GO、DOTNET 等，RocketMQ 支持多语言客户端。
客户端版本	客户端所使用的 RocketMQ 客户端库的版本，检查此字段有助于确认客户端版本是否与服务器兼容或是否存在已知 Bug。 注意： 请避免使用存在已知风险的客户端版本，详见各语言客户端风险说明。
消息堆积	仅 4.x 集群展示。堆积的消息数量。
操作	- 查看详情**：**可查看消费者详情。 - 查看堆栈：可以查看该客户端的堆栈，也可以在堆栈上方搜索关键词，查看对应的堆栈信息。

- 订阅关系

参数名	含义说明
Topic	该 Group 所订阅的 Topic 名称。
类型	Topic 的类型，包括“普通消息、顺序消息、延时消息和事务消息”。
消息堆积条数	当前 Group 的消息堆积中，各 Topic 下的消息条数
队列数	该 Topic 下的队列总数，队列是 RocketMQ 进行负载均衡和并行消费的最小单元，一个消费者会分配到一个或多个队列进行消费。
过滤模式	消费者订阅消息时的过滤方式。 - TAG：标签过滤模式，通过设置消息的 Tag（标签）进行订阅，是最常用的方式。 - SQL92：使用类 SQL 表达式对消息的 Property（用户属性）进行复杂过滤。
订阅一致性	检查该消费者组下所有消费者实例的订阅关系是否一致。 - 一致：所有消费者实例都订阅了完全相同的 Topic 和 Tag（或 SQL 表达式）。 - 不一致：存在至少一个消费者实例的订阅关系与其他实例不同，这是导致消息丢失或混乱的常见原因，需修复。
操作	如果当前 Group 没有在线客户端订阅某个 Topic，则可以在操作列单击取消订阅以清理历史订阅关系数据，避免历史脏数据的影响。

设置 offset

消费进度（offset）表示消费者消费消息的位置，在某些业务场景下，例如数据处理异常、业务逻辑变更或系统运维需求，可能会需要重置消费位点，以重新消费或跳过特定消息。

> 说明：
>
> TDMQ-RocketMQ 支持给离线的 Group 重置 offset（消费位点），但目前仅支持 Push 消费模式下的消费者组，否则会出现重置失败的情况。

1. 进入 Group 列表页面。
   - 5.x 集群：单击目标 Group 操作列的重置位点。
   - 4.x 集群：单击目标 Group 操作列的重置位点或者重置重试队列位点。

   关于重置重试队列位点的说明：

特性	4.x 集群	5.x 集群
重试队列粒度	以 Group 为单位 一个 Group 只有一个重试队列，它订阅的所有 Topic 的失败消息都发到这里。	以 Group&Topic 为粒度 一个 Group 针对它订阅的每个 Topic，都会有一个独立的专属重试队列。

特性	4.x 集群	5.x 集群
重置普通队列的影响	重置某个 Topic 的消费位点时，不会自动重置重试队列。	重置某个 Topic 的消费位点时，会自动重置与之对应的那个专属重试队列的位点。
为何这样设计？	为了避免“交叉影响”。 如果一个 Group 订阅了 Topic A 和 Topic B，当重置 Topic A 的位点时，如果也重置了共用的重试队列，那么 Topic B 的失败消息也会被影响。	因为天然隔离。 每个 Topic 都有自己独立的“重试队列”，重置 Topic A 时，只会动 Topic A 自己的重试队列，不会影响 Topic B 的重试队列，所以可以安全地自动完成。
额外功能	提供了一个独立的 “重置重试队列位点” 功能。如果需要清除所有失败消息，可以单独执行这个操作。	无需额外功能，因为重置 Topic 时已自动处理。

2. 在弹窗中，可以选择从最新位点开始或者从指定时间点开始设定 Topic 的消费位移 offset（即指定该订阅下的消费者从哪里开始消费消息）。
   如果选择重置位点“从最新位点开始”，则 Group 在指定 Topic 中的堆积消息将被全部清除，请谨慎操作。



3. 单击提交，完成设置。

编辑 Group

1. 在 Group 列表中，单击目标 Group 操作列的编辑。
2. 在弹窗中，对 Group 信息进行编辑。
3. 单击提交，完成修改。

删除 Group

- 批量删除：在 Group 列表中，勾选所有需要删除的 Group，单击左上角的批量删除，在弹出的提示框中，单击提交，完成删除。
- 单个删除：在 Group 列表中，找到需要删除的 Group，单击操作列的删除，在弹出的提示框中，单击提交，完成删除。

> 注意：
>
> 删除 Group 后，由该 Group 标识的消费者将立即停止接收消息，该 Group 下的所有配置将会被清空，且无法恢复，请您谨慎执行该操作。

导入和导出 Group 元数据

TDMQ RocketMQ 版支持在控制台上批量导入和导出 Group 元数据，以简化 Group 在多环境间的迁移、备份与批量配置。



- **元数据导出：**通过 Group 列表页右上角的 按钮直接导出元数据，元数据的导出格式为 .xlsx 的表格文件。
- **元数据导入：**如果您需要将一个集群的 Group 信息载入到另一个集群内，在导出元数据后，可以单击



Group 列表页右上角的 按钮，将 Group 数据导入到指定的集群内。

角色与鉴权

TDMQ RocketMQ 版通过 ACL（Access Control List）策略实现对集群中生产者和消费者权限的控制，您可以在集群下创建多个角色并为其赋予不同资源的生产或者消费权限，以达到不同用户之间权限隔离的目的。每种角色都有其对应的唯一密钥，用户可以通过在客户端中添加密钥来访问 RocketMQ 进行消息的生产消费，当客户端进行消息生产或消费时，系统会进行身份鉴权，未被授权的操作将被拒绝。
这种机制有效实现了不同业务单元之间的权限隔离，既保障了消息系统的安全性，又满足了多团队协作场景下的资源管控需求，通过权限最小化原则从根本上避免了越权访问导致的数据混乱问题。

使用场景

- 用户需要安全地使用 RocketMQ 进行消息的生产消费。
- 用户需要对同一集群设置不同角色的生产消费权限。
  例如：一个公司有 A 部门和 B 部门，A 部门的系统产生交易数据，B 部门的系统根据这些交易数据做数据分析和展示。那么遵循权限最小化原则，可以配置两种角色，A 部门角色只授予往交易系统集群中生产消息的权限，B 部门则只授予消费消息的权限。这样可以很大程度避免由于权限不清带来的数据混乱、业务脏数据等问题。

操作步骤

新增角色

1. 登录 RocketMQ 控制台。
2. 在左侧导航栏选择集群管理，选择地域后，点击要配置角色的集群的“ ID ”，进入基本信息页面。
3. 在页面上方选择集群权限（5.x 集群）或者角色管理（4.x 集群）页签，单击添加角色，输入角色名称，并配置好生产和消费权限。
4. 单击保存，完成当前集群角色的创建。

角色授权

1. 在角色列表页面复制角色密钥。



> 注意
>
> 密钥泄露很可能导致您的数据泄露，请妥善保管您的密钥。

2. 将复制的角色密钥添加到客户端的参数中。
   以下给出一种推荐的方式。

   i. 声明 ACL_SECRET_KEY 和 ACL_SECRET_ACCESS 两个字段，使用各类框架的话建议从配置文件中读取。

   ```
   private static final String ACL_ACCESS_KEY = "eyJr****";
   private static final String ACL_SECRET_KEY = "bigdata"; //使用上图页面的角色名称即可
   ```

   ii. 声明一个静态函数，用于载入一个 RocketMQ Client 的 RPCHook 对象

   ```
   static RPCHook getAclRPCHook() {
       return new AclClientRPCHook(new SessionCredentials(ACL_ACCESS_KEY, ACL_SECRET_KEY));
   }
   ```

   iii. 在创建 RocketMQ producer 、 pushConsumer 或 pullConsumer 的时候，引入 RPCHook 对象

以下为创建一个 producer 的代码示例：

```java
DefaultMQProducer producer = new DefaultMQProducer("rocketmq-mw***|namespace", "ProducerGroupName", getAclRPCHook());
```

3. （仅 4.x 集群需操作）在 TDMQ RocketMQ 控制台选择之前设定了角色的集群后，切换到命名空间，选择需要配置生产消费权限的一个命名空间，单击操作列的配置权限。单击添加角色，在下拉列表中找到刚刚新创建的角色，勾选上需要的权限，单击保存。
4. 检查权限是否生效。
5. 您可以运行配置好的客户端访问对应命名空间中的 Topic 资源，按照刚刚配置的权限进行生产或消费，看是否会产生没有权限的报错信息，如果没有即代表配置成功。

编辑权限

1. 在角色列表中，找到需要编辑权限的角色，单击操作列的编辑。
2. 在编辑的弹框中，修改权限信息后，单击保存。

删除权限

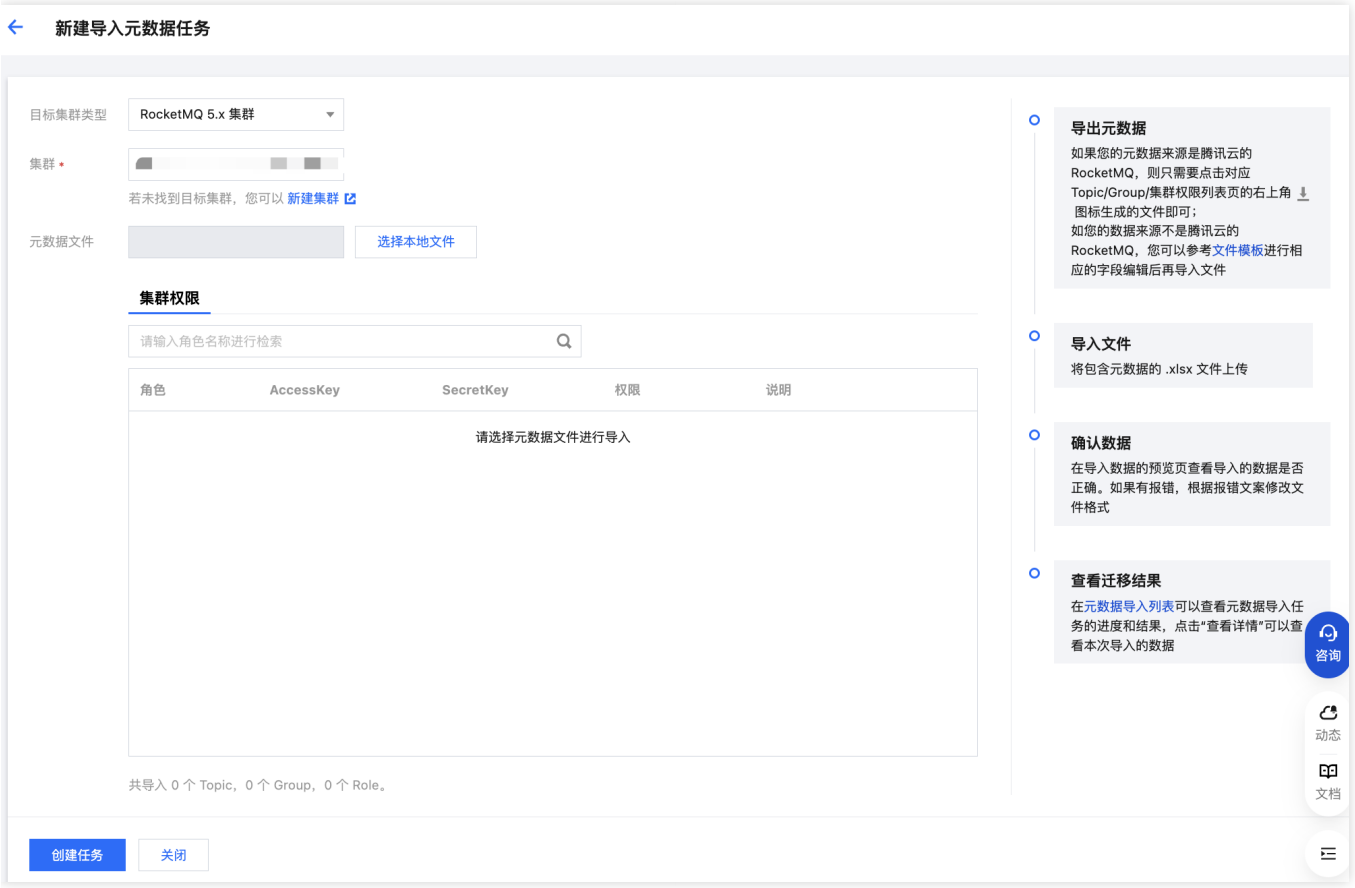
> 注意
>
> 删除角色后，原先使用该角色进行生产或消费消息的密钥（ AccessKey 和 SecretKey ）将立即失效。请确保当前业务已经没有使用该角色进行消息的生产消费，否则可能会出现客户端无法生产消费而导致的异常。

1. 在角色列表中，找到需要删除权限的角色，单击操作列的删除。
2. 在删除的弹框中，单击删除，即可删除该角色。

导入/导出角色

您可以通过角色列表页右上角的 按钮直接导出元数据，元数据的导出格式为 .xlsx 格式的表格文件。

如果您需要将一个集群的角色和权限导入到另一个集群内，在导出元数据后，您可以单击角色列表页右上角的 按钮，将权限数据导入到指定的集群内。如果您的数据来源不是云平台的 RocketMQ，可以参考文件模板链接处找到对应的模板，进行相应的字段编辑后再导入文件。



监控大盘

操作场景

TDMQ RocketMQ 版支持集群监控功能，包括集群的生产消费指标、存储指标和消费者组指标等，您可以根据这些监控数据，分析集群的使用情况，针对可能存在的风险及时处理。同时您也可以对监控项设置报警规则，以便数据异常时收到报警消息，及时处理风险，保障系统的稳定运行。

监控指标

TDMQ RocketMQ 版支持的监控指标如下：
5.x 集群
集群

分类	指标	单位	含义
生产相关指标	生产消息条数	Count/s	当前集群每秒生产的消息条数。
	生产流量	MBytes/s	当前集群生产消息的总流量变化。
	生产者数量	Count	当前集群的生产者客户端数量
	生产成功率	%	当前集群消息生产成功率变化，仅 5.x 客户端能采集到该指标
	生产消息平均耗时(发送RT)	ms	当前集群生产消息平均耗时
	各类型消息数量变化	/	当前集群生产不同类型的消息数量变化
	消息平均大小	Bytes	当前集群消息的平均大小变化
	生产耗时分布	Count	不同生产消息耗时范围的热力分布图
	生产消息大小分布	Count	不同消息大小的热力分布图
消费相关指标	消费消息条数	Count/s	当前集群每秒消费的消息条数。
	消息堆积条数	Count	当前集群消息堆积条数变化。
	消费流量	MBytes/s	当前集群消费消息的总流量变化。
	消费耗时	ms	当前集群的消费者在开始处理消息到处理完成的耗时，仅 5.x 客户端能采集到该指标

分类	指标	单位	含义
	已就绪消息数	Count	状态为 Ready 的消息数量，表示在服务端可以被消费者消费的消息数量
	处理中的消息数	Count	状态为 Inflight 的消息数量，表示消息在客户端消费中，客户端未返回消费结果的消息数量
	每秒被保存的死信消息条数	Count	每秒新增的状态为 DLQ 的消息数量，表示消息达最大重试次数后依旧消费失败，但是保存到指定Topic 的消息数
	已就绪消息的排队时间	ms	最早一条已就绪消息的就绪时间和当前时刻的时间差，反映了消费者拉取消息的及时性
	消费处理滞后时间	ms	最早一条未返回响应的消息的就绪时间和当前时刻的时间差，反映了消费者完成消费消息的及时性
	消费耗时分布	Count	不同消费消息耗时范围的热力分布图
	本地缓存队列中的消息条数	Count	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	本地缓存队列中的消息总大小	Bytes	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	本地缓存队列中的平均排队时间	ms	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
报错相关指标	客户端报错次数	Count	客户端请求服务端过程中报错的次数
	Router Not Found	Count	通常为生产消息时对应的资源名称和格式错误
	Subscription Not Exist	Count	通常为消费消息时对应的资源名称和格式错误
	收发消息权限错误	Count	客户端代码中使用的用户没有对应资源的生产/消费权限

Topic

分类	指标	单位	含义
生产相关指标	生产消息条数	Count/s	当前 Topic 每秒生产的消息条数。
	生产流量	MBytes/s	当前 Topic 生产消息的总流量变化。
	生产者数量	Count	当前 Topic 的生产者客户端数量
	生产成功率	%	当前 Topic 消息生产成功率变化，仅 5.x 客户端能采集到该指标
	生产消息平均耗时(发送RT)	ms	当前Topic生产消息平均耗时

分类	指标	单位	含义
	消息平均大小	Bytes	当前 Topic 消息的平均大小变化
	生产耗时分布	Count	不同生产消息耗时范围的热力分布图
	生产消息大小分布	Count	不同消息大小的热力分布图
消费相关指标	消费消息条数	Count/s	当前 Topic 每秒消费的消息条数。
	消息堆积条数	Count	当前 Topic 消息堆积条数变化。
	消费流量	MBytes/s	当前 Topic 消费消息的总流量变化。
	消费者数量	Count	当前 Topic 的消费者客户端数量
	消费耗时	ms	当前 Topic 的消费者在开始处理消息到处理完成的耗时，仅 5.x 客户端能采集到该指标
	已就绪消息数	Count	状态为 Ready 的消息数量，表示在服务端可以被消费者消费的消息数量
	处理中的消息数	Count	状态为 Inflight 的消息数量，表示消息在客户端消费中，客户端未返回消费结果的消息数量
	每秒被保存的死信消息条数	Count	每秒新增的状态为 DLQ 的消息数量，表示消息达最大重试次数后依旧消费失败，但是保存到指定Topic 的消息数
	已就绪消息的排队时间	ms	最早一条已就绪消息的就绪时间和当前时刻的时间差，反映了消费者拉取消息的及时性
	消费处理滞后时间	ms	最早一条未返回响应的消息的就绪时间和当前时刻的时间差，反映了消费者完成消费消息的及时性
	消费耗时分布	Count	不同消费消息耗时范围的热力分布图
	本地缓存队列中的消息条数	Count	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	本地缓存队列中的消息总大小	Bytes	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	本地缓存队列中的平均排队时间	ms	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列

Grouo

分类	指标	单位	含义
消费相关	消费消息条数	Count/s	当前 Group 每秒消费的消息条数。

分类	指标	单位	含义
指标	消息堆积条数	Count	当前 Group 消息堆积条数变化。
	消费流量	MBytes/s	当前 Group 消费消息的总流量变化。
	消费者数量	Count	当前 Group 的消费者客户端数量
	消费耗时	ms	当前 Group 的消费者在开始处理消息到处理完成的耗时，仅 5.x 客户端能采集到该指标
	已就绪消息数	Count	状态为 Ready 的消息数量，表示在服务端可以被消费者消费的消息数量
	处理中的消息数	Count	状态为 Inflight 的消息数量，表示消息在客户端消费中，客户端未返回消费结果的消息数量
	每秒被保存的死信消息条数	Count	每秒新增的状态为 DLQ 的消息数量，表示消息达最大重试次数后依旧消费失败，但是保存到指定Topic 的消息数
	已就绪消息的排队时间	ms	最早一条已就绪消息的就绪时间和当前时刻的时间差，反映了消费者拉取消息的及时性
	消费处理滞后时间	ms	最早一条未返回响应的消息的就绪时间和当前时刻的时间差，反映了消费者完成消费消息的及时性
	消费耗时分布	Count	不同消费消息耗时范围的热力分布图
	本地缓存队列中的消息条数	Count	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	本地缓存队列中的消息总大小	Bytes	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列
	每秒生成的重试消息条数	Count/s	消息被某个客户端消费失败后，转入重试队列的消息条数变化
	本地缓存队列中的平均排队时间	ms	仅展示 5.x 版本的PushConsumer客户端的数据，SimpleConsumer客户端没有缓存队列

4.x 集群

集群

分类	指标	单位	含义
计费相关指标	集群总 TPS	Count/s	当前集群生产和消费消息的 API 调用次数之和（按照计费规则进行折算）。
	生产 TPS	Count/s	当前集群每秒生产消息的 API 调用次数（按照计费规则进行折算）。

分类	指标	单位	含义
	消费 TPS	Count/s	当前集群每秒消费消息的 API 调用次数（按照计费规则进行折算）。
生产相关指标	生产消息条数	Count/s	当前集群每秒生产的消息条数。
	生产流量	MBytes/s	当前集群生产消息的总流量变化。
消费相关指标	消费消息条数	Count/s	当前集群每秒消费的消息条数。
	消息堆积条数	Count	当前集群消息堆积条数变化。
	消费流量	MBytes/s	当前集群消费消息的总流量变化。
	重试消息堆积数量	Count	服务端中堆积的等待重试的消息数量。

Topic

分类	指标	单位	含义
计费相关指标	集群总 TPS	Count/s	当前集群生产和消费消息的 API 调用次数之和（按照计费规则进行折算）。
	生产 TPS	Count/s	当前集群每秒生产消息的 API 调用次数（按照计费规则进行折算）。
	消费 TPS	Count/s	当前集群每秒消费消息的 API 调用次数（按照计费规则进行折算）。
生产相关指标	生产消息条数	Count/s	当前集群每秒生产的消息条数。
	生产流量	MBytes/s	当前集群生产消息的总流量变化。
消费相关指标	消费消息条数	Count/s	当前集群每秒消费的消息条数。
	消息堆积条数	Count	当前集群消息堆积条数变化。
	消费流量	MBytes/s	当前集群消费消息的总流量变化。

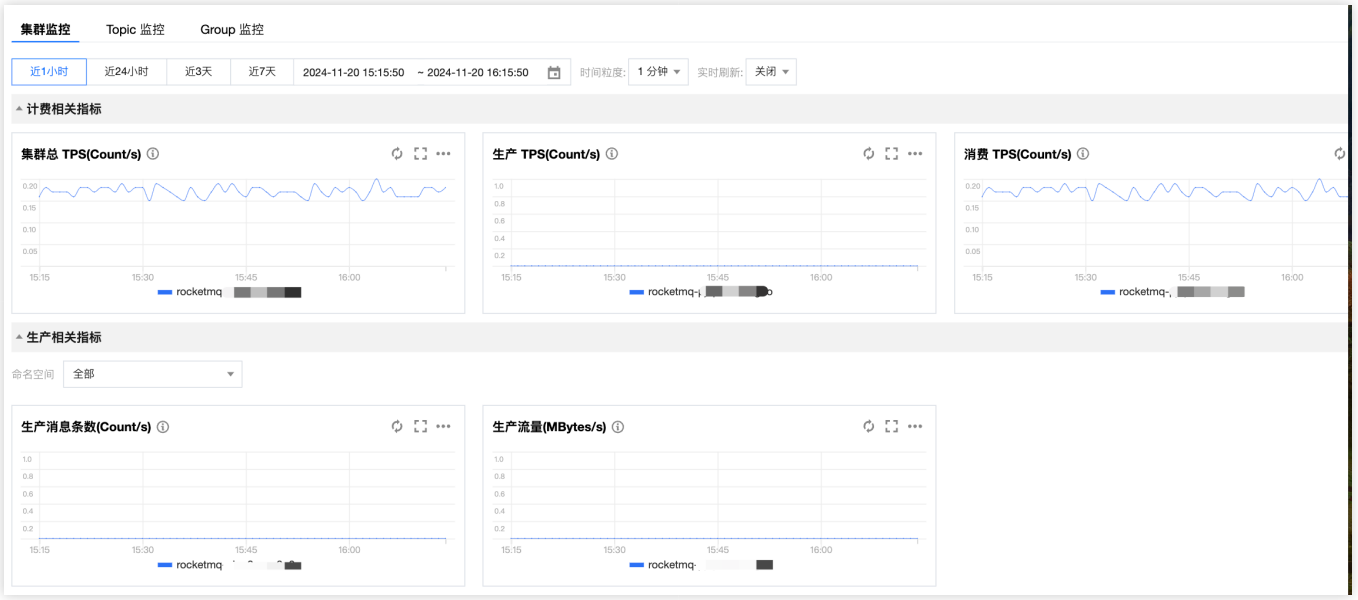
Group

监控指标	单位	说明
消费消息条数	Count/s	该 Group 每秒消费的消息条数。

监控指标	单位	说明
消息堆积条数	Count	该 Group 消息堆积条数变化。
消费流量	MBytes/s	该 Group 消费消息的总流量变化。
在线消费者数	Count	该 Group 中在线的消费者总数。
重试消息堆积数量	Count	服务端中堆积的等待重试的消息数量。

查看监控数据

1. 登录 TDMQ RocketMQ 版控制台。
2. 在左侧导航栏选择监控大盘，选择要查看的资源页签，设置好时间范围后，查看对应的监控数据。



配置告警规则



1. 在监控大盘页面，点击右上角的 指标，跳转至告警配置页面。
2. 单击新建，配置告警策略。
    - 策略类型：选择消息队列 TDMQ/RocketMQ。
    - 告警对象：选择需要配置告警策略的 RocketMQ 资源。
    - 触发条件：支持配置模板或者手动配置，默认选择手动配置。
        - 手动配置
            - 指标：例如"消息消费速率"，选择统计粒度为1分钟，则在1分钟内，消息消费速率连续 N 个数据

点超过阈值，就会触发告警。

- 告警频次：例如“每30分钟告警一次”，指每30分钟内，连续多个统计周期指标都超过了阈值，如果有一次告警，30分钟内就不会再次进行告警，直到下一个30分钟，如果指标依然超过阈值，才会再次告警。

- 选择模板

a. 勾选选择模板后，单击新建触发条件模板，跳转到设置触发条件模板页面。

b. 在左上角单击新建触发条件模板，在新建模板页面，配置告警策略。

- 策略类型：选择 消息队列 TDMQ/RocketMQ。
- 触发条件：根据我们提供的告警策略建议或者您实际的业务需求，设置好告警策略。

c. 确认无误后，单击保存，返回新建告警策略页，单击刷新，就会出现刚配置的告警策略模板。

3. 选择通知模板，您可以选择系统预设的通知模板，预设模板的告警对象通常为主账号负责人。如您需要通知到实例负责人或其他人员，也可以单击新建模板来新建通知模板，设置告警接收对象和接收渠道。
4. 单击完成。

消息查询
查询普通消息

当一条消息从生产者发送到 TDMQ RocketMQ 版服务端，再由消费者进行消费，TDMQ RocketMQ 版会完整记录这条消息中间的流转过程，并以消息轨迹的形式呈现在控制台。
消息轨迹记录了消息从生产端到 TDMQ RocketMQ 版服务端，最后到消费端的整个过程，包括各阶段的时间（精确到微秒）、执行结果、生产者 IP、消费者 IP 等。

- 如果您使用的是 5.0 及以上版本的 gRPC 客户端进行消息的生产和消费，则无需在客户端另行开启轨迹开关。
- 如果您使用的是 4.x 版本的客户端，或者 5.0 以上版本的 Remoting 客户端，则需要在客户端来设置开启消息轨迹功能，具体设置示例如下：
  更多关于客户端的说明请参见 社区客户端说明。
  生产者设置

```
DefaultMQProducer producer = new DefaultMQProducer(namespace, groupName,
  // ACL权限
  new AclClientRPCHook(new SessionCredentials(AK, SK)), true, null);
```

Push 消费者设置

```
// 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(NAMESPACE,groupName,
    new AclClientRPCHook(new SessionCredentials(AK, SK)),
    new AllocateMessageQueueAveragely(), true, null);
```

Pull 消费者设置

```
DefaultLitePullConsumer pullConsumer = new DefaultLitePullConsumer(NAMESPACE,groupName,
    new AclClientRPCHook(new SessionCredentials(AK, SK)));
// 设置NameServer的地址
pullConsumer.setNamesrvAddr(NAMESERVER);
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_LAST_OFFSET);
pullConsumer.setAutoCommit(false);
pullConsumer.setEnableMsgTrace(true);
pullConsumer.setCustomizedTraceTopic(null);
```

Spring Boot Starter 接入（2.2.2版本及以上运行）

```
package com.lazycece.sbac.rocketmq.messagemodel;

import lombok.extern.slf4j.Slf4j;
import org.apache.rocketmq.spring.annotation.MessageModel;
```

```
import org.apache.rocketmq.spring.annotation.RocketMQMessageListener;
import org.apache.rocketmq.spring.core.RocketMQListener;
import org.springframework.stereotype.Component;

/**
 * @author lazycece
 * @date 2019/8/21
 */
@Slf4j
@Component
public class MessageModelConsumer {

@Component
    @RocketMQMessageListener(
            topic = "topic-message-model",
            consumerGroup = "message-model-consumer-group",
            enableMsgTrace = true,
            messageModel = MessageModel.CLUSTERING)
    public class ConsumerOne implements RocketMQListener<String> {
        @Override
        public void onMessage(String message) {
            log.info("ConsumerOne: {}", message);
        }
    }

}
```

操作场景

当您需要排查以下问题时，就可以使用 TDMQ RocketMQ 版控制台的消息查询功能，按照时间维度或者根据日志中查到的消息 ID 、消息 Key 或者 消息 Tag，来查看具体某条消息的消息内容、消息参数和消息轨迹。

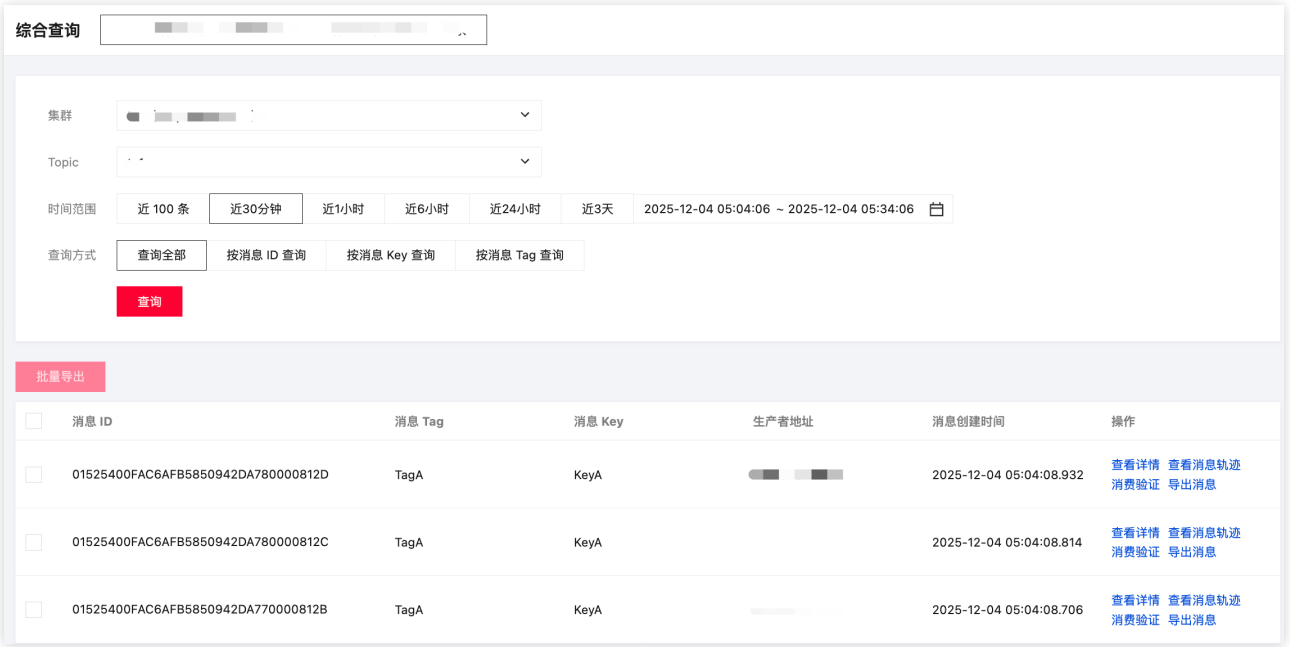
- 查看某条消息的具体内容，具体参数。
- 查看消息由哪个生产 IP 发送，是否发送成功，消息到服务端的具体时间。
- 查看消息是否已持久化。
- 查看消费由哪些消费者消费了，是否消费成功，消息确认消费的具体时间。
- 需要做分布式系统的性能分析，查看 MQ 对相关消息处理的时延。

查询消息

1. 登录 RocketMQ 控制台，在左侧导航栏单击消息查询 > 综合查询。
2. 在综合查询页面，选择好地域后根据页面提示输入查询条件。
    - 集群：选择需要查询的 Topic 所在的集群。
    - Topic：选择需要查询的 Topic。
    - 时间范围：选择需要查询的时间范围，支持近100条（默认按时间顺序展示最近的 100 条消息），近30分钟，近1小时，近6小时，近24小时，近3天和自定义时间范围。

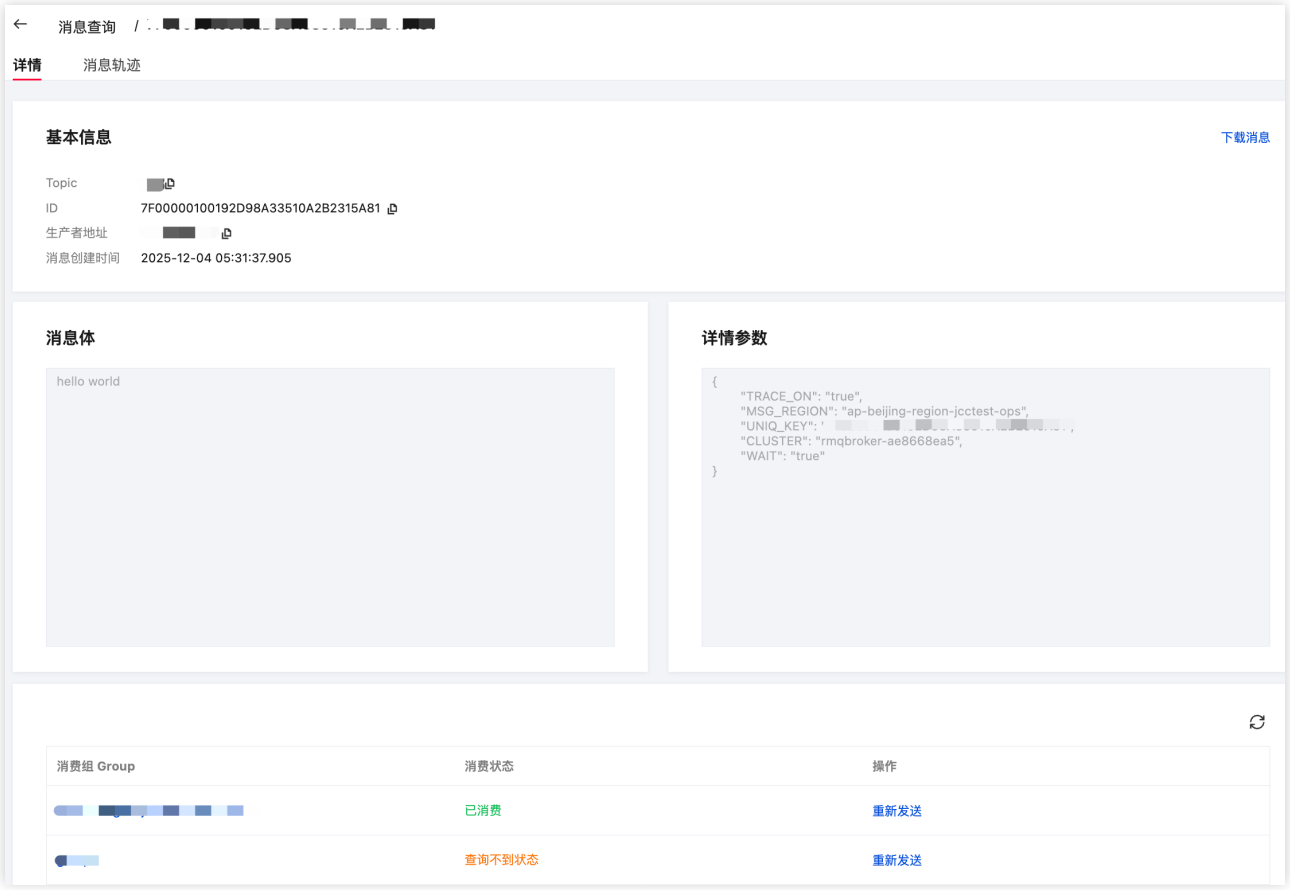
- 查询方式：消息查询功能支持以下查询方式。
  - **查询全部：**该方式适合查询全部消息。
  - **按消息 ID 查询：**该方式属于精确查询、速度快、精确匹配。
  - **按消息 Key 查询：**该方式属于模糊查询，适用于您没有记录消息 ID 但是设置了消息 Key 的场景。查询结果最多展示64条，可以通过细化查询时间来精确定位需要查询的消息。
  - **按消息 Tag 查询：**该方式属于模糊查询，适用于您没有记录消息 ID 但是设置了消息 Tag 的场景。查询结果最多展示64条，可以通过细化查询时间来精确定位需要查询的消息。

3. 单击查询，下方列表会展示所有查询到的结果并分页展示。

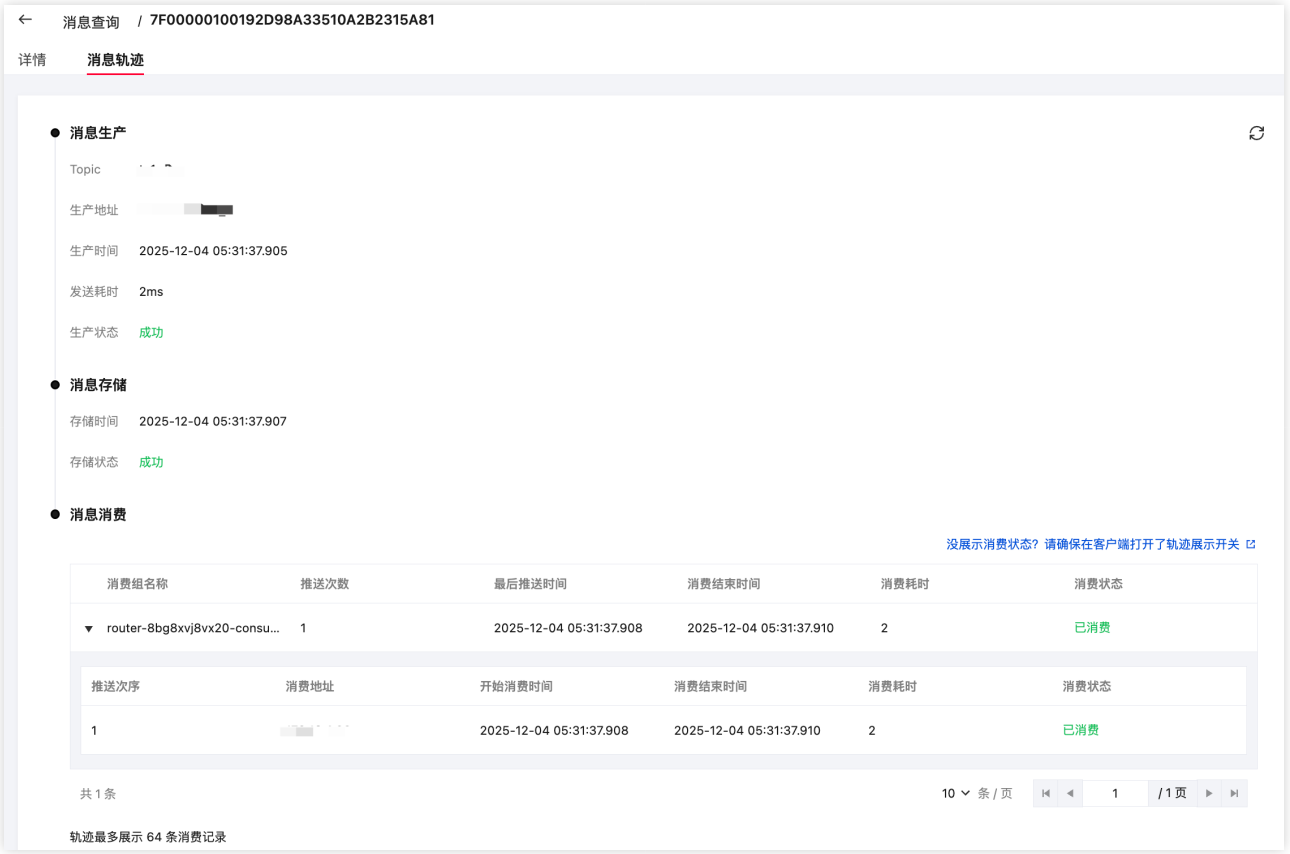


4. 找到您希望查看内容或参数的消息，单击操作列的查看详情，即可查看消息的基本信息、内容（消息体）、详情参数和消费状态。

   在消费状态模块，您可以查看到消费该消息的 Group 以及消费状态，同时还可以在操作栏进行如下操作：
   - 重新发送：将消息重新发送到指定的客户端，如消息已消费成功，重新发送该消息可能导致消费重复。
   - 异常诊断：若消费异常，可以查看异常诊断信息。



5. 单击操作列的查看消息轨迹，或者在详情页单击 Tab 栏的消息轨迹，即可查看该消息的消息轨迹（详细说明请参见 消息轨迹与说明）。



消费验证

在查询到某条消息后，您可以单击操作列的消费验证将该条消息发送到指定的客户端来验证该消息。该功能可能会导致消息重复。

> 说明：
>
> - 消费验证功能仅用于验证客户端的消费逻辑是否正常，并不会影响正常的收消息流程，因此消息的消费状态等信息在消费验证后并不会改变。
> - 消费验证功能目前暂不支持发送到 Python 和 C++ 客户端。



导出消息

在查询到某条消息后，您可以单击操作列的导出消息将该条消息的消息体，消息 Tag，消息 Key，消息生产时间和消费属性等信息。
您也可以勾选要导出的消息后，单击左上角的**批量导出，**实现批量导出勾选的消息。

下载消息

在消息详情页，您可以单击右上方的下载消息按钮，可以将单条消息的消息体和消息头保存到本地。

查看死信队列

操作场景

死信队列是一种特殊的消息队列，用于集中处理无法被正常消费的消息的队列。当消息在达到一定重试次数后仍未能被正常消费，TDMQ RocketMQ 版会判定这条消息在当前情况下无法被消费，将其投递至死信队列。
实际场景中，消息可能会由于持续一段时间的服务宕机，网络断连而无法被消费。这种场景下，消息不会被立刻丢弃，死信队列会对这种消息进行较为长期的持久化，用户可以在找到对应解决方案后，创建消费者订阅死信队列来完成对当时无法处理消息的处理。

查询限制

消息查询最多可以查询近3天的消息。

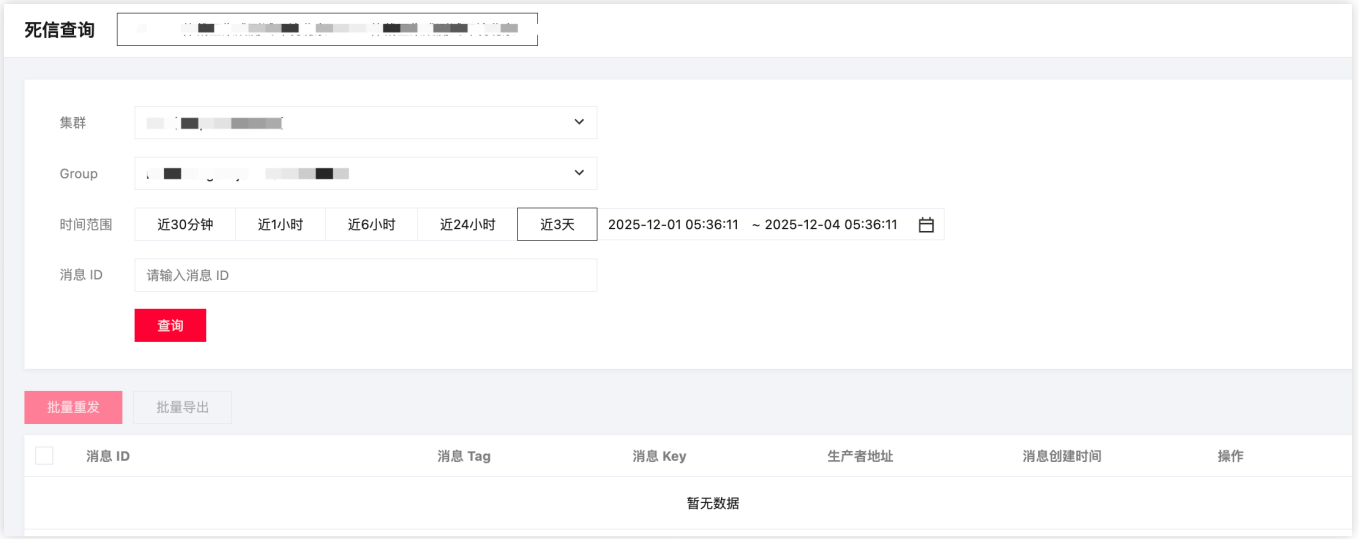
特性说明

- 当消息被投递到死信队列后，消息不会再被消费者正常消费。消息查询最多可以查询近3天的消息，请尽量在死信消息产生的3天内进行处理，否则消息可能会被删除。
- 一个死信队列包含了对应的一个 Group 中所有 Topic 产生的所有死信消息。如果一个Group 中没有产生死信消息，则不会为其创建死信队列，也查询不到死信消息。

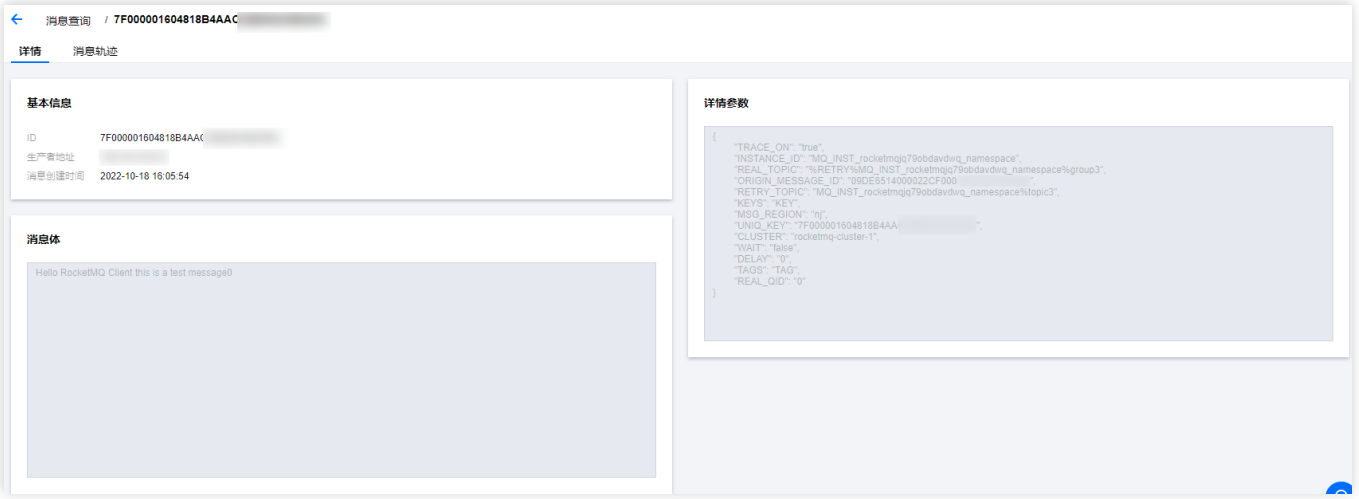
操作步骤

1. 登录 TDMQ RocketMQ 控制台，在左侧导航栏单击消息查询。
2. 在消息查询页面，选择好地域后根据页面提示输入查询条件。
    - 时间范围：选择需要查询的时间范围，支持近30分钟，近1小时，近6小时，近24小时，近3天和自定义时间范围。
    - 集群：选择需要查询的死信消息所在的集群。
    - 命名空间：选择需要查询的死信消息所在的命名空间。（仅 4.x 集群需选择）
    - Group：选择需要查询的死信消息所在的Group。
    - 消息 ID：非必填。
        - 不填写消息 ID：属于模糊查询。根据 Group ID 和死信消息产生的时间范围，批量查询该 Group ID在某段时间内产生的所有死信消息。
        - 填写消息 ID：属于精确查询。根据 Group ID与 Message ID 精确定位到任意一条消息。

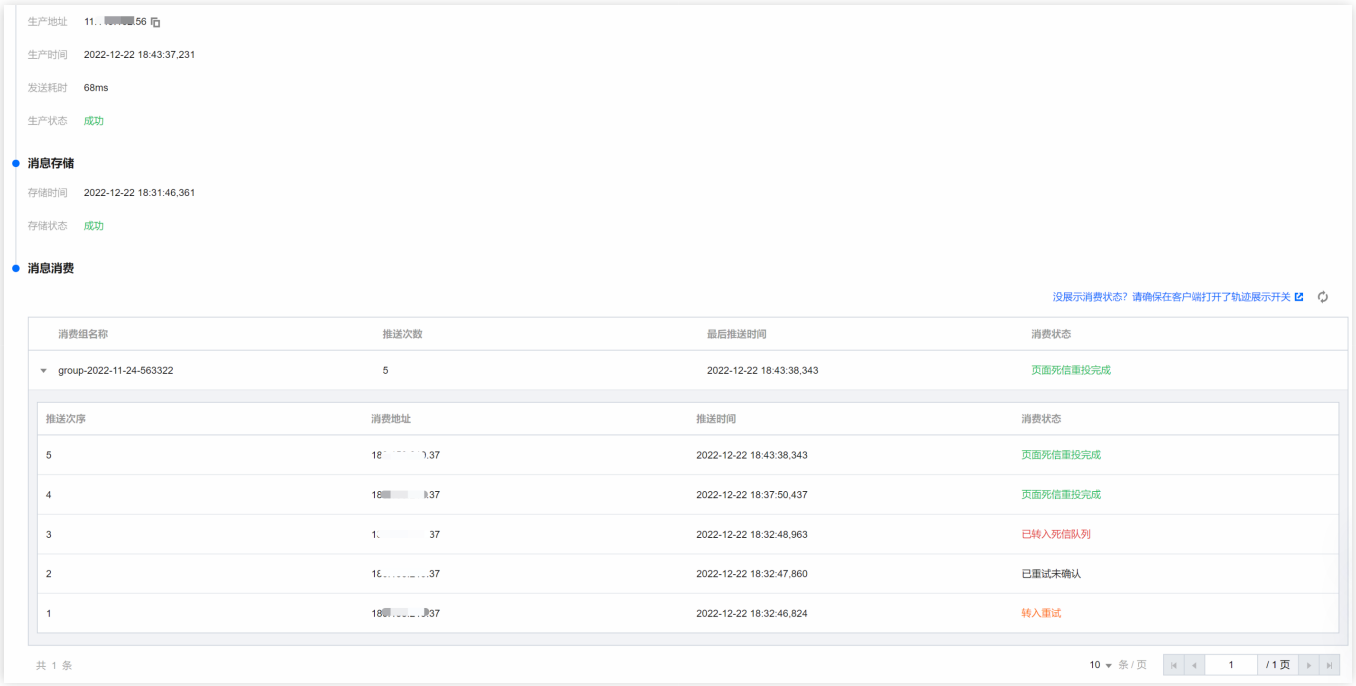
3. 单击查询，下方列表会展示所有查询到的结果并分页展示。



4. 您可以勾选多条死信消息后单击左上角的批量重发将死信消息批量重新发送到原队列的重试队列，也可以单击某条消息操作列的重发消息将单条死信消息重新发送。死信消息在被重新发送后，会被投递到原队列的重试队列，不会在死信队列中被立即删除，在达到消息生命周期（3天）后才会被删除。
5. 找到您希望查看内容或参数的消息，单击操作列的查看详情，即可查看消息的基本信息、内容（消息体）以及参数。



6. 单击操作列的查看消息轨迹，或者在详情页单击 Tab 栏的消息轨迹，即可查看该消息的消息轨迹（详细说明请参见 消息轨迹查询结果说明）。可以看到当死信消息被重新投递后，消费状态变成页面死信重投完成。


生产地址 11..[redacted].56
生产时间 2022-12-22 18:43:37,231
发送耗时 68ms
生产状态 成功
消息存储
存储时间 2022-12-22 18:31:46,361
存储状态 成功
消息消费
没展示消费状态？请确保在客户端打开了轨迹展示开关
消费组名称 推送次数 最后推送时间 消费状态
group-2022-11-24-563322 5 2022-12-22 18:43:38,343 页面死信重投完成
推送次序 消费地址 推送时间 消费状态
5 18[redacted].37 2022-12-22 18:43:38,343 页面死信重投完成
4 18[redacted]37 2022-12-22 18:37:50,437 页面死信重投完成
3 1[redacted]37 2022-12-22 18:32:48,963 已转入死信队列
2 18[redacted].37 2022-12-22 18:32:47,860 已重试未确认
1 18[redacted]37 2022-12-22 18:32:46,824 转入重试
共 1 条
10 条 / 页
1
/ 1 页

查询重试消息

为了给业务处理失败，消息消费失败的情况兜底，保证消息生命周期的完整性，TDMQ RocketMQ 版支持消费失败后重试的能力。

- 对于 RocketMQ 4.x 集群，消息的重试次数以您在客户端内设置消息重试次数为准。
- 对于 RocketMQ 5.x 集群，您在创建 Group 时可以设置消息的重试次数，如果您使用的是 5.x 客户端，则重试次数以您在服务端设定的为准；如果您使用的是 4.x 客户端，则重试次数依旧以客户端内设置消息重试次数为准。

操作场景

当您需要查看某个 Topic 下是否有重试消息时，您可以在控制台查询消息，并且可以展开查看消息每次重试的时间和生产者地址等信息，并且支持导出消息和查看消息的详细内容。

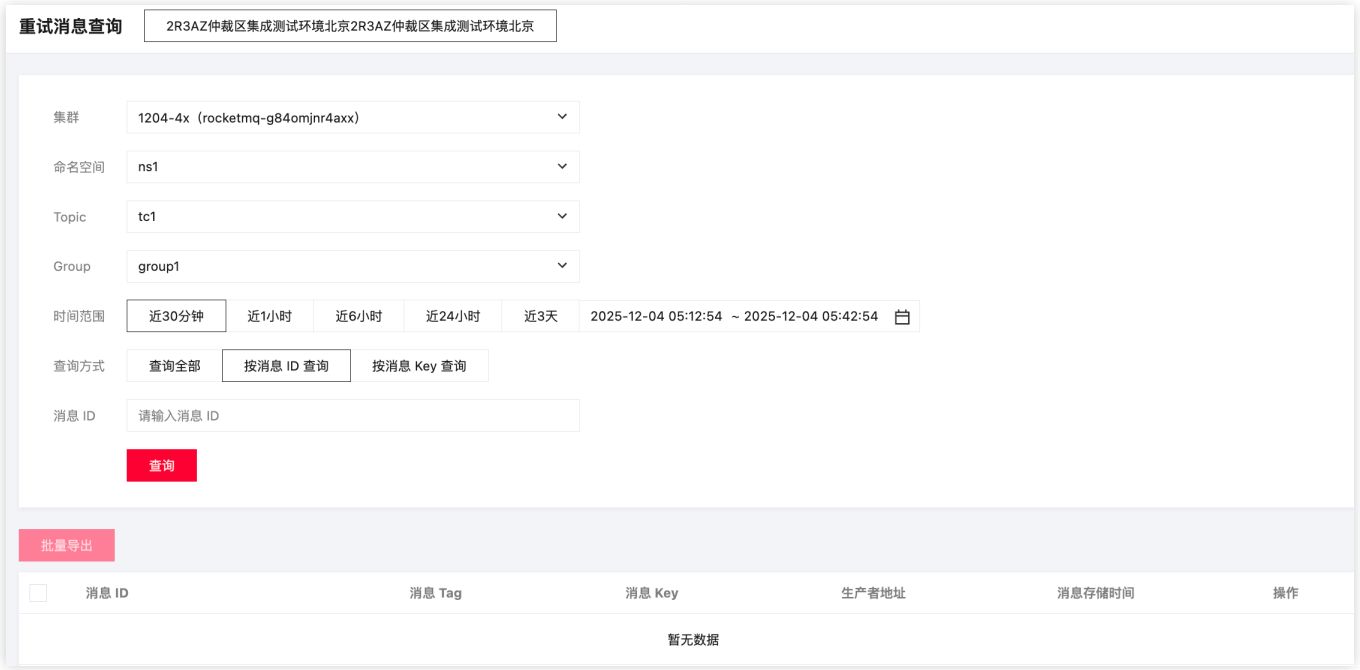
操作步骤

1. 登录 RocketMQ 控制台，在左侧导航栏单击 重试消息查询。

2. 在消息查询页面，选择好地域后根据页面提示输入查询条件。
    - 集群：选择需要查询的 Topic 所在的集群。
    - Topic：选择需要查询的 Topic。
    - Group：如果查询的集群为 5.x 集群，则需要选择该 Topic 下订阅的具体 Group。4.x 集群无需填写。
    - 时间范围：选择需要查询的时间范围，支持近30分钟，近1小时，近6小时，近24小时，近3天和自定义时间范围。
    - 查询方式：消息查询功能支持以下查询方式。
        - **查询全部：**该方式适合在对于重试消息的信息不明确的情况下使用，用于查询当前 Topic 下的全部重试消息。
        - **按消息 ID 查询：**该方式属于精确查询、速度快、精确匹配。
        - **按消息 Key 查询：**该方式属于模糊查询，适用于您没有记录消息 ID 但是设置了消息 Key 的场景。

        > 说明：
        >
        > 为了保证查询速度，当您选择 “查询全部” 时，服务端会按照时间先后查询最近的消息，但由于查询时间和展示的限制，可能无法快速定位到您需要排查的消息。建议您使用更加明确的搜索条件，如 消息 ID 和消息 Key。

3. 单击查询，下方列表会展示所有查询到的结果并分页展示。



4. 查询完成后，您可以点击单条消息，查看当前消息的重试情况，如重试的次数和生产者地址等信息。您也可以单击操作栏的其他操作选项查看消息详情和轨迹信息。

查询定时与延时消息

操作场景

定时/延时消息是 RocketMQ 中比较重要的消息类型。

- 定时消息：消息在发送至服务端后，实际业务并不希望消费端马上收到这条消息，而是推迟到某个时间点被消费，这类消息统称为定时消息。
- 延时消息：消息在发送至服务端后，实际业务并不希望消费端马上收到这条消息，而是推迟一段时间后再被消费，这类消息统称为延时消息。
  实际上，延时消息可以看成是定时消息的一种特殊用法，其实现的最终效果和定时消息是一致的。

查询限制

由于 RocketMQ 给消息设置有消息保留时间，即消费完成的消息会在服务端保留一段时间（体验版和基础版的时间为 3 天，专业版和铂金版支持 Topic 维度的时间调整，支持7天及更长时间），在消息的保留时间内的延时和定时消息均可被查询到。
例如：假设某集群于 1月1日1时发送了一条延时10天的延时消息，集群的保留时间为三天，则1月11日1时，消息发送给消费者，假定消费者消费成功，则在1月14日1时前，均可以在当前页面查询到本条消息。

> 说明：
>
> 当前仅 5.x 集群支持查询定时和延时消息

操作步骤

1. 登录 TDMQ RocketMQ 控制台，在左侧导航栏单击延时消息查询。
2. 在消息查询页面，选择好地域后根据页面提示输入查询条件。
   - 集群：选择需要查询的延时消息所在的集群。
   - 消息 ID：填写 Message ID 精确定位到任意一条消息。
3. 单击查询，下方列表会展示所有查询到的结果并分页展示。



4. 完成查询后，可以查看消息的创建和生产时间，和其他的消息查询结果类似，可以对消息进行查看轨迹和导出等操作。
5. 单击操作列的查看消息轨迹，或者在详情页单击 Tab 栏的消息轨迹，即可查看该消息的消息轨迹（详细说明请参见消息轨迹与说明）。在“消息消费”部分可以查看到对应消息的定时到期发送消息的时间和消费状态。

消息轨迹与说明

消息轨迹记录了消息从生产端到 TDMQ RocketMQ 版服务端，最后到消费端的整个过程，包括各阶段的时间（精确到微秒）、执行结果、生产者 IP、消费者 IP 等。

前提条件

- 您已经部署好生产端和消费端服务，并在3天内有消息生产和消费。
- 如果您使用的是 5.0 及以上版本的 gRPC 客户端进行消息的生产和消费，则无需在客户端另行开启轨迹开关。
- 如果您使用的是 4.x 版本的客户端，或者 5.0 以上版本的 Remoting 客户端，则需要在客户端来设置开启消息轨迹功能，具体设置示例如下：
- 更多关于客户端的说明请参见 社区客户端说明。
  生产者设置

```
DefaultMQProducer producer = new DefaultMQProducer(namespace, groupName,
    // ACL权限
    new AclClientRPCHook(new SessionCredentials(AK, SK)), true, null);
```

Push 消费者设置

```
// 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(NAMESPACE,group
Name,
    new AclClientRPCHook(new SessionCredentials(AK, SK)),
    new AllocateMessageQueueAveragely(), true, null);
```

Pull 消费者设置

```
DefaultLitePullConsumer pullConsumer = new DefaultLitePullConsumer(NAMESPACE,groupNam
e,
    new AclClientRPCHook(new SessionCredentials(AK, SK)));
// 设置NameServer的地址
pullConsumer.setNamesrvAddr(NAMESERVER);
```

```
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_LAST_OFFSET);
pullConsumer.setAutoCommit(false);
pullConsumer.setEnableMsgTrace(true);
pullConsumer.setCustomizedTraceTopic(null);
```

Spring Boot Starter 接入（2.2.2版本及以上运行）

```
DefaultLitePullConsumer pullConsumer = new DefaultLitePullConsumer(NAMESPACE,groupNam
e,
    new AclClientRPCHook(new SessionCredentials(AK, SK)));
// 设置NameServer的地址
pullConsumer.setNamesrvAddr(NAMESERVER);
pullConsumer.setConsumeFromWhere(ConsumeFromWhere.CONSUME_FROM_LAST_OFFSET);
pullConsumer.setAutoCommit(false);
pullConsumer.setEnableMsgTrace(true);
pullConsumer.setCustomizedTraceTopic(null);
```

消息轨迹查询结果说明

消息轨迹查询出来的结果分为三段：消息生产、消息存储和消息消费。

消息生产

参数	说明
生产地址	对应生产者的地址以及端口。
生产时间	TDMQ RocketMQ 版服务端确认接收到消息的时间，精确到毫秒。
发送耗时	消息从生产端发送到 TDMQ RocketMQ 版服务端的时间消耗，精确到微秒。
生产状态	表示消息生产成功或失败，如果状态为失败一般是消息在发送过程中遇到了头部数据部分丢失，以上几个字段可能会为空值。

消息存储

参数	说明

参数	说明
存储时间	消息被持久化的时间。
存储状态	表示消息持久化成功或失败，如果状态为失败则表明消息未落盘成功，可能由于底层磁盘损坏或无多余容量导致，遇见此类情况需联系技术支持咨询。

消息消费

消息消费是以列表形式呈现的，TDMQ RocketMQ 版支持集群消费和广播消费两种消费模式。
列表中展示的信息说明：

参数	说明
消费组名称	消费组的名称。
消费模式	消费组的消费模式，支持集群消费和广播消费两种模式。
推送次数	TDMQ RocketMQ 版服务端向消费者投递该消息的次数。
最后推送时间	TDMQ RocketMQ 版服务端最后一次向消费者投递该消息的时间。
消费结束时间	TDMQ RocketMQ 版服务端接收到消费者回复确认信息的时间。
消费耗时	TDMQ RocketMQ 版服务端第一次向消费者投递该消息到接收到消费者回复确认信息的时间。
消费状态	- 已推送未确认：TDMQ RocketMQ 版服务端已向消费者投递消息，未接收到消费者回复的确认消息。 - 已消费：消费者回复确认信息（ACK）到 TDMQ RocketMQ 版服务端，服务端接收到确认信息。 - 转入重试：已超时，服务端仍未接收到确认信息，将再次投递消息。 - 已重试未确认：TDMQ RocketMQ 版服务端已再次向消费者投递消息，未接收到消费者回复的确认消息。 - 已转入死信队列：消息经过一定重试次数后仍未能被正常消费，被投递至死信队列。 说明：如果消费模式为广播模式，则消费状态只有已推送一种。

单击订阅名称左方的右三角，查看服务端每次推送消息的详情。

参数	说明
推送次序	TDMQ RocketMQ 版服务端第几次向消费者投递该消息。
消费地址	收到消息的消费者地址及端口。

参数	说明
开始消费时间	TDMQ RocketMQ 版服务端向消费者投递该消息的时间。
消费结束时间	TDMQ RocketMQ 版服务端接收到消费者回复确认信息的时间。
消费耗时	TDMQ RocketMQ 版服务端向消费者投递该消息到接收到消费者回复确认信息的时间。
消费状态	- 已推送未确认：TDMQ RocketMQ 版服务端已向消费者投递消息，未接收到消费者回复的确认消息。 - 已消费：消费者回复确认信息（ACK）到 TDMQ RocketMQ 版服务端，服务端接收到确认信息。 - 转入重试：已超时，服务端仍未接收到确认信息，将再次投递消息。 - 已重试未确认：TDMQ RocketMQ 版服务端已再次向消费者投递消息，未接收到消费者回复的确认消息。 - 已转入死信队列：消息经过一定重试次数后仍未能被正常消费，被投递至死信队列。 - 页面死信重投完成：在死信队列重发页面上，用户已经将死信消息重新投递到原队列的重试队列中。

导入元数据

客户在使用 TDMQ RocketMQ 版时，通常会面临着存量业务的切换，例如从自建的 RocketMQ 或者其他平台的托管版 RocketMQ 迁移到 TDMQ-RocketMQ 版上。
客户可以先将源集群的元数据的导出，再导入到TDMQ-RocketMQ上，后续 依次对消费者和生产者切换接入地址。

前提条件

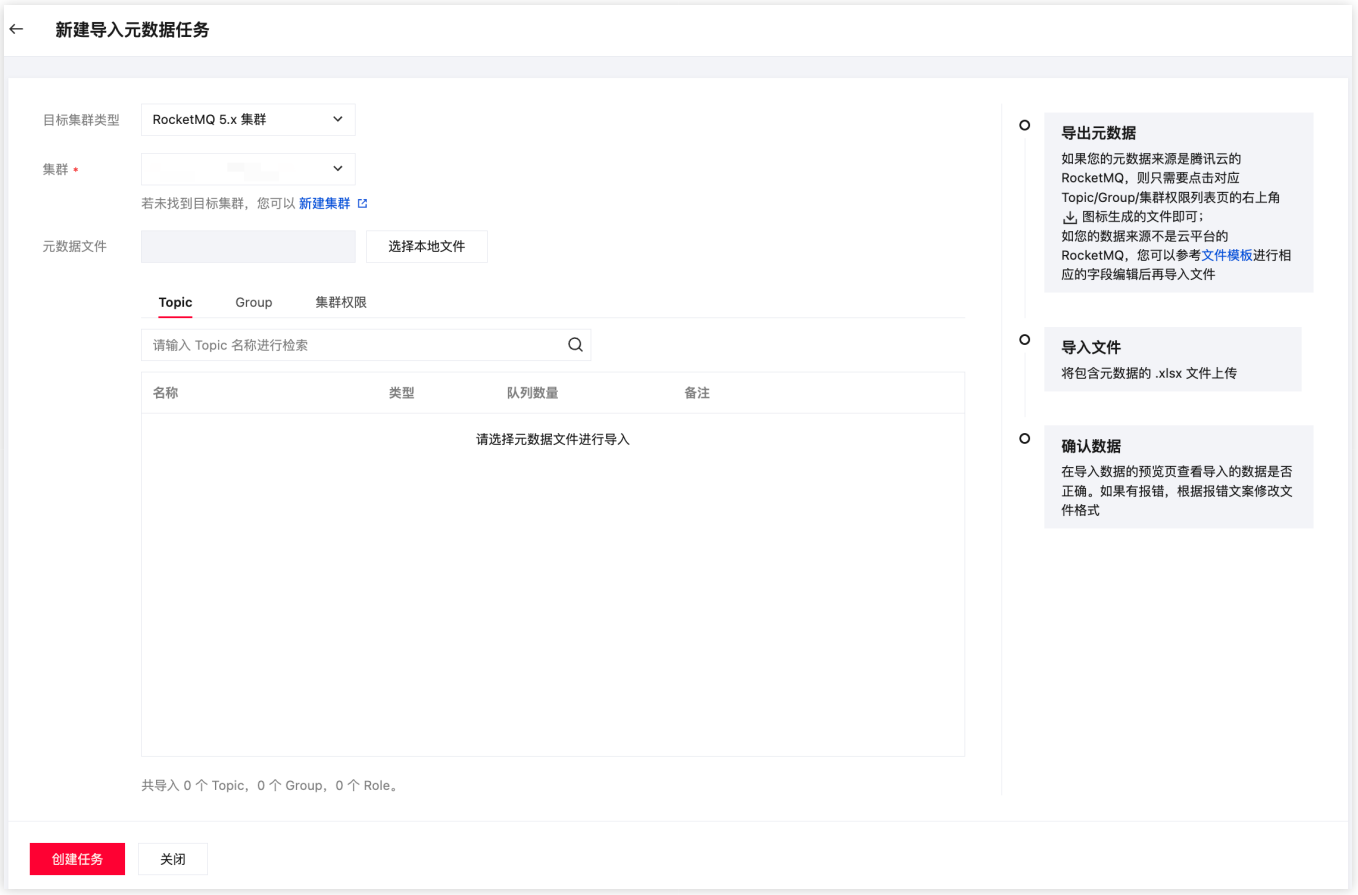
导出元数据

- 如果您的元数据来源是云平台的 RocketMQ，则只需要点击Topic/Group 列表页的右上角导入图标生成的文件即可；
- 如您的数据来源不是云平台的 RocketMQ，您可以参考文件模板进行相应的字段编辑后再导入文件。

操作步骤

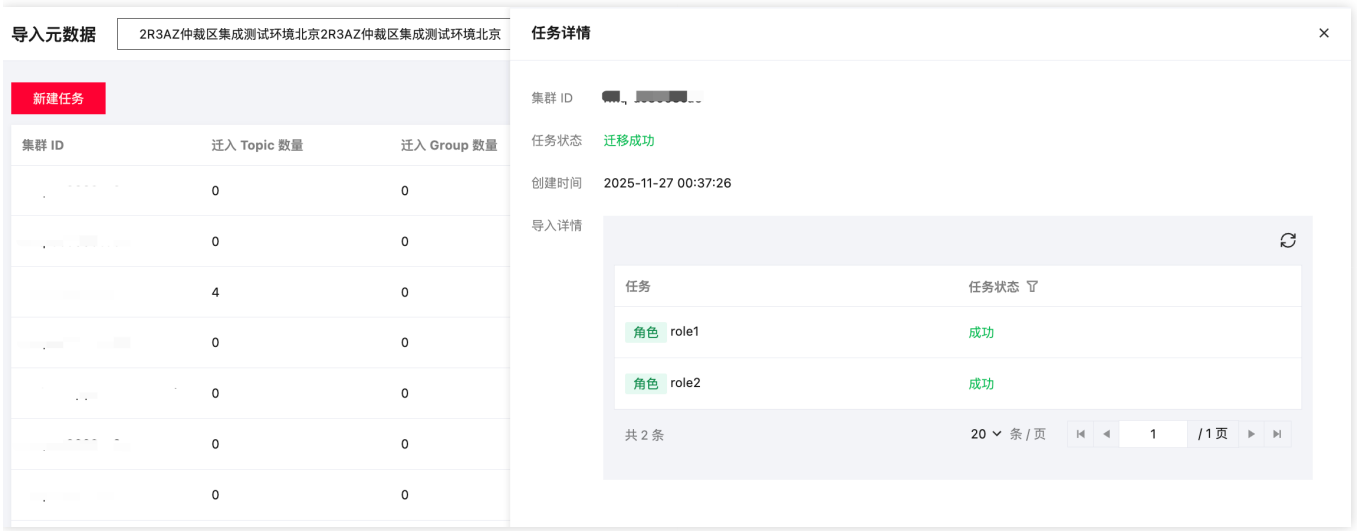
导入元数据

1. 登录 TDMQ RocketMQ 版控制台。
2. 在左侧导航栏选择导入元数据，单击新建任务，选择要导入的数据的集群并上传元数据文件。
3. 系统会自动扫描出元数据信息，即 Topic、Group、集群权限等。您可以在页面上确认相关信息，如果有报错，根据报错文案修改文件格式，确认无误后，单击新建任务，即可开始导入数据。



查看任务详情

任务创建完成后，您可以在操作栏的查看详情查看导入进展和任务状态。



单元管理

操作场景

在单元化场景下，TDMQ RocketMQ 可以和 单元化管理平台（TCUA）一同部署，构建客户的单元化的环境。
用户在 单元化管理平台（TCUA）指定需要支持单元化的 RocketMQ 集群，RocketMQ 集群会按照在客户在 单元化管理平台（TCUA）指定的单元架构规划，对指定的集群进行相应的适配。

操作步骤

前提条件

- 客户已经按照 单元化管理平台（TCUA）的文档指引，配置了业务单元并推送了架构规划。



- 架构规划推送完成后，RocketMQ 会收到架构变动的信息，会根据推送的单元配置信息创建符合要求的集群。

查看单元化集群

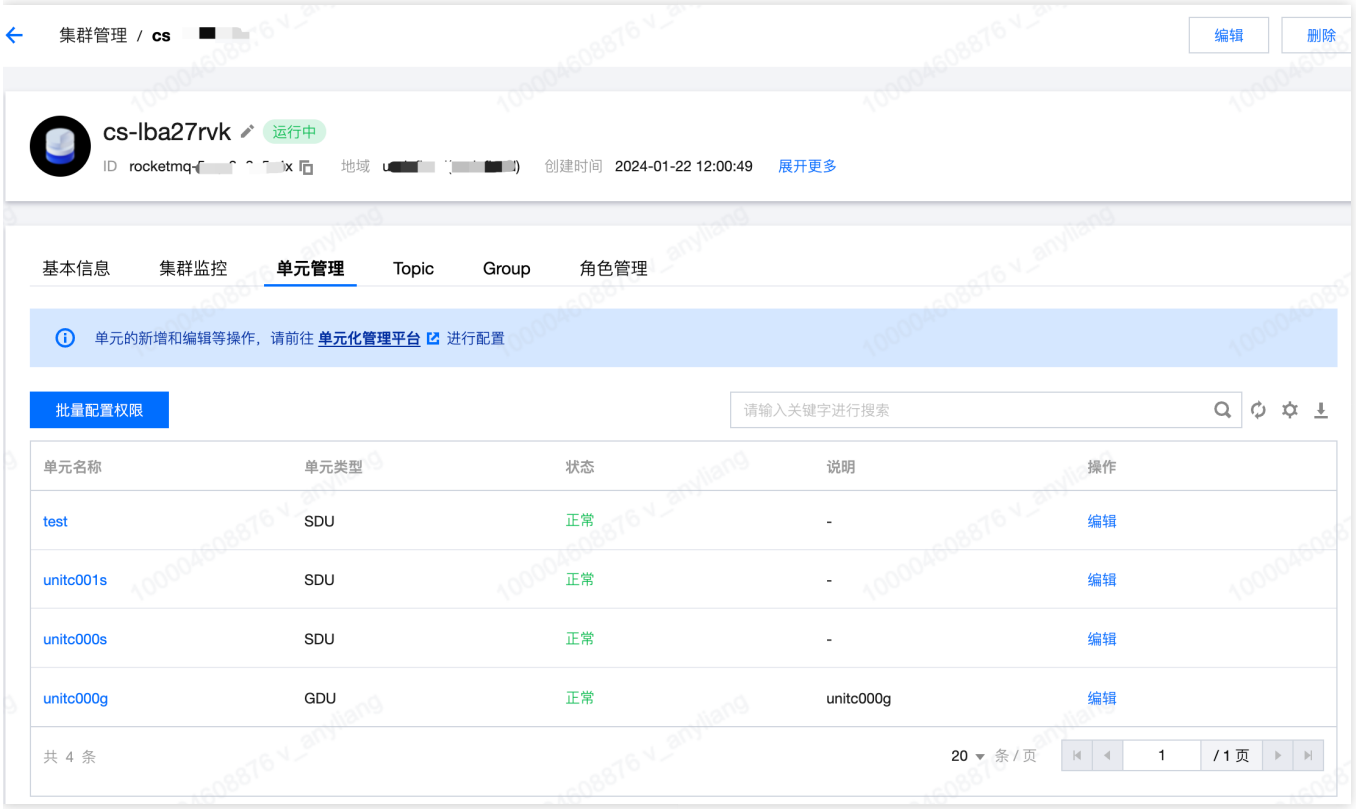
通过点击页面上方的 云产品 > 消息队列 RocketMQ，进入到消息队列 RocketMQ 的控制台。在左侧的 资源管理 > 集群管理 页面可以查看新建的单元化 RocketMQ 集群。您可以通过筛选 是否开启单元化 列的状态筛选已经开启了单元化的集群。
同时，您也可以在集群列表搜索您在 单元化管理平台（TCUA）配置的云空间的 ID和名称，RocketMQ 集群的名称会使用云空间的 ID，RocketMQ 集群的说明使用云空间的名称。



点击集群名进入集群详情后，可以查看集群的基本信息，单元化的集群使用同一个内网接入点（非单元化的集群，不同的命名空间有不同的接入点）。

查看单元

如下图，点击 单元管理 页签，您可以查看此前在 单元化管理平台（TCUA）配置的各个单元的信息和单元类型。



创建角色和配置权限

在 角色管理 页签，您可以为当前集群创建多个角色，后续您将通过这些角色的 AccessKey 和 SecretKey 进行消息的生产和消费。点击 添加角色 按钮，填写角色的名称和说明，为集群创建新的角色。

创建完成角色后，您可以为不同类型的单元配置权限。在 角色管理 页签，点击单元列表左上角的 批量配置权限 按钮，为单

元配置生产和消费权限。
在单元化的集群内，单元权限的配置是以“单元类型”为维度进行配置的，即如果给一个 SDU 类型的单元配置了角色A的生产和消费权限，则角色A即拥有全部 SDU 单元的生产和消费权限。

新建和删除单元

为了保证单元架构的一致性，RocketMQ 控制台不支持单元的新建和删除，您需要前往 单元化管理平台（TCUA）进行架构的调整。

单元化路由

操作场景

在单元化的场景下，TDMQ RocketMQ 支持生产者客户端给位于不同单元的消费者发送消息，同时也支持消费者客户端消费来自不同单元的生产者发送的消息。

操作步骤

前提条件

已经在单元化管理平台上完成了架构规划，并将架构推送到了 RocketMQ 并创建了支持单元化的 RocketMQ 集群。

创建单元化路由规则

在左侧导航栏选择 单元化路由 页签，点击 新建路由规则，按照要求填写以下字段：

- 规则名称：200字符以内，只能包含 中文、数字、字母、“-”和“_”。
- 规则类型：支持集群内单元路由和跨集群路由。
    - 集群内单元路由：仅能选择支持单元化的 RocketMQ 集群。
    - 跨集群单元路由：需同时选择好源集群和目标集群。
- Topic：选择需要跨单元路由规则适用的 Topic；对于支持单元化的集群，为了保证各个单元环境的一致性，在任一单元创建了 topic/group 后，会在其他单元创建同样的 topic/group。
- 路由类型：选择不同类型间的单元的路由，当前支持四种（从 SDU 到 SDU，从 SDU 到 GDU，从 GDU 到 SDU 和从GDU 到 GDU）。
- 复制起始位置：支持从最新的位点开始复制或者指定时间点开始复制。
- 是否立即开启任务：如果打开开关，在规则创建完成后就按照当前规则的配置进行跨单元的消息路由。



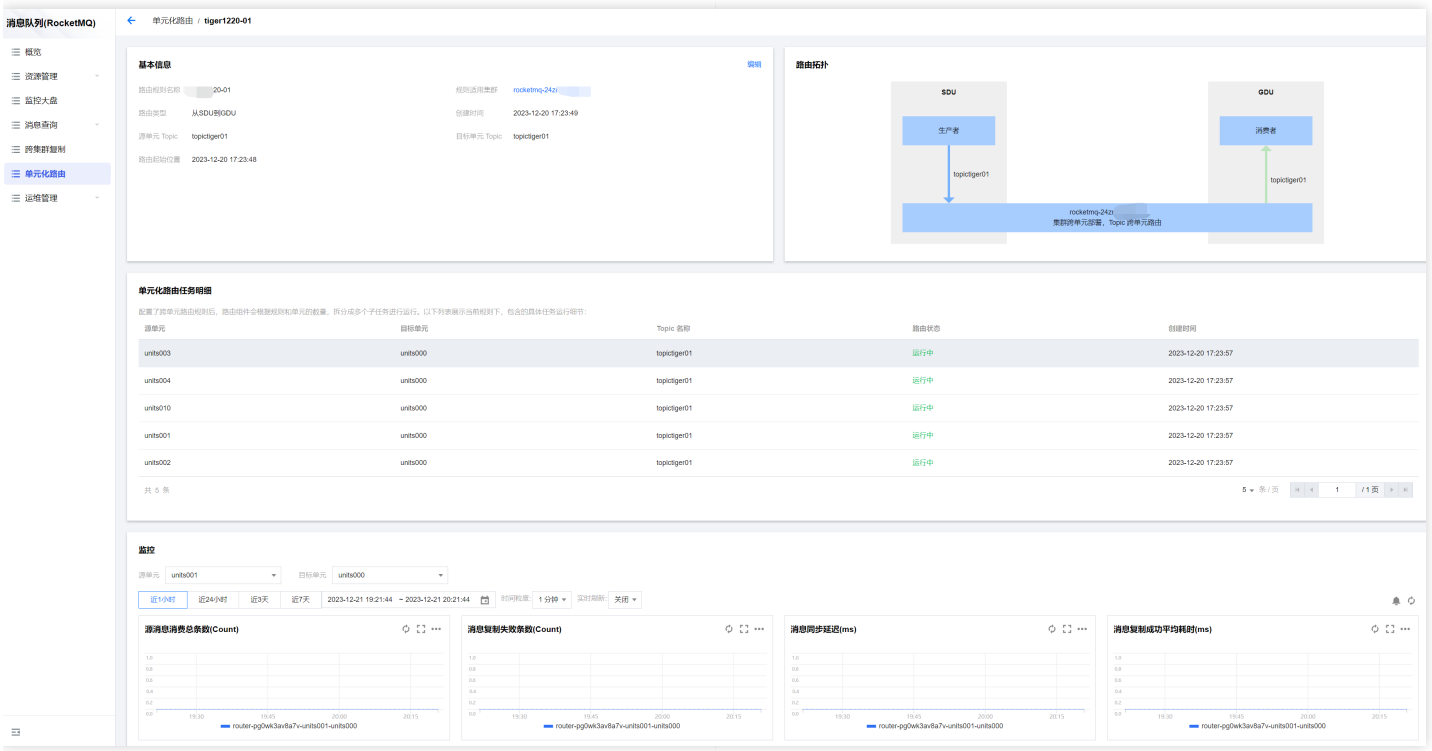
查看规则详情

路由规则列表页

在单元化路由规则创建完成后，您可以在单元化路由的规则列表页看到新增的单元化路由规则和状态。单击操作列的启动/暂停可以快速的开启和暂停单元化路由。
运行中的单元化路由无法修改配置信息，如果要修改单元化路由规则的配置，请先暂停单元化路由后。单击操作栏的编辑，或者进入路由规则详情页，单击"基本信息"右上角的 编辑，修改规则的信息。

路由规则详情页

您可以单击路由规则 ID，进入规则详情页。



在页面的“基本信息”部分会展示当前规则的详细信息，如规则的创建时间，路由的起始时间等等。

在页面的“路由拓扑”部分会展示当前规则的拓扑概览，因为涉及的单元可能会比较多，因此该拓扑图并未展示所有的单元，仅展示单元类型间的消息路由图示。

在页面的“单元化路由任务明细”部分，会详细展示按照当前规则运行的若干子任务。以上图为例，规则的类型为“从 SDU 到 GDU”，在任务明细处，展示了规则在执行时，分为了5个子任务，即从不同的 5 个 SDU 路由到 1 个 GDU 时，同时执行的路由任务。

在页面的“监控”部分，您可以查看当前规则在运行时，不同子任务消息路由情况的实时监控。通过选择切换 源单元 和 目标单元，查看不同子任务的详细数据，如消息路由的条数，消息路由的速率（TPS），消息路由失败条数、消息路由延迟 和 平均路由耗时等指标。

消息跨地域复制

操作场景

TDMQ RocketMQ 支持客户在两个集群之间同步消息（同个地域或不同地域间），您可以按照 Topic 维度，把集群 A 的某个 Topic 的消息复制到集群 B 的某个 Topic。在进行某个 Topic 的消息复制时，RocketMQ 支持按照特定的条件进行过滤（如 Tag 或者 SQL 表达式），支持复制任务的任意启停，并且支持通过监控查看复制任务的进度和健康程度。

操作步骤

创建任务

进入跨集群复制页面，单击页面上方的新建任务， 按照要求填写以下字段：

- 任务名称：200字符以内，只能包含 中文、数字、字母、“-”和“_”；
- 源 Topic：通过下拉依次选择地域、集群、命名空间和 Topic，如果找不到需要的集群或 Topic 可以在集群列表页进行新建。
- 目标 Topic：通过下拉依次选择地域、集群、命名空间和 Topic，如果找不到需要的集群或 Topic 可以在集群列表页进行新建。
- 过滤类型：支持 TAG 过滤和 SQL 过滤两个方式。
- 复制起始位置：支持从最新的位点开始复制或者指定时间点开始复制。
- 是否立即开启任务：如果打开开关，在任务创建完成后就按照当前任务的配置进行复制。

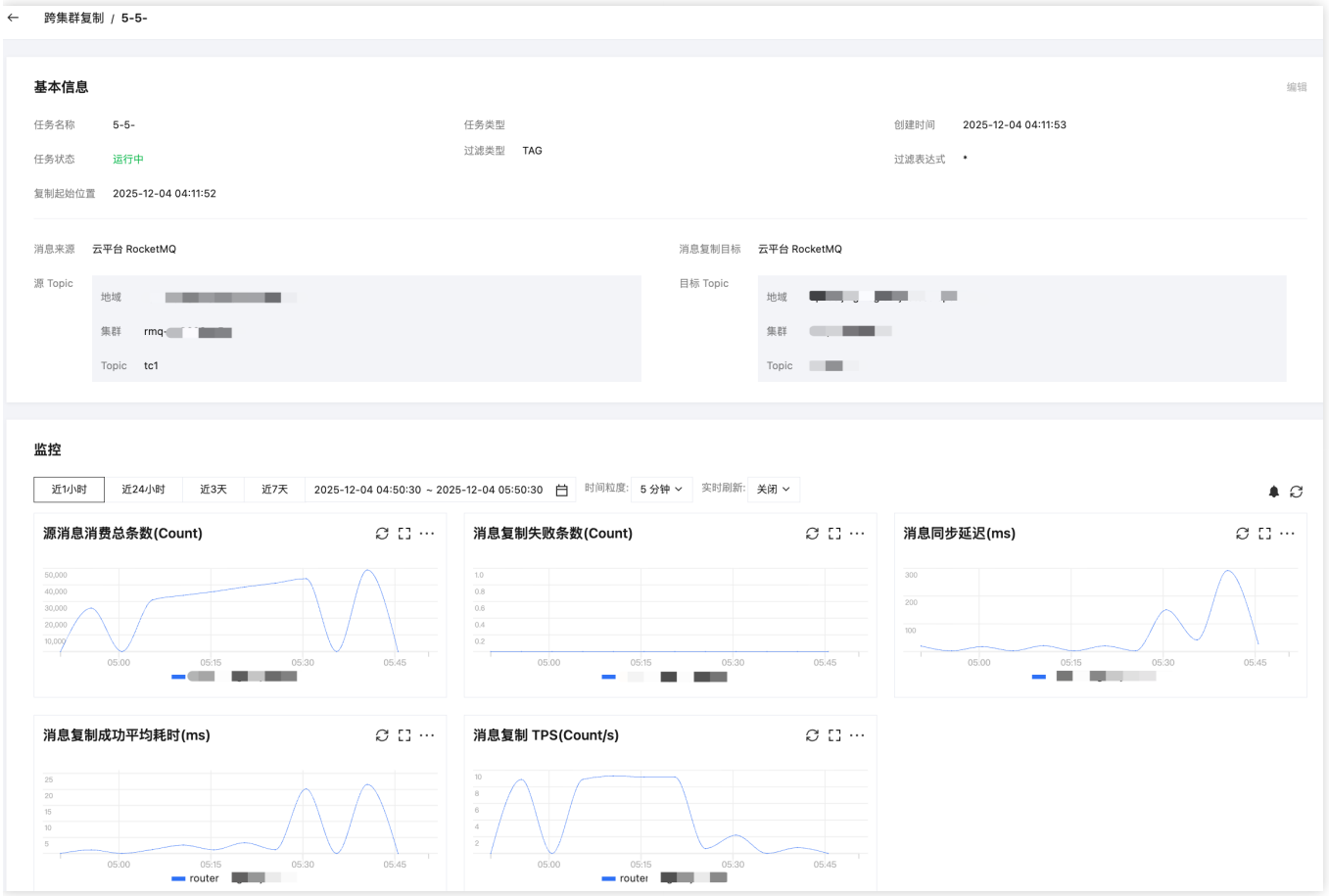


单击创建任务后，会跳转到任务列表页，在任务初始化后即创建完成。
您创建的复制任务是单向的，即如果您创建一个 Topic A 到 Topic B 的复制任务，Topic A 的消息会自动复制到 Topic B；如果您需要双向的复制任务，您需要再次新建一个从 Topic B 到 Topic A 的复制任务。

查看任务详情

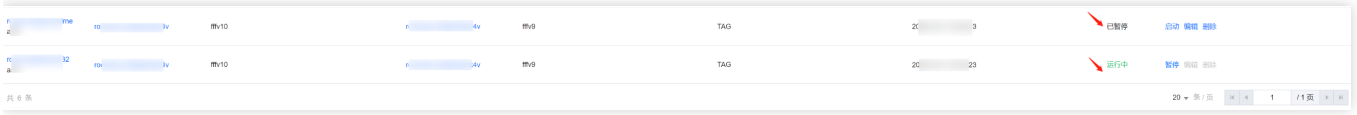
在任务创建完成后，您可以在任务的列表页看到新增的复制任务，同时可以快速查看任务的状态。单击操作列的启动/暂停

可以快速的开启和暂停任务。
进行中的任务不能修改配置信息，如果要修改复制任务的配置，请先暂停任务后，单击操作栏的编辑，或者进入任务详情页，单击“基本信息”右上角的编辑，修改任务的信息。
您可以单击任务名称，进入任务详情页查看任务的详细配置，如过滤规则和起始时间等等。 在监控部分，您可以查看当前消息复制任务的实时监控，比如复制成功的速率（XX条消息/秒），复制失败的速率，消息复制延时等。



异常处理

正常情况下，状态栏会展示 “运行中” 或者 “已暂停” 的状态；如果状态为 “启动失败”，您需要检查任务运行状态和任务详细配置是否正确，如 SQL 表达式是否正确等；鼠标悬浮在失败状态上会有具体的失败原因。



如果任务状态失败，您可以单击操作栏的编辑，或者进入任务详情页，单击“基本信息”右上角的编辑，重新更正任务的信息。

开发指南
消息类型
普通消息

普通消息是一种基础的消息类型，由生产投递到指定 Topic 后，被订阅了该 Topic 的消费者所消费。普通消息的 Topic 中无顺序的概念，可以使用多个分区数来提升消息的生产和消费效率，在吞吐量巨大时其性能最好。
普通消息区别于有特性的定时与延迟消息、顺序消息和事务消息。这四种类型的消息对应的 Topic 不能混用，只能用于收发相同类型的消息，例如普通消息的 Topic 只能用于收发普通消息，不能用于收发延迟消息、顺序消息和事务消息。

定时与延时消息

本文主要介绍消息队列 TDMQ RocketMQ 版中定时与延迟消息的概念和使用方式。

相关概念

- 定时消息：消息在发送至服务端后，实际业务并不希望消费端马上收到这条消息，而是推迟到某个时间点被消费，这类消息统称为定时消息。
- 延时消息：消息在发送至服务端后，实际业务并不希望消费端马上收到这条消息，而是推迟一段时间后再被消费，这类消息统称为延时消息。
  实际上，延时消息可以看成是定时消息的一种特殊用法，其实现的最终效果和定时消息是一致的。

使用方式

下面给出一段具体代码示例来展示如何使用 TDMQ RocketMQ 版的定时和延时消息。

定时消息

定时消息直接在 message 发送前写入标准毫秒化的时间戳到 __STARTDELIVERTIME 属性即可。

```
Message msg = new Message("test-topic", ("message content").getBytes(StandardCharsets.UTF_8));
// 设定消息在 2021-10-01 00:00:00 被发送
try {
    long timeStamp = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss").parse("2021-10-01 00:00:0
0").getTime();
// 将 __STARTDELIVERTIME 设定到 msg 的属性中
    msg.putUserProperty("__STARTDELIVERTIME", String.valueOf(timeStamp));
    SendResult result = producer.send(msg);
    System.out.println("Send delay message: " + result);
} catch (ParseException e) {
    //TODO 添加对 Timestamp 解析失败的处理方法
    e.printStackTrace();
}
```

延时消息

在RocketMQ中，支持延迟消息，但是不支持任意时间精度的延迟消息，只支持特定级别的延迟消息。如果要支持任意时间精度，不能避免在Broker层面做消息排序，再涉及到持久化的考量，那么消息排序就不可避免产生巨大的性能开销。
消息延迟级别分别为1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h，共18个级别。在发

送消息时，设置消息延迟级别即可，设置消息延迟级别时有以下3种情况：

- 设置消息延迟级别等于0时，则该消息为非延迟消息。
- 设置消息延迟级别大于等于1并且小于等于18时，消息延迟特定时间，如：设置消息延迟级别等于1，则延迟1s；设置消息延迟级别等于2，则延迟5s，以此类推。
  设置消息延迟级别大于18时，则该消息延迟级别为18，如：设置消息延迟级别等于20，则延迟2h。

```
Message msg = new Message("OneMoreTopic"
        , "DelayMessage", "This is a delay message.".getBytes());

//"1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h"
//设置消息延迟级别为3，也就是延迟10s。
msg.setDelayTimeLevel(3);

// 发送消息到一个Broker
SendResult sendResult = producer.send(msg);
// 通过sendResult返回消息是否成功送达
System.out.printf("%s Send Status: %s, Msg Id: %s %n"
        , sdf.format(new Date())
        , sendResult.getSendStatus()
        , sendResult.getMsgId());

// 如果不再发送消息，关闭Producer实例。
producer.shutdown();
```

使用限制

- 使用延时消息时，请确保客户端的机器时钟和服务端的机器时钟（所有地域均为当地时区）保持一致，否则会有时差。
- 定时和延时消息在精度上会有1秒左右的偏差。
- 关于定时时间范围，最大为40天。
- 使用定时消息时，设置的时刻在当前时刻以后才会有定时效果，否则消息将被立即发送给消费者。

顺序消息

顺序消息是消息队列 RocketMQ 提供的一种高级消息类型，对于一个指定的Topic，消息严格按照先进先出（FIFO）的原则进行消息发布和消费，即先发送的消息先消费，后发送的消息后消费。

顺序消息适用于对消息发送和消费顺序有严格要求的情况。

使用场景

顺序消息和普通消息的对比如下：

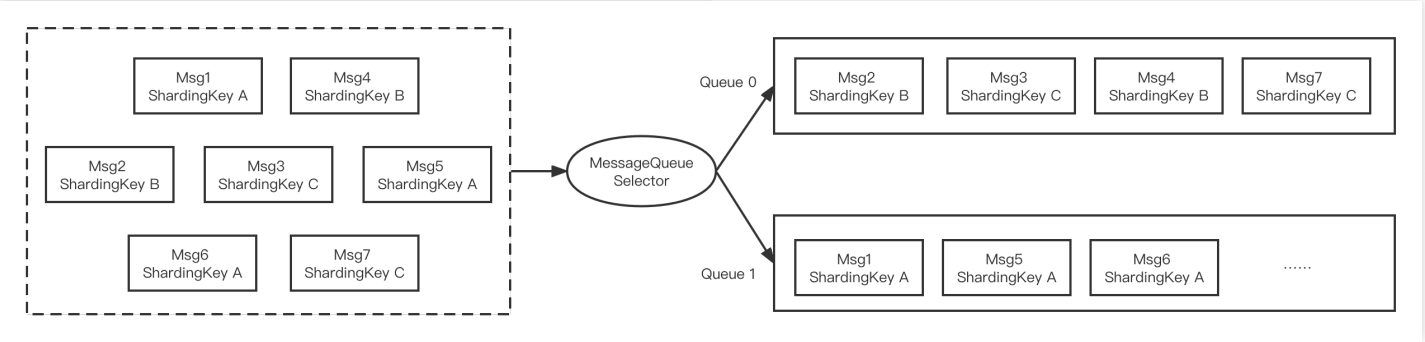
消息类型	消费顺序	性能	适用场景
普通消息	无顺序	高	适用于对吞吐量要求高，且对生产和消费顺序无要求
顺序消息	指定的 Topic 内的消息遵循先入先出（FIFO）规则	一般	吞吐量要求一般，但是要求特定的 Topic 严格地按照 FIFO 原则进行消息发布和消费的场景

对应到具体的业务场景，顺序消息可以被用在以下场景中：

- **订单创建场景：**在一些电商系统中，同一个订单相关的创建订单消息、订单支付消息、订单退款消息、订单物流消息必须严格按照先后顺序来进行生产或者消费，否则消费中传递订单状态会发生紊乱，影响业务的正常进行。因此，该订单的消息必须按照一定的顺序在客户端和消息队列中进行生产和消费，同时消息之间有先后的依赖关系，后一条消息需要依赖于前一条消息的处理结果。
- **日志同步场景：**在有序事件处理或者数据实时增量同步的场景中，顺序消息也能发挥较大的作用，如同步 mysql 的 binlog 日志时，需要保证数据库的操作是有顺序的。
- **金融场景：**在一些撮合交易的场景下，比如某些证券交易，在价格相同的情况下，先出价者优先处理，则需要按照FIFO的方式生产和消费顺序消息。

实现原理

在 RocketMQ 中支持顺序消息的原理如下图所示。我们可以按照某一个标准对消息进行分区（比如图中的 ShardingKey），同一个ShardingKey 的消息会被分配到同一个队列中，并按照顺序被消费。



顺序消息的代码如下所示：

```
public class Producer {
    public static void main(String[] args) throws UnsupportedEncodingException {
        try {
            DefaultMQProducer producer = new DefaultMQProducer("please_rename_unique_group_nam
e");
            producer.start();

            String[] tags = new String[] {"TagA", "TagB", "TagC", "TagD", "TagE"};
            for (int i = 0; i < 100; i++) {
                int orderId = i % 10;
                Message msg =
                    new Message("TopicTest", tags[i % tags.length], "KEY" + i,
                        ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHARSET));
                SendResult sendResult = producer.send(msg, new MessageQueueSelector() {
                    @Override
                    public MessageQueue select(List<MessageQueue> mqs, Message msg, Object arg) {
                        Integer id = (Integer) arg;
                        int index = id % mqs.size();
                        return mqs.get(index);
                    }
                }, orderId);

                System.out.printf("%s%n", sendResult);
            }

            producer.shutdown();
        } catch (MQClientException | RemotingException | MQBrokerException | InterruptedException e) {
            e.printStackTrace();
        }
    }
}
```

这里的区别主要是调用了 SendResult send(Message msg, MessageQueueSelector selector, Object arg) 方法， MessageQueueSelector 是队列选择器，arg 是一个 Java Object 对象，可以传入作为消息发送分区的分类标

准。

MessageQueueSelector 的接口如下：

```
public interface MessageQueueSelector {
    MessageQueue select(final List<MessageQueue> mqs, final Message msg, final Object arg);
}
```

其中 mqs 是可以发送的队列，msg 是消息，arg 是上述 send 接口中传入的 Object 对象，返回的是该消息需要发送到的队列。上述例子里，是以 orderId 作为分区分类标准，对所有队列个数取余，来对将相同 orderId 的消息发送到同一个队列中。
生产环境中建议选择最细粒度的分区键进行拆分，例如，将订单ID、用户ID作为分区键关键字，可实现同一终端用户的消息按照顺序处理，不同用户的消息无需保证顺序。

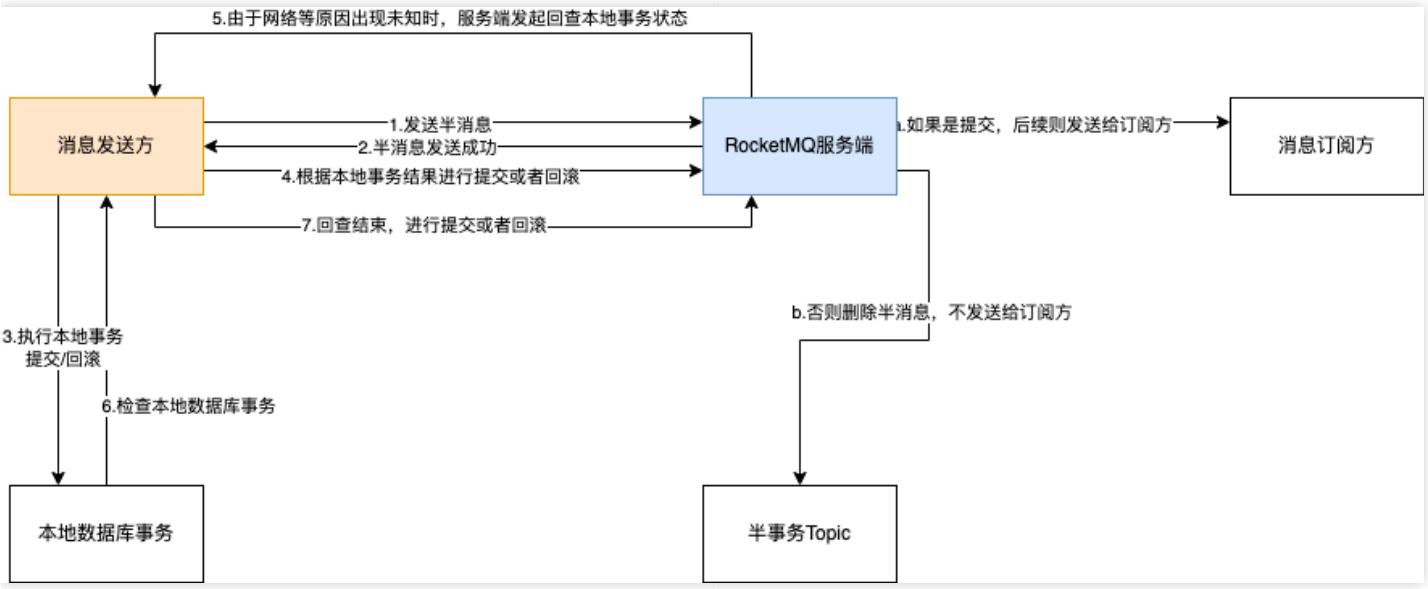
> 注意：
>
> 为了保证消息的高可用，目前TDMQ RocketMQ版不支持单队列的“全局顺序消息”（已经创建了全局顺序消息的用户可以正常使用）；如果您想保证全局的顺序性，您可以通过使用一致的 ShardingKey 来实现。

事务消息

本文主要介绍消息队列 TDMQ RocketMQ 版中事务消息的概念、技术原理、应用场景和使用方式。

功能介绍

事务消息实现了消息生产者本地事务与消息发送的原子性，保证了消息生产者本地事务处理成功与消息发送成功的最终一致。用户实现类似 X/Open XA 的分布事务功能，通过消息队列 TDMQ RocketMQ 版能达到分布式事务的最终一致。



1. 生产者发送消息到 RocketMQ 中（1）。
2. 服务端收到消息后将消息存储到半消息Topic 中（2）。
3. 当本地事务执行完成（3）。
4. 生产者主动将事务执行结果发送到 RocketMQ 中（4）。
5. 若本地事务执行结果超过一定期限还没反馈，RocketMQ 将执行回查逻辑（5）
6. 生产者收到消息回查后，需要检查对应消息的本地事务执行的最终结果，并反馈（6、7）。事务执行状态有以下三种情况：
    - TransactionStatus.COMMIT 提交事务，消费者可以消费到该消息。
    - TransactionStatus.ROLLBACK 回滚事务，消息被丢弃，消费者不会消费到该消息。
    - TransactionStatus.UN_KNOW 无法判断状态，等待再次发送回查。
7. 当事务执行成功，RocketMQ 将事务消息提交到 real topic，待消费者消费（a）。

应用场景

使用 TDMQ RocketMQ 版的事务消息来处理交易事务，可以大大提升处理效率和性能。计费的交易链路通常比较长，出错或者超时的概率比较高，借助 TDMQ 的自动重推和海量堆积能力来实现事物补偿，支付 Tips 通知和交易流水推送可以通过 TDMQ 来实现最终一致性。

消息过滤

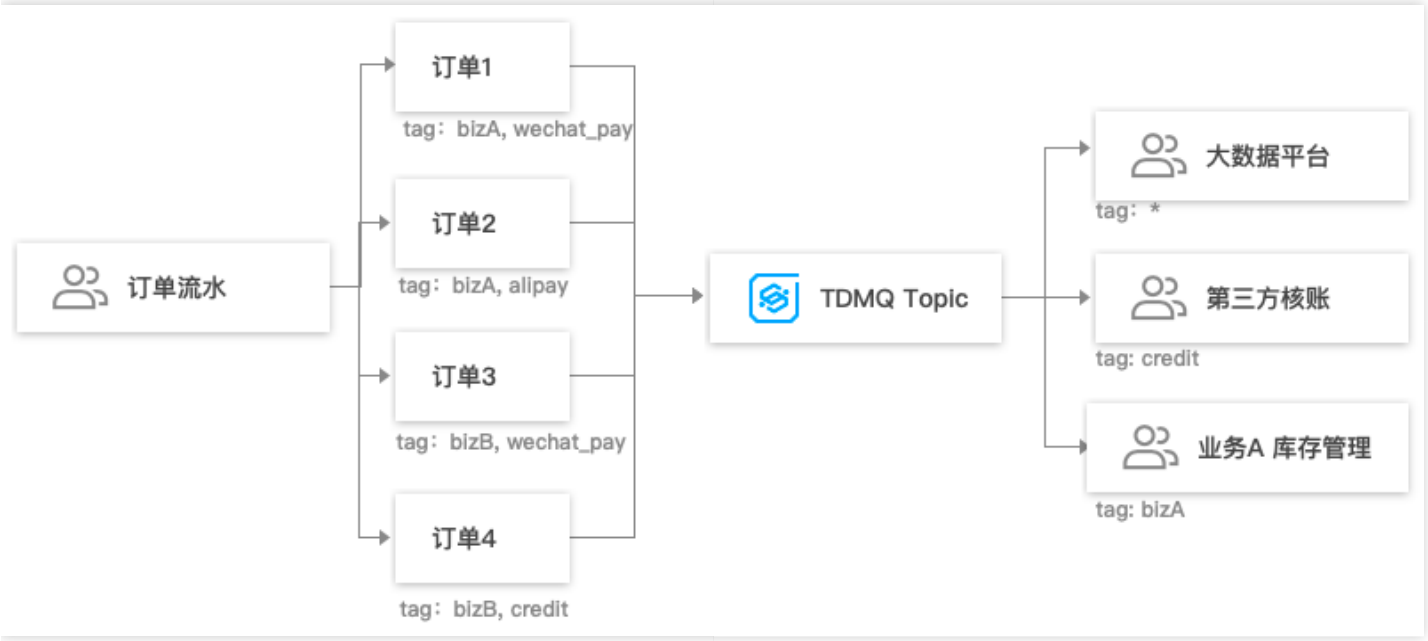
本文主要介绍 TDMQ RocketMQ 版中消息过滤的功能、应用场景和使用方式。

功能介绍

消息过滤功能指消息生产者向 Topic 中发送消息时，设置消息属性对消息进行分类，消费者订阅 Topic 时，根据消息属性设置过滤条件对消息进行过滤，只有符合过滤条件的消息才会被投递到消费端进行消费。
消费者订阅 Topic 时若未设置过滤条件，无论消息发送时是否有设置过滤属性，Topic 中的所有消息都将被投递到消费端进行消费。

应用场景

通常，一个 Topic 中存放的是相同业务属性的消息，例如交易流水 Topic 包含了下单流水、支付流水、发货流水等，业务若只想消费者其中一种类别的流水，可在客户端进行过滤，但这种过滤方式会带来带宽的资源浪费。
针对上述场景，TDMQ 提供 Broker 端过滤的方式，用户可在生产消息时设置一个或者多个 Tag 标签，消费时指定 Tag 订阅。



使用方式

TAG 过滤

发送消息

发送消息时，每条消息必须指明 Tag。

```
Message msg = new Message("TOPIC","TagA","Hello world".getBytes());
```

订阅消息

- 订阅所有 Tag：消费者如需订阅某 Topic 下所有类型的消息，Tag 用星号（*）表示。

```
consumer.subscribe("TOPIC", "*", new MessageListener() {
    public Action consume(Message message, ConsumeContext context) {
        System.out.println(message.getMsgID());
        return Action.CommitMessage;
    }
});
```

- 订阅单个 Tag：消费者如需订阅某 Topic 下某一种类型的消息，请明确标明 Tag。

```
consumer.subscribe("TOPIC", "TagA", new MessageListener() {
    public Action consume(Message message, ConsumeContext context) {
        System.out.println(message.getMsgID());
        return Action.CommitMessage;
    }
});
```

- 订阅多个 Tag：消费者如需订阅某 Topic 下多种类型的消息，请在多个 Tag 之间用两个竖线（||）分隔。

```
consumer.subscribe("TOPIC", "TagA||TagB", new MessageListener() {
    public Action consume(Message message, ConsumeContext context) {
        System.out.println(message.getMsgID());
        return Action.CommitMessage;
    }
});
```

SQL 过滤

发送消息

发送代码和简单的消息没有区别 主要是在构造消息体的时候，带上自定义属性，允许多个。

```
int totalMessagesToSend = 5;
```

```
for (int i = 0; i < totalMessagesToSend; i++) {
    Message msg = new Message(TOPIC_NAME,"Hello RocketMQ.".getBytes(StandardCharsets.UTF_8));
    msg.putUserProperty("key1","value1");
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

订阅消息

对于消费消息，主要是订阅的时候，带上对应的SQL表达式，其他的和普通的消费消息流程没有区别。

```
//订阅所有消息
pushConsumer.subscribe(TOPIC_NAME, MessageSelector.bySql("True"));

// 订阅topic 订阅单个key的sql，最常用
//pushConsumer.subscribe(TOPIC_NAME,    MessageSelector.bySql("key1 IS NOT NULL AND key1='val
ue1'"));

//订阅多个属性
//pushConsumer.subscribe(TOPIC_NAME,    MessageSelector.bySql("key1 IS NOT NULL AND key2 IS N
OT NULL  AND key1='value1' AND key2='value2'"));

// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

> 说明
>
> 上述是对消息的发布和订阅方式的简单介绍。更多操作可参见 GitHub Demo 或 RocketMQ 官方文档 。

消费模式

本文主要介绍消息队列 TDMQ RocketMQ 版中集群消费和广播消费的相关功能和应用场景。

功能介绍

- 集群消费：当使用集群消费模式时，任意一条消息只需要被集群内的任意一个消费者处理即可。
- 广播消费：当使用广播消费模式时，每条消息会被推送给集群内所有注册过的消费者，保证消息至少被每个消费者消费一次。

应用场景

- 集群消费：适用于每条消息只需要被处理一次的场景。
- 广播消费：适用于每条消息需要被集群下每一个消费者处理的场景。

代码示例

- 集群订阅
  同一个 Group ID 所标识的所有 Consumer 平均分摊消费消息。例如某个 Topic 有9条消息，一个 Group ID 有3个 Consumer 实例，那么在集群消费模式下每个实例平均分摊，只消费其中的3条消息。

  ```
  // 集群订阅方式设置（不设置的情况下，默认为集群订阅方式）。
  properties.put(PropertyKeyConst.MessageModel, PropertyValueConst.CLUSTERING);
  ```

- 广播订阅
  同一个 Group ID 所标识的所有 Consumer 都会各自消费某条消息一次。例如某个 Topic 有9条消息，一个 Group ID 有3个 Consumer 实例，那么在广播消费模式下每个实例都会各自消费9条消息。

  ```
  // 广播订阅方式设置。
  properties.put(PropertyKeyConst.MessageModel, PropertyValueConst.BROADCASTING);
  ```

  > 说明
  >
  > 请确保同一个 Group ID 下所有 Consumer 实例的订阅关系保持一致。

消息重试

本文主要介绍消息队列 TDMQ RocketMQ 版中消息重试与使用方法。

功能介绍

当消息第一次被消费者消费后，没有得到正常的回应，或者用户主动要求服务端重投，TDMQ RocketMQ 版会通过消费重试机制自动重新投递该消息，直到该消息被成功消费，当重试达到一定次数后，消息仍未被成功消费，则会停止重试，将消息投递到死信队列中。
当消息进入到死信队列中，表示 TDMQ RocketMQ 版已经无法自动处理这批消息，一般这时就需要人为介入来处理这批消息。您可以通过编写专门的客户端来订阅死信 Topic，处理这批之前处理失败的消息。

> 说明
>
> 只有当消费模式为集群消费模式时，Broker 才会自动进行重试，广播消费模式下不会进行重试。

出现以下三种情况会按照消费失败处理并会发起重试：

1. 消费者返回 ConsumeConcurrentlyStatus.RECONSUME_LATER。
2. 消费者返回 null。
3. 消费者主动/被动抛出异常。

重试次数

当消息需要重试时，TDMQ RocketMQ 中配置了如下的 messageDelayLevel 参数来设置重试次数与时间间隔。

```
messageDelayLevel=1s 5s 10s 30s 1m 2m 3m 4m 5m 6m 7m 8m 9m 10m 20m 30m 1h 2h
```

重试次数与重试时间间隔关系如下：

第几次重试	距离上一次重试的时间间隔	第几次重试	距离上一次重试的时间间隔
1	1秒	10	6分钟
2	5秒	11	7分钟
3	10秒	12	8分钟
4	30秒	13	9分钟
5	1分钟	14	10分钟

第几次重试	距离上一次重试的时间间隔	第几次重试	距离上一次重试的时间间隔
6	2分钟	15	20分钟
7	3分钟	16	30分钟
8	4分钟	17	1小时
9	5分钟	18	2小时

使用方式

4.x SDK

如果用户需要自行调整这个重试次数。可以通过设置 consumer 的参数来决定。

```
pushConsumer.setMaxReconsumeTimes(3);
```

5.x SDK

无需特别处理，5.0的 SDK 遵循上述规则。

死信队列

本文主要介绍消息队列 TDMQ RocketMQ 版中消息死信队列与使用方法。

功能介绍

- 当消息第一次被消费者消费后，没有得到正常的回应，或者用户主动要求服务端重投，TDMQ RocketMQ 版会通过消费重试机制自动重新投递该消息，直到该消息被成功消费，当重试达到一定次数后，消息仍未被成功消费，则会停止重试，将消息投递到死信队列中。
- 当消息进入到死信队列中，表示 TDMQ RocketMQ 版已经无法自动处理这批消息，一般这时就需要人为介入来处理这批消息。您可以通过编写专门的客户端来订阅死信 Topic，处理这批之前处理失败的消息。

特性说明

- 不同于重试队列，会自动消费，死信队列的消息需要有新的代码处理，或者人工介入
- 消息有效期依然遵守默认三天删除的规则。
- 死信队列开头%DLQ%，和消费组一一对应，因此一个死信队列包含了对应Group ID 的所有死信消息，不论消息属于哪个Topic。

集群消费与广播消费

本文主要介绍消息队列 TDMQ RocketMQ 版中集群消费和广播消费的相关功能和应用场景。

功能介绍

- 集群消费：当使用集群消费模式时，任意一条消息只需要被集群内的任意一个消费者处理即可。
- 广播消费：当使用广播消费模式时，每条消息会被推送给集群内所有注册过的消费者，保证消息至少被每个消费者消费一次。

应用场景

- 集群消费：适用于每条消息只需要被处理一次的场景。
- 广播消费：适用于每条消息需要被集群下每一个消费者处理的场景。

代码示例

- 集群订阅

5.x SDK 默认为集群消费模式，无需特殊设置。

> 说明：
>
> 请确保同一个 Group ID 下所有 Consumer 实例的订阅关系保持一致。

- 同一个 Group ID 所标识的所有 Consumer 平均分摊消费消息。例如某个 Topic 有9条消息，一个 Group ID 有3个 Consumer 实例，那么在集群消费模式下每个实例平均分摊，只消费其中的3条消息。

```
// 集群订阅方式设置（不设置的情况下，默认为集群订阅方式）。
properties.put(PropertyKeyConst.MessageModel, PropertyValueConst.CLUSTERING);
```

- 广播订阅

5.x gRPC SDK 已经不支持广播消费，如果需要使用，可以给每个消费者创建一个单独的订阅组实现类似的功能。

> 说明：
>
> 请确保同一个 Group ID 下所有 Consumer 实例的订阅关系保持一致。

TDMQ RocketMQ 5.x 在开源代码的基础上适配了广播消费，对于 5.x Remoting SDK 可以使用广播消费，开启的方式和4.x SDK 相同。

```
// 广播订阅方式设置。
properties.put(PropertyKeyConst.MessageModel, PropertyValueConst.BROADCASTING);
```

同一个 Group ID 所标识的所有 Consumer 都会各自消费某条消息一次。例如某个 Topic 有9条消息，一个 Group ID 有3个 Consumer 实例，那么在广播消费模式下每个实例都会各自消费9条消息。

```
// 广播订阅方式设置。
properties.put(PropertyKeyConst.MessageModel, PropertyValueConst.BROADCASTING);
```

> 说明：
>
> 请确保同一个 Group ID 下所有 Consumer 实例的订阅关系保持一致。

SDK文档
SDK 概述

消息队列 RocketMQ 版是由云平台独立维护的代码分支且 100% 兼容 RocketMQ 协议的消息队列，具备高性能、高可用、高可靠、高弹性、高安全等特征。本文主要介绍 RocketMQ 集群不同版本 SDK 的兼容性说明，以及使用客户端收发消息前连接信息的收集，例如获取集群的接入点、访问集群的角色和密钥等。

兼容性说明

RocketMQ 5.0 提供了全新的基于 gRPC 协议的 5.x SDK，新版本 SDK 更加轻量化，多语言支持更好，建议优先使用。同时，消息队列 RocketMQ 版 5.x 系列也支持存量业务继续使用 4.x SDK 访问，兼容性说明如下：

<table>
<tr><th>服务端版本</th><th colspan="2">客户端版本</th><th>兼容性</th></tr>
<tr><td rowspan="3">5.x</td><td colspan="2">5.x SDK</td><td>完全兼容</td></tr>
<tr><td rowspan="2">4.x SDK</td><td>版本 >= 4.9.5</td><td>• PushConsumer CONSUME_FROM_TIMESTAMP 暂不生效（控制台可以重置位点）。
• 消费者 setPullBatchSize 最大值为 32（服务端会最终限制为 32）。</td></tr>
<tr><td>版本 < 4.9.5</td><td>• PushConsumer CONSUME_FROM_TIMESTAMP 暂不生效（控制台可以重置位点）。
• 消费者 setPullBatchSize 最大值为 32（服务端会最终限制为 32）。
• PullConsumer 消费暂不支持。</td></tr>
</table>

客户端网络环境说明

客户端可通过以下三种网络方式连接并访问 RocketMQ 集群：

1. 云内网络访问
   客户端部署在当前云平台内部，网络默认互通，可以直接使用 VPC 网络接入点访问。
2. 云外网络访问
   客户端部署在当前云平台外部，由于网络隔离，无法直接通信。若需要跨网通信，可以咨询运维使用云外接入点

获取连接信息

以下连接信息均需登录 TDMQ RocketMQ 版控制台获取。

1. 集群接入点地址和端口信息。

在集群基本信息页面的接入信息模块获取，根据网络类型的不同选择不同的接入点。

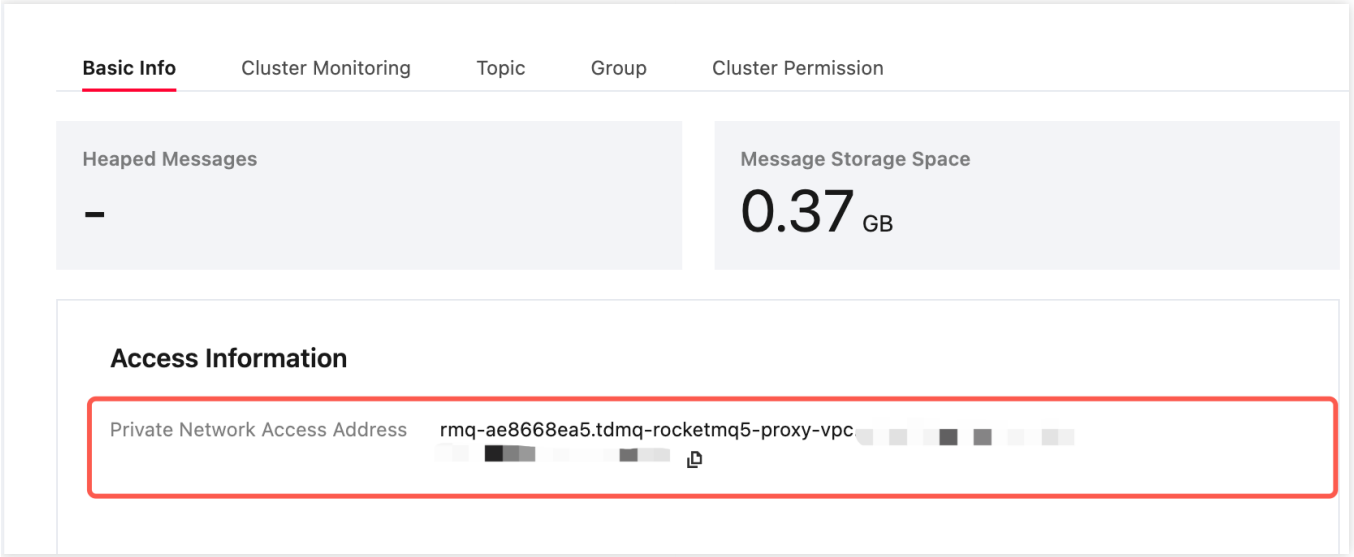
4.x

在控制台的 命名空间 列表页面获取。



5.x

集群基本信息 页面获取



2. 访问集群的角色和密钥，需确保该用户已拥有相应的访问权限。

在控制台的**集群权限（5.x 集群）或者角色管理（4.x 集群）**页面获取。



3. Topic 名称。

在控制台的 Topic 列表页面获取。



4. Group 名称。

   在控制台的 Group列表页面获取。

4.x SDK
Spring Boot Starter 接入
发送与接收普通消息

操作场景

本文以调用 Spring Boot Starter 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

操作步骤

步骤1：添加依赖

在 pom.xml 中添加依赖。

```
<dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-spring-boot-starter</artifactId>
    <version>2.2.2</version>
</dependency>
```

步骤2：准备配置

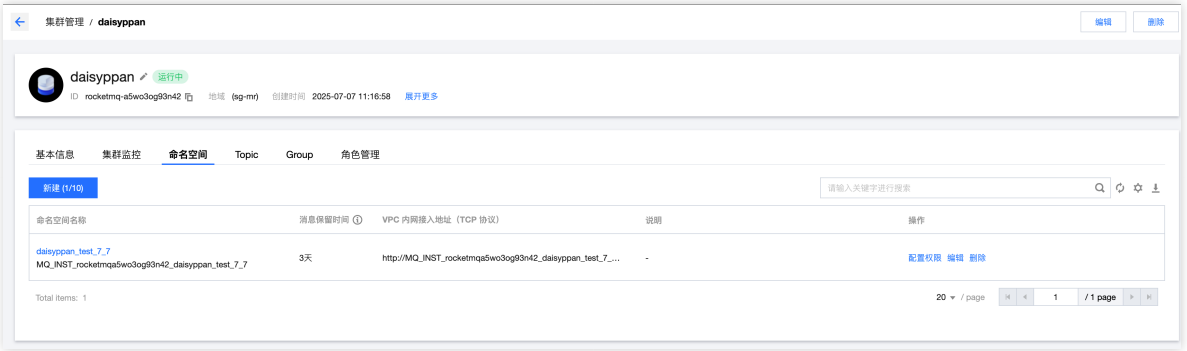
在配置文件中添加配置信息。

```
server:
  port: 8082

#rocketmq配置信息
rocketmq:
  # tdmq-rocketmq服务接入地址
  name-server: 见租户端接入地址
  # 生产者配置
  producer:
    # 生产者组名
    group: group111
    # 角色密钥
    access-key: eyJrZXlJZC....
    # 已授权的角色名称
    secret-key: admin
  # 消费者公共配置
  consumer:
```

```
  # 角色密钥
  access-key: eyJrZXlJZC....
  # 已授权的角色名称
  secret-key: admin

# 自定义配置，根据业务进行配置
namespace: rocketmq-xxx|namespace1
producer1:
  topic: testdev1
consumer1:
  group: group111
  topic: testdev1
  subExpression: TAG1
consumer2:
  group: group222
  topic: testdev1
  subExpression: TAG2
```

参数	说明
name-server	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。 
group	生产者 Group 的名称，在控制台 Group 页面复制。
secret-key	角色名称，在 角色管理 页面复制。 
access-key	角色密钥，在角色管理页面复制。
namespace	命名空间的名称，在控制台命名空间页面复制。
topic	Topic 的名称，在控制台 topic 页面复制。
subExpression	用来设置消息的 TAG。

步骤3：发送消息

1. 在需要发送消息的类中注入 RcoketMQTemplate 。

```
@Value("${rocketmq.namespace}%${rocketmq.producer1.topic}")
    private String topic;  // topic名称 (需要使用topic全称，所以在这里对topic名称进行拼接)

    @Autowired
    private RocketMQTemplate rocketMQTemplate;
```

2. 发送消息，消息体可以是自定义对象，也可以是 Message 对象（org.springframework.messaging包中）。

```
SendResult sendResult = rocketMQTemplate.syncSend(destination, message);
/*----------------------------------------------------------------------*/
rocketMQTemplate.syncSend(destination, MessageBuilder.withPayload(message).build())
```

3. 完整示例如下。

```
/**
* Description: 消息生产者
*/
@Service
public class SendMessage {
        // 需要使用topic全称，所以进行topic名称的拼接，也可以自己设置  格式：namespace全称%topic名称
 @Value("${rocketmq.namespace}%${rocketmq.producer1.topic}")
 private String topic;
    @Autowired
 private RocketMQTemplate rocketMQTemplate;
       /**
   * 同步发送
   *
   * @param message 消息内容
   * @param tags    订阅tags
   */
 public void syncSend(String message, String tags) {
       // springboot不支持使用header传递tags，根据要求，需要在topic后进行拼接 formats: `topicName:tags`，不拼接标识无tag
       String destination = StringUtils.isBlank(tags) ? topic : topic + ":" + tags;
       SendResult sendResult = rocketMQTemplate.syncSend(destination,
                   MessageBuilder.withPayload(message)
                           .setHeader(MessageConst.PROPERTY_KEYS, "yourKey")  // 指定业务key
                           .build());
       System.out.printf("syncSend1 to topic %s sendResult=%s %n", topic, sendResult);
 }
}
```

> 说明
>
> 该示例为同步发送。异步发送，单向发送等请参见 Demo或者前往GitHub 项目。

步骤4：消费消息

```
@Service
  @RocketMQMessageListener(
        consumerGroup = "${rocketmq.namespace}%${rocketmq.consumer1.group}",  // 消费组，格式：namespace全称%group名称
```

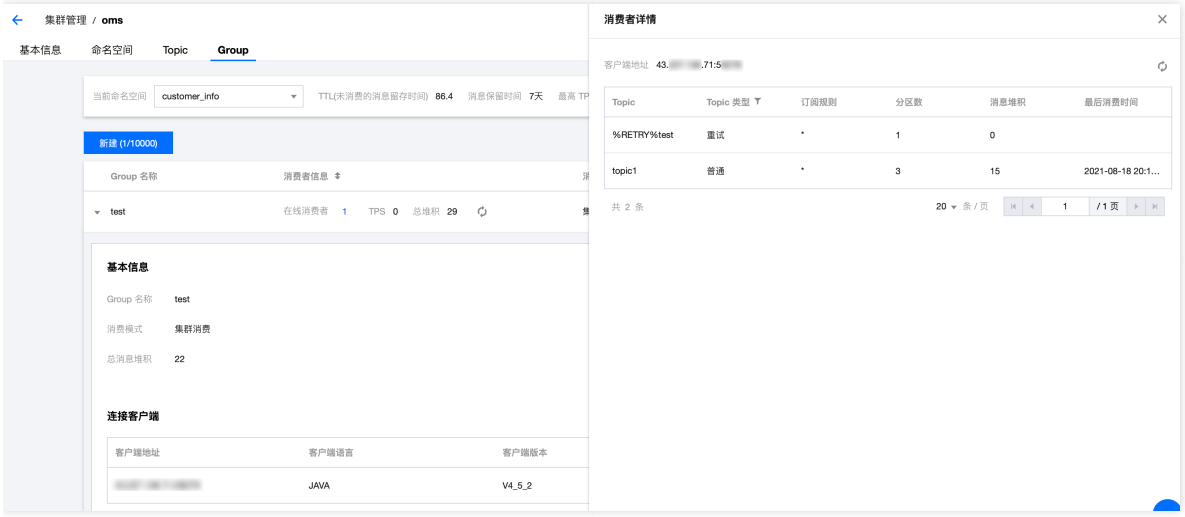
```
        // 需要使用topic全称，所以进行topic名称的拼接，也可以自己设置  格式：namespace全称%topic名称
        topic = "${rocketmq.namespace}%${rocketmq.consumer1.topic}",
        selectorExpression = "${rocketmq.consumer1.subExpression}" // 订阅表达式, 不配置表示订阅所有消息
)
public class MessageConsumer implements RocketMQListener<String> {

    @Override
    public void onMessage(String message) {
        System.out.println("Tag1Consumer receive message：" + message);
    }
}
```

可根据业务需求配置多个消费者。消费者其他配置可根据具体业务需求进行配置。

步骤5：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。



发送与接收过滤消息

操作场景

本文以调用 Spring Boot Starter 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过滤收发的完整过程。

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
- 已经了解发送与接收普通消息

操作步骤

发送消息

和普通发送消息相同，发送时，只需要将 rocketMQTemplate 的发送 Topic 拼接上 对应的 tag 即可。

```
 // springboot不支持使用header传递tags，根据要求，需要在topic后进行拼接 formats: topicName:tags，不拼接标识无tag
        String destination = StringUtils.isBlank(tags) ? topic : topic + ":" + tags;
        // object消息类型
        SendResult sendResult = rocketMQTemplate.syncSend(destination,
                MessageBuilder.withPayload(message)
                        .setHeader(MessageConst.PROPERTY_KEYS, "yourKey")  // 指定业务key
                        .build());
        System.out.printf("syncSend1 to topic %s sendResult=%s %n", topic, sendResult);
```

例如 topic 是 TopicTest，tag是 TAG1, 那么调用 rocketMQTemplate 方法的第一个参数将是 TopicTest:TAG1

消费消息

设置 selectorExpression 字段为对应的过滤tag即可。如下代码中，将 rocketmq.consumer1.subExpression 设置为TAG1 就可以消费 TAG1 的消息

```
@Service
@RocketMQMessageListener(
```

```
        consumerGroup = "${rocketmq.namespace}%${rocketmq.consumer1.group}",  // 消费组，格式：namespace全称%group名称
        // 需要使用topic全称，所以进行topic名称的拼接，也可以自己设置  格式：namespace全称%topic名称
        topic = "${rocketmq.namespace}%${rocketmq.consumer1.topic}",
        selectorExpression = "${rocketmq.consumer1.subExpression}" // 订阅表达式, 不配置表示订阅所有消息
)
public class Tag1Consumer implements RocketMQListener<String> {

    @Override
    public void onMessage(String message) {
        System.out.println("Tag1Consumer receive message：" + message);
    }
}
```

发送与接收延迟消息

操作场景

本文以调用 Spring Boot Starter SDK 为例介绍通过开源 SDK 实现消息收发的操作过程，帮助您更好地理解延迟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
- 已经了解发送与接收普通消息

操作步骤

发送消息

和普通发送消息相同，在调用发送方法的时候，传递对应的延迟级别即可。

```
SendResult sendResult = rocketMQTemplate.syncSend(
        destination,
        MessageBuilder.withPayload(message).build(),
        5000,
        delayLevel);
```

> 延迟等级与时间对应关系
>
> 其他延迟级别和具体延迟时间的对应关系如下：
>
> 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
>
> 1s、 5s、 10s、 30s、 1m、 2m、 3m、 4m、 5m、 6m、 7m、 8m、 9m、 10m、 20m、 30m、 1h、 2h；

消费消息

和普通消息相同，无需特殊处理。

Java SDK
发送与接收普通消息

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

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 maven 工程为例，在 pom.xml 添加以下依赖：

> 说明
>
> 依赖版本要求 ≥ 4.9.3, 当前建议为4.9.4

```
<!-- in your <dependencies> block -->
  <dependency>
      <groupId>org.apache.rocketmq</groupId>
      <artifactId>rocketmq-client</artifactId>
      <version>4.9.4</version>
  </dependency>

  <dependency>
      <groupId>org.apache.rocketmq</groupId>
      <artifactId>rocketmq-acl</artifactId>
      <version>4.9.4</version>
  </dependency>
```

步骤2：生产消息

创建消息生产者

```
// 实例化消息生产者Producer
    DefaultMQProducer producer = new DefaultMQProducer(
        groupName,
        new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)) // ACL权限
    );
         // 设置NameServer的地址
    producer.setNamesrvAddr(nameserver);
```

```
// 启动Producer实例
producer.start();
```

参数	说明
groupName	生产者组名称，建议使用对应的 topic 名字
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 
secretKey	角色名称，在 角色管理 页面复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。

发送消息

发送消息有多种方式：同步发送、异步发送、单向发送等。

同步发送

```
for (int i = 0; i < 10; i++) {
        // 创建消息实例，设置topic和消息内容
        Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHAR
SET));
        // 发送消息
        SendResult sendResult = producer.send(msg);
        System.out.printf("%s%n", sendResult);
    }
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
TAG	用来设置消息的 TAG。

异步发送

```
// 设置发送失败后不重试
    producer.setRetryTimesWhenSendAsyncFailed(0);
    // 设置发送消息的数量
    int messageCount = 10;
    final CountDownLatch countDownLatch = new CountDownLatch(messageCount);
    for (int i = 0; i < messageCount; i++) {
        try {
            final int index = i;
            // 创建消息实体，设置topic和消息内容
            Message msg = new Message(topic_name, "TAG", ("Hello rocketMq " + index).getBytes(RemotingHelper.DEFAULT_
CHARSET));
            producer.send(msg, new SendCallback() {
                @Override
```

```
            public void onSuccess(SendResult sendResult) {
                // 消息发送成功逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d OK %s %n", index, sendResult.getMsgId());
            }

            @Override
            public void onException(Throwable e) {
                // 消息发送失败逻辑
                countDownLatch.countDown();
                System.out.printf("%-10d Exception %s %n", index, e);
                e.printStackTrace();
            }
        });
    } catch (Exception e) {
        e.printStackTrace();
    }
}
countDownLatch.await(5, TimeUnit.SECONDS);
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
TAG	用来设置消息的 TAG。

单向发送

```
for (int i = 0; i < 10; i++) {
    // 创建消息实例，设置topic和消息内容
    Message msg = new Message(topic_name, "TAG", ("Hello RocketMQ " + i).getBytes(RemotingHelper.DEFAULT_CHARSET));
    // 发送单向消息
    producer.sendOneway(msg);
}
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
TAG	用来设置消息的 TAG。

步骤3：消费消息

创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。推荐Push消费模式

```
// 实例化消费者
DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
    groupName,
    new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
// 设置NameServer的地址
pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
groupName	生产者组名称，在控制台集群管理中Group 页签中复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

订阅消息

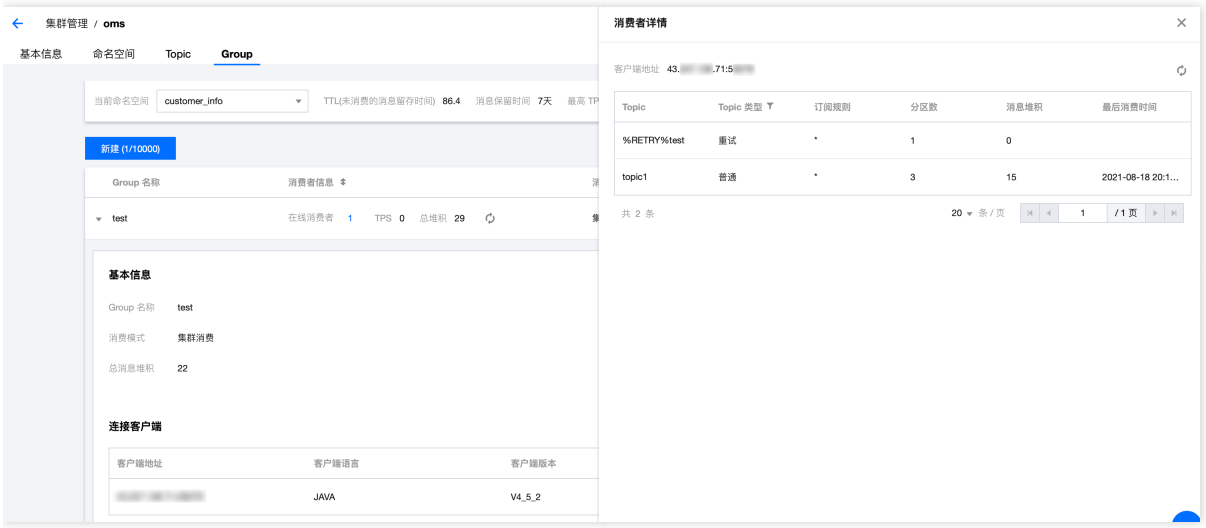
根据消费模式不同，订阅方式也有所区别。

```
// 订阅topic
    pushConsumer.subscribe(topic_name, "*");
    // 注册回调实现类来处理从broker拉取回来的消息
    pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    });
    // 启动消费者实例
    pushConsumer.start();
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。


集群管理 / oms
基本信息
命名空间
Topic
Group
当前命名空间
customer_info
TTL(未消费的消息留存时间) 86.4
消息保留时间 7天
新建 (1/10000)
Group 名称
消费者信息
test
在线消费者 1
TPS 0
总堆积 29
基本信息
Group 名称 test
消费模式 集群消费
总消息堆积 22
连接客户端
客户端地址
客户端语言
客户端版本
JAVA
V4_5_2
消费者详情
客户端地址
Topic
Topic 类型
订阅规则
分区数
消息堆积
最后消费时间
%RETRY%test
重试
1
0
topic1
普通
3
15
2021-08-18 20:1...
共 2 条
20 条 / 页
1
/ 1 页

发送与接收延迟消息

操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现定时消息收发的操作过程。

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

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 maven 工程为例，在 pom.xml 添加以下依赖：

> 说明
>
> 依赖版本要求 ≥ 4.9.3, 当前建议为4.9.4

```
<!-- in your <dependencies> block -->
 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.9.4</version>
 </dependency>

 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-acl</artifactId>
    <version>4.9.4</version>
 </dependency>
```

步骤2：生产消息

创建消息生产者

```
// 实例化消息生产者Producer
  DefaultMQProducer producer = new DefaultMQProducer(
     groupName,
     new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)) // ACL权限
  );
      // 设置NameServer的地址
  producer.setNamesrvAddr(nameserver);
  // 启动Producer实例
  producer.start();
```

参数	说明
groupName	生产者组名称，建议使用对应的topic名字
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

发送消息

固定延迟级别的消息

```
int totalMessagesToSend = 5;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message message = new Message(TOPIC_NAME, ("Hello scheduled message " + i).getBytes());
    // 设置消息延迟等级
    message.setDelayTimeLevel(5);
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

任意延迟时间的消息

```
int totalMessagesToSend = 1;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message message = new Message(TOPIC_NAME, ("Hello timer message " + i).getBytes());
    // 设置发送消息的时间
    long timeStamp = System.currentTimeMillis() + 30000;
    // 若需要发送定时消息，则需要设置定时时间，消息将在指定时间进行投递，例如消息将在2022-08-08 08:08:08投递。
    // 若设置的时间戳在当前时间之前，则消息将被立即投递给Consumer。
    //将 __STARTDELIVERTIME 设定到 msg 的属性中
    message.putUserProperty("__STARTDELIVERTIME", String.valueOf(timeStamp));
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

步骤3：消费消息

创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。推荐Push消费模式

```
// 实例化消费者
        DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
```

```
        groupName,
        new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
    // 设置NameServer的地址
    pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
groupName	生产者组名称，在控制台集群管理中Group 页签中复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

订阅消息

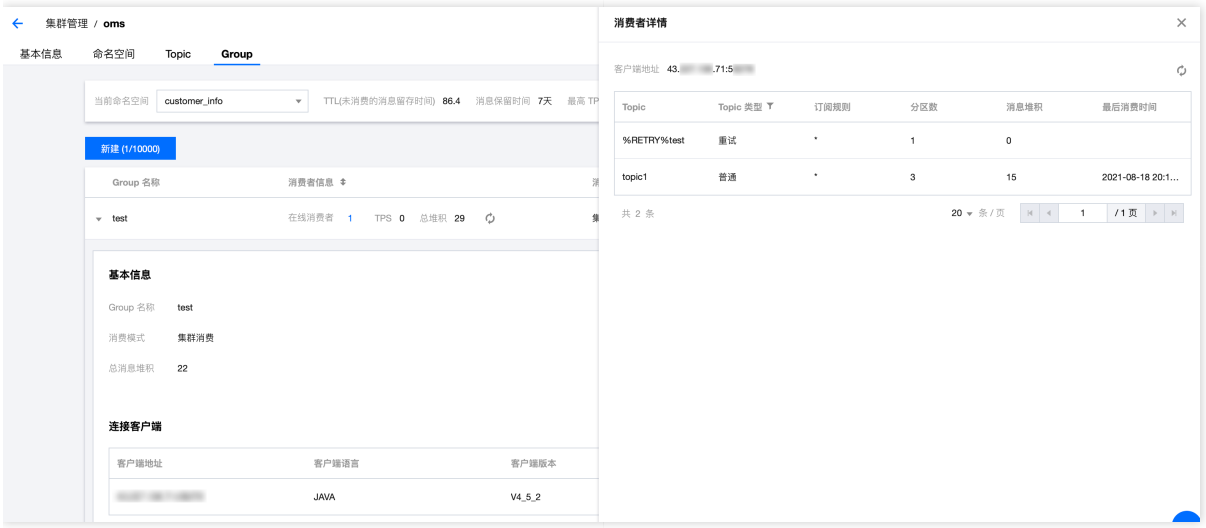
根据消费模式不同，订阅方式也有所区别。

```
// 订阅topic
    pushConsumer.subscribe(topic_name, "*");
    // 注册回调实现类来处理从broker拉取回来的消息
    pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    });
    // 启动消费者实例
    pushConsumer.start();
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。


集群管理 / oms
基本信息
命名空间
Topic
Group
当前命名空间
customer_info
TTL(未消费的消息留存时间) 86.4
消息保留时间 7天
新建 (1/10000)
Group 名称
消费者信息
test
在线消费者 1
TPS 0
总堆积 29
基本信息
Group 名称 test
消费模式 集群消费
总消息堆积 22
连接客户端
客户端地址
客户端语言
客户端版本
JAVA
V4_5_2
消费者详情
客户端地址
Topic
Topic 类型
订阅规则
分区数
消息堆积
最后消费时间
%RETRY%test
重试
1
0
topic1
普通
3
15
2021-08-18 20:1...
共 2 条
20 条 / 页
1
/ 1 页

发送与接收顺序消息

操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现定时消息收发的操作过程。

前提条件

- 完成资源创建与准备（如果是全局顺序消息，需要创建单队列topic）
- 安装1.8或以上版本 JDK
- 安装2.5或以上版本 Maven

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 maven 工程为例，在 pom.xml 添加以下依赖：

> 说明
>
> 依赖版本要求 ≥ 4.9.3, 当前建议为4.9.4

```
<!-- in your <dependencies> block -->
 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.9.4</version>
 </dependency>

 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-acl</artifactId>
    <version>4.9.4</version>
 </dependency>
```

步骤2：生产消息

创建消息生产者

```
// 实例化消息生产者Producer
  DefaultMQProducer producer = new DefaultMQProducer(
     groupName,
     new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)) // ACL权限
  );
      // 设置NameServer的地址
  producer.setNamesrvAddr(nameserver);
  // 启动Producer实例
  producer.start();
```

参数	说明
groupName	生产者组名称，建议使用对应的topic名字
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

发送消息

全局顺序消息

发送代码和简单的消息没有区别

```
int totalMessagesToSend = 5;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message message = new Message(TOPIC_NAME, ("Hello scheduled message " + i).getBytes());
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

分区有序的消息

```
for (int i = 0; i < 3; i++) {
    int orderId = i % 3;
    // 构造消息示例
    Message msg = new Message(TOPIC_NAME, "your tag", "KEY" + i,("Hello RocketMQ " + i).getBytes(StandardCharsets.UT
F_8));
    SendResult sendResult = producer.send(msg, new MessageQueueSelector() {
      @Override
      public MessageQueue select(List<MessageQueue> mqs, Message msg1, Object arg) {
          Integer id = (Integer) arg;
          int index = id % mqs.size();
          return mqs.get(index);
      }
     }, orderId);
     System.out.printf("%s%n", sendResult);
}
```

步骤3：消费消息

创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。推荐Push消费模式

```
// 实例化消费者
    DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
      groupName,
      new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
    // 设置NameServer的地址
    pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
groupName	生产者组名称，在控制台集群管理中Group 页签中复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

订阅消息

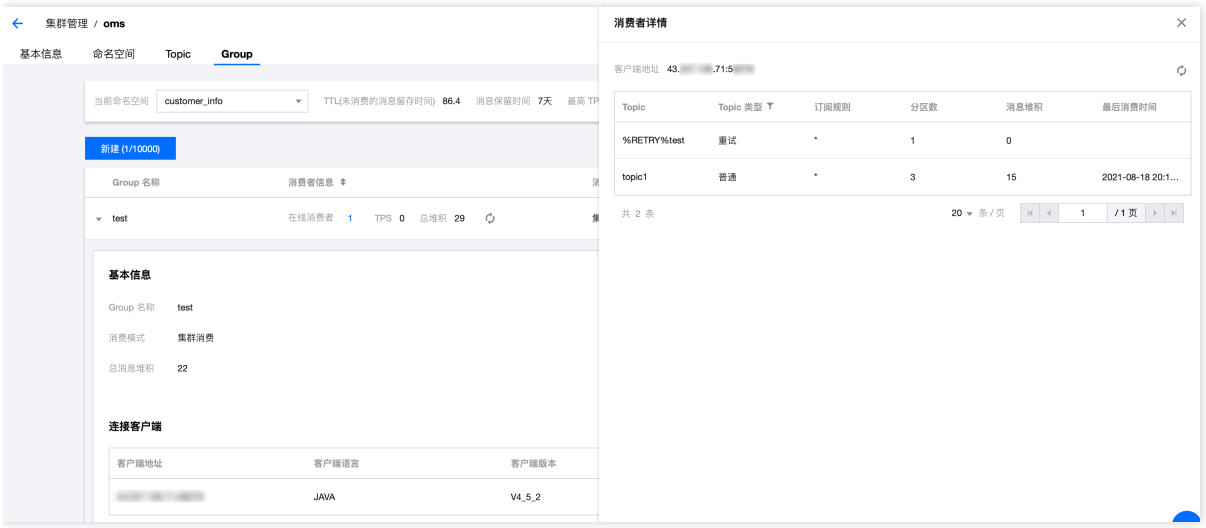
根据消费模式不同，订阅方式也有所区别。

```
// 订阅topic
    pushConsumer.subscribe(topic_name, "*");
    // 注册回调实现类来处理从broker拉取回来的消息
    pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
      // 消息处理逻辑
      System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
      // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
      return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    });
    // 启动消费者实例
    pushConsumer.start();
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。


集群管理 / oms
基本信息
命名空间
Topic
Group
当前命名空间
customer_info
TTL(未消费的消息留存时间) 86.4
消息保留时间 7天
新建 (1/10000)
Group 名称
消费者信息
test
在线消费者 1
TPS 0
总堆积 29
基本信息
Group 名称 test
消费模式 集群消费
总消息堆积 22
连接客户端
客户端地址
客户端语言
客户端版本
JAVA
V4_5_2
消费者详情
客户端地址
Topic
Topic 类型
订阅规则
分区数
消息堆积
最后消费时间
%RETRY%test
重试
1
0
topic1
普通
3
15
2021-08-18 20:1...
共 2 条
20 条 / 页
1
/ 1 页

发送与接收事务消息

操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现事务消息收发的操作过程。

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

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 maven 工程为例，在 pom.xml 添加以下依赖：

> 说明
>
> 依赖版本要求 ≥ 4.9.3, 当前建议为4.9.4

```
<!-- in your <dependencies> block -->
 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.9.4</version>
 </dependency>

 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-acl</artifactId>
    <version>4.9.4</version>
 </dependency>
```

步骤2：生产消息

实现 TransactionListener

```
public class TransactionListenerImpl implements TransactionListener {

    //半消息发送成功后，回调该方法执行本地事务
    @Override
    public LocalTransactionState executeLocalTransaction(Message msg, Object arg) {
      //这里执行数据库事务，如果成功，就返回成功，否则返回未知，或者回滚，等待回查
      return LocalTransactionState.UNKNOW;
    }
    //回查本地事务
    @Override
```

```
    public LocalTransactionState checkLocalTransaction(MessageExt msg) {
      //这里查询本地db的数据状态，然后决定是否是否提交
      return LocalTransactionState.COMMIT_MESSAGE;
    }

}
```

创建消息生产者

```
//需要用户实现一个TransactionListener 实例，
TransactionListener transactionListener = new TransactionListenerImpl();
// 实例化事务消息生产者
ProducerTransactionMQProducer producer = new TransactionMQProducer("transaction_group",
// ACL权限
new AclClientRPCHook(new SessionCredentials(ClientCreater.ACCESS_KEY, ClientCreater.SECRET_KEY)));
      // 设置NameServer的地址
producer.setNamesrvAddr(ClientCreater.NAMESERVER);
producer.setTransactionListener(transactionListener);
producer.start();
```

参数	说明
groupName	生产者组名称，建议使用对应的topic名字
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

发送消息

```
for (int i = 0; i < 3; i++) {
    // 构造消息示例
    Message msg = new Message(TOPIC_NAME, "your tag", "KEY" + i,("Hello RocketMQ " + i).getBytes(StandardCharsets.UTF_8));
    SendResult sendResult = producer.sendMessageInTransaction(msg,null);
     System.out.printf("%s%n", sendResult);
}
```

步骤3：消费消息

创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。推荐Push消费模式

```
// 实例化消费者
      DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
```

```
        groupName,
        new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
    // 设置NameServer的地址
    pushConsumer.setNamesrvAddr(nameserver);
    pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
      // 消息处理逻辑
      System.out.printf("%s Receive transaction messages: %s %n", Thread.currentThread().getName(), msgs);
      // 标记该消息已经被成功消费
      return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
      });
```

参数	说明
groupName	生产者组名称，在控制台集群管理中Group 页签中复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

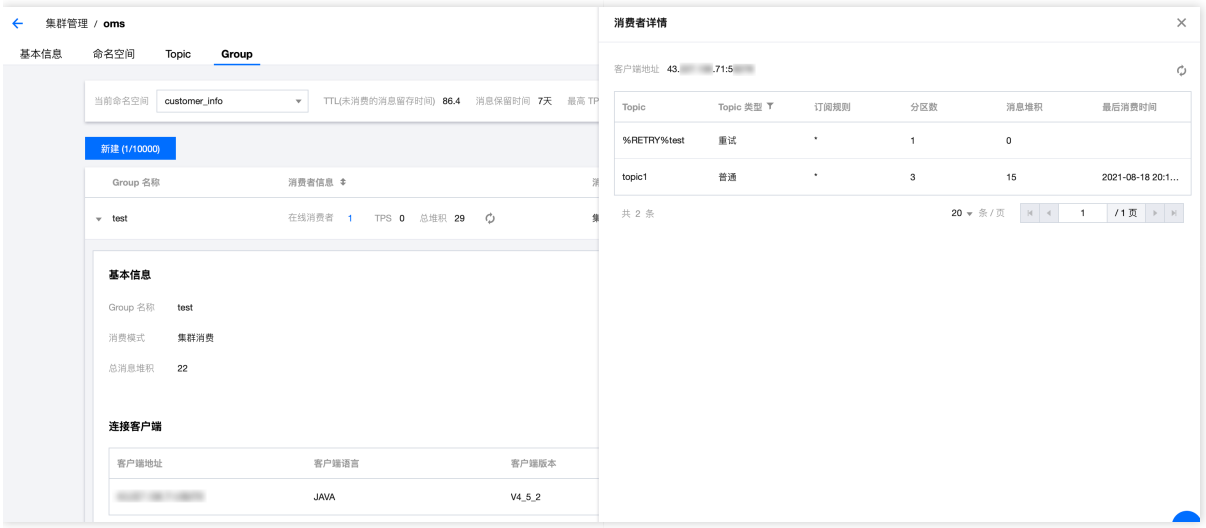
订阅消息

根据消费模式不同，订阅方式也有所区别。

```
// 订阅topic
    pushConsumer.subscribe(topic_name, "*");
    // 注册回调实现类来处理从broker拉取回来的消息
    pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
    });
    // 启动消费者实例
    pushConsumer.start();
```

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。


集群管理 / oms
基本信息
命名空间
Topic
Group
当前命名空间
customer_info
TTL(未消费的消息留存时间) 86.4
消息保留时间 7天
新建 (1/10000)
Group 名称
消费者信息
test
在线消费者 1
TPS 0
总堆积 29
基本信息
Group 名称 test
消费模式 集群消费
总消息堆积 22
连接客户端
客户端地址
客户端语言
客户端版本
JAVA
V4_5_2
消费者详情
客户端地址
Topic
Topic 类型
订阅规则
分区数
消息堆积
最后消费时间
%RETRY%test
重试
1
0
topic1
普通
3
15
2021-08-18 20:1...
共 2 条
20 条 / 页
1
/ 1 页

发送与接收过滤消息

操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现过滤消息收发的操作过程。目前支持两种，分别是Tag 和 SQL 方式

前提条件

- 完成资源创建与准备（如果是全局顺序消息，需要创建单队列topic）
- 安装1.8或以上版本 JDK
- 安装2.5或以上版本 Maven
- 参考普通消息的发送和接收

TAG使用步骤

创建生产者消费者的基础代码不在赘述，可以参考普通消息的发送和接收即可。

- 对于生产消息，主要是在构造消息体的时候，带上TAG
- 对于消费消息，主要是订阅的时候，带上单个TAG 或者 * 或者多个TAG表达式

步骤1：生产消息

发送消息

发送代码和简单的消息没有区别 主要是在构造消息体的时候，带上TAG，仅允许一个。

```
int totalMessagesToSend = 5;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message msg = new Message(TOPIC_NAME, "Tag1", "Hello RocketMQ.".getBytes(StandardCharset
s.UTF_8));
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

步骤2：消费消息

订阅消息。

```
// 订阅topic 订阅所有的TAG
pushConsumer.subscribe(topic_name, "*");

//订阅指定的TAG
//pushConsumer.subscribe(TOPIC_NAME, "Tag1");

//订阅多个TAG
//pushConsumer.subscribe(TOPIC_NAME, "Tag1||Tag2");

// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

参数	说明
topic_name	在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
"*"	订阅表达式如果为 null 或*表达式表示订阅全部，同时支持 "tag1 || tag2 || tag3" 标识订阅多个类型的 tag。

SQL 使用步骤

创建生产者消费者的基础代码不在赘述，可以参考普通消息的发送和接收即可。

- 对于生产消息，主要是在构造消息体的时候，带上用户自定义的properties
- 对于消费消息，主要是订阅的时候，带上对应的SQL表达式

步骤1：生产消息

发送代码和简单的消息没有区别 主要是在构造消息体的时候，带上自定义属性，允许多个。

```
int totalMessagesToSend = 5;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message msg = new Message(TOPIC_NAME,"Hello RocketMQ.".getBytes(StandardCharsets.UTF_8));
    msg.putUserProperty("key1","value1");
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
```

```
}
```

步骤2：消费消息

对于消费消息，主要是订阅的时候，带上对应的SQL表达式，其他的和普通的没有区别

```
pushConsumer.subscribe(TOPIC_NAME, MessageSelector.bySql("True"));

// 订阅topic 订阅单个key的sql，最常用
//pushConsumer.subscribe(TOPIC_NAME,    MessageSelector.bySql("key1 IS NOT NULL AND key1='val
ue1'"));

//订阅多个属性
//pushConsumer.subscribe(TOPIC_NAME,    MessageSelector.bySql("key1 IS NOT NULL AND key2 IS N
OT NULL  AND key1='value1' AND key2='value2'"));

// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
        // 消息处理逻辑
        System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
        // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
        return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

发送与接收广播消息

操作场景

本文以调用 Java SDK 为例介绍通过开源 SDK 实现广播消息收发的操作过程。

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

操作步骤

步骤1：安装 Java 依赖库

在 Java 项目中引入相关依赖，以 maven 工程为例，在 pom.xml 添加以下依赖：

> 说明
>
> 依赖版本要求 ≥ 4.9.3, 当前建议为4.9.4

```
<!-- in your <dependencies> block -->
 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-client</artifactId>
    <version>4.9.4</version>
 </dependency>

 <dependency>
    <groupId>org.apache.rocketmq</groupId>
    <artifactId>rocketmq-acl</artifactId>
    <version>4.9.4</version>
 </dependency>
```

步骤2：生产消息

创建消息生产者

```
// 实例化消息生产者Producer
  DefaultMQProducer producer = new DefaultMQProducer(
     groupName,
     new AclClientRPCHook(new SessionCredentials(accessKey, secretKey)) // ACL权限
  );
      // 设置NameServer的地址
  producer.setNamesrvAddr(nameserver);
  // 启动Producer实例
  producer.start();
```

参数	说明
groupName	生产者组名称，建议使用对应的topic名字
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。新版共享集群与专享集群命名接入点地址在命名空间列表获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 

发送消息

和普通的消息没有区别，广播消息主要表达的是消费方的行为。

```
int totalMessagesToSend = 5;
for (int i = 0; i < totalMessagesToSend; i++) {
    Message message = new Message(TOPIC_NAME, ("Hello scheduled message " + i).getBytes());
    // 发送消息
    SendResult sendResult = producer.send(message);
    System.out.println("sendResult = " + sendResult);
}
```

步骤3：消费消息

创建消费者

TDMQ RocketMQ 版支持 push 和 pull 两种消费模式。推荐Push消费模式

```
// 实例化消费者
        DefaultMQPushConsumer pushConsumer = new DefaultMQPushConsumer(
            groupName,
            new AclClientRPCHook(new SessionCredentials(accessKey, secretKey))); //ACL权限
        // 设置NameServer的地址
        pushConsumer.setNamesrvAddr(nameserver);
```

参数	说明
groupName	生产者组名称，在控制台集群管理中Group 页签中复制。
nameserver	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。新版共享集群与专享集群命名接入点地址在命名空间列表获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。

参数	说明
	

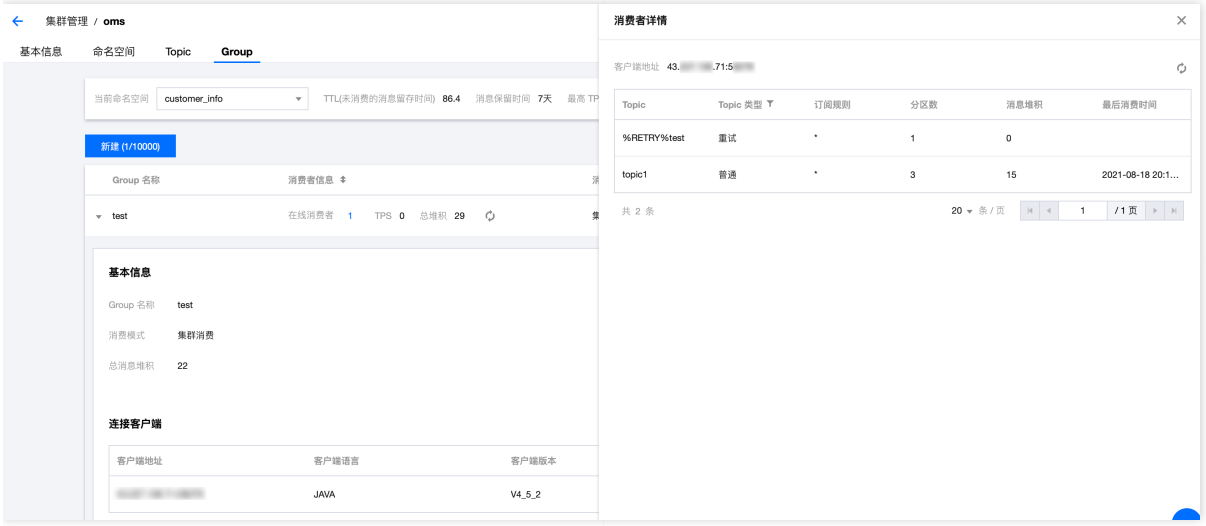
订阅消息

主要增加消费模式的设置。

```
//设置广播消费模式
pushConsumer.setMessageModel(MessageModel.BROADCASTING);
// 订阅topic
pushConsumer.subscribe(topic_name, "*");
// 注册回调实现类来处理从broker拉取回来的消息
pushConsumer.registerMessageListener((MessageListenerConcurrently) (msgs, context) -> {
  // 消息处理逻辑
  System.out.printf("%s Receive New Messages: %s %n", Thread.currentThread().getName(), msgs);
  // 标记该消息已经被成功消费, 根据消费情况，返回处理状态
  return ConsumeConcurrentlyStatus.CONSUME_SUCCESS;
});
// 启动消费者实例
pushConsumer.start();
```

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。



C++ SDK

操作场景

本文以调用 C++ 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

- 安装 GCC
- 下载 Demo

操作步骤

1. 准备环境。

   i. 需要在客户端环境安装 RocketMQ-Client-CPP 库，根据官方文档进行安装即可 安装 CPP 动态库，推荐使用 master 分支构建。

   ii. 在项目中引入 RocketMQ-Client-CPP 相关头文件及动态库。

2. 初始化消息生产者。

```
// 设置生产组名称
DefaultMQProducer producer(groupName);
// 设置服务接入地址
producer.setNamesrvAddr(nameserver);
// 设置用户权限
producer.setSessionCredentials(
    accessKey,  // 角色密钥
    secretKey, // 角色名称
    "");
// 设置命名空间(命名空间全称)
producer.setNameSpace(namespace);
// 请确保参数设置完成在启动之前
producer.start();
```

3. 发送消息。

```
// 初始化消息内容
```

```
MQMessage msg(
    topicName,  // topic名称
    TAGS,       // 消息tag
    KEYS,       // 消息业务key
    "Hello cpp client, this is a message."  // 消息内容
);

try {
    // 发送消息
    SendResult sendResult = producer.send(msg);
    std::cout << "SendResult:" << sendResult.getSendStatus() << ", Message ID: " << sendResul
t.getMsgId()
        << std::endl;
} catch (MQException e) {
    std::cout << "ErrorCode: " << e.GetError() << " Exception:" << e.what() << std::endl;
}
```

4. 资源释放。

```
// 释放资源
producer.shutdown();
```

5. 初始化消费者。

```
// 消息监听
class ExampleMessageListener : public MessageListenerConcurrently {
public:
    ConsumeStatus consumeMessage(const std::vector<MQMessageExt> &msgs) {
        for (auto item = msgs.begin(); item != msgs.end(); item++) {
            // 业务
            std::cout << "Received Message Topic:" << item->getTopic() << ", MsgId:" << item->g
etMsgId() << ", TAGS:"
                      << item->getTags() << ", KEYS:" << item->getKeys() << ", Body:" << item->get
Body() << std::endl;
        }
        // 消费成功返回CONSUME_SUCCESS
        return CONSUME_SUCCESS;
        // 消费失败返回RECONSUME_LATER，该消息将会被重新消费
        // return RECONSUME_LATER;
    }
};

// 初始化消费者
DefaultMQPushConsumer *consumer = new DefaultMQPushConsumer(groupName);
```

```
// 设置服务地址
consumer->setNamesrvAddr(nameserver);
// 设置用户权限
consumer->setSessionCredentials(
    accessKey,
    secretKey,
    "");
// 设置命名空间
consumer->setNameSpace(namespace);
// 设置实例名称
consumer->setInstanceName("CppClient");

// 请注册自定义侦听函数用来处理接收到的消息，并返回响应的处理结果。
ExampleMessageListener *messageListener = new ExampleMessageListener();
// 订阅消息
consumer->subscribe(topicName, TAGS);
// 设置消息监听
consumer->registerMessageListener(messageListener);

// 准备工作完成，必须调用启动函数，才可以正常工作。
consumer->start();
```

6. 资源释放。

```
// 资源释放
consumer->shutdown();
```

Go SDK

操作场景

本文以调用 Go 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 Go
- 下载 Demo

操作步骤

1. 在客户端环境执行如下命令下载 RocketMQ 客户端相关的依赖包。

   ```
   go get github.com/apache/rocketmq-client-go/v2
   ```

2. 在对应的方法内创建生产者，如您需要发送普通消息，则在 syncSendMessage.go 文件内修改对应的参数。
   延时消息目前支持任意精度的延时，且不受 delay level 的影响。
   普通消息

```
// 服务接入地址（注意：需要在接入地址前面加上 http:// 或 https:// 否则无法解析）
 var serverAddress = "见租户端接入地址"
 // 授权角色名
 var secretKey = "admin"
 // 授权角色密钥
 var accessKey = "eyJrZXlJZC...."
 // 命名空间全称
 var nameSpace = "MQ_INST_rocketmqem4xxxx"
 // 生产者组名称
 var groupName = "group1"
  // 创建消息生产者
 p, _ := rocketmq.NewProducer(
    // 设置服务地址
    producer.WithNsResolver(primitive.NewPassthroughResolver([]string{serverAddress})),
    // 设置acl权限
    producer.WithCredentials(primitive.Credentials{
       SecretKey: secretKey,
       AccessKey: accessKey,
    }),
     // 设置生产组
    producer.WithGroupName(groupName),
    // 设置命名空间名称
    producer.WithNamespace(nameSpace),
     // 设置发送失败重试次数
    producer.WithRetry(2),
```

```
    )
     // 启动producer
    err := p.Start()
    if err != nil {
        fmt.Printf("start producer error: %s", err.Error())
        os.Exit(1)
    }
```

延时消息

```
// topic名称
    var topicName = "topic1"
   // 生产者组名称
    var groupName = "group1"
    // 创建消息生产者
    p, _ := rocketmq.NewProducer(
      // 设置服务地址
       producer.WithNsResolver(primitive.NewPassthroughResolver([]string{"见租户端接入地址"})),
      // 设置acl权限
       producer.WithCredentials(primitive.Credentials{
          SecretKey: "admin",
          AccessKey: "eyJrZXIJZC.....",
       }),
       // 设置生产组
       producer.WithGroupName(groupName),
      // 设置命名空间名称
       producer.WithNamespace("rocketmq-xxx|namespace_go"),
       // 设置发送失败重试次数
       producer.WithRetry(2),
    )
    // 启动producer
    err := p.Start()
    if err != nil {
        fmt.Printf("start producer error: %s", err.Error())
        os.Exit(1)
    }

    for i := 0; i < 1; i++ {
        msg := primitive.NewMessage(topicName, []byte("Hello RocketMQ Go Client! This is a delay message."))
        // 设置延迟等级
        // 等级与时间对应关系：
        // 1s、 5s、 10s、 30s、 1m、 2m、 3m、 4m、 5m、 6m、 7m、 8m、 9m、 10m、 20m、 30m、 1h、 2h；
        // 1   2   3   4   5   6  7   8  9  10  11  12 13  14   15   16  17  18
        //如果想用延迟级别，那么设置下面这个方法

        msg.WithDelayTimeLevel(3)
        //如果想用任意延迟消息，那么设置下面这个方法，WithDelayTimeLevel 就不要设置了,单位为具体的毫秒，如下则是10s后投递
        delayMills := int64(10 * 1000)
        msg.WithProperty("__STARTDELIVERTIME", strconv.FormatInt(time.Now().Unix()+delayMills, 10))
    // 发送消息

       res, err := p.SendSync(context.Background(), msg)
        if err != nil {
            fmt.Printf("send message error: %s\n", err)
        } else {
```

```
        fmt.Printf("send message success: result=%s\n", res.String())
    }
}

// 释放资源
err = p.Shutdown()
if err != nil {
    fmt.Printf("shutdown producer error: %s", err.Error())
}
```

参数	说明
secretKey	角色名称，在 角色管理 页面复制。
accessKey	 角色密钥，在 角色管理 页面复制密钥列复制。
nameSpace	命名空间的名称，在控制台命名空间页面复制。
serverAddress	集群接入地址，在控制台集群管理页面的集群列表操作栏的接入地址处获取。
groupName	生产者组名称，在控制台 Group 页面复制。

3. 发送消息同上（以同步发送方式为例）。

```
// topic名称
  var topicName = "topic1"
  // 构造消息内容
  msg := &primitive.Message{
      Topic: topicName, // 设置topic名称
      Body:  []byte("Hello RocketMQ Go Client! This is a new message."),
  }
  // 设置tag
  msg.WithTag("TAG")
  // 设置key
  msg.WithKeys([]string{"yourKey"})
  // 发送消息
  res, err := p.SendSync(context.Background(), msg)

  if err != nil {
      fmt.Printf("send message error: %s\n", err)
  } else {
      fmt.Printf("send message success: result=%s\n", res.String())
  }
```

参数	说明
topicName	Topic 名称在控制台集群管理中 Topic 页签中复制具体 Topic 名称。
TAG	消息 TAG 标识。

参数	说明
yourKey	消息业务 key。

资源释放。

```
// 关闭生产者
  err = p.Shutdown()
  if err != nil {
     fmt.Printf("shutdown producer error: %s", err.Error())
  }
```

4. 创建消费者。

```
// 服务接入地址（注意：需要在接入地址前面加上 http:// 或 https:// 否则无法解析）
  var serverAddress = "见租户端接入地址"
  // 授权角色名
  var secretKey = "admin"
  // 授权角色密钥
  var accessKey = "eyJrZXlJZC...."
  // 命名空间全称
  var nameSpace = "rocketmq-xxx|namespace_go"
  // 生产者组名称
  var groupName = "group11"
  // 创建consumer
  c, err := rocketmq.NewPushConsumer(
     // 设置消费者组
     consumer.WithGroupName(groupName),
     // 设置服务地址
     consumer.WithNsResolver(primitive.NewPassthroughResolver([]string{serverAddress})),
     // 设置acl权限
     consumer.WithCredentials(primitive.Credentials{
        SecretKey: secretKey,
        AccessKey: accessKey,
     }),
     // 设置命名空间名称
     consumer.WithNamespace(nameSpace),
     // 设置从起始位置开始消费
     consumer.WithConsumeFromWhere(consumer.ConsumeFromFirstOffset),
     // 设置消费模式（默认集群模式）
     consumer.WithConsumerModel(consumer.Clustering),

     //广播消费,设置一下实例名，设置为应用的系统名即可。如果不设置，会使用pid，这会导致重启消费重复
     consumer.WithInstance("xxxx"),
  )
  if err != nil {
     fmt.Println("init consumer2 error: " + err.Error())
     os.Exit(0)
  }
```

Python SDK

操作场景

本文以调用 Python 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 Python
- 安装 pip
- 下载 Demo

操作步骤

步骤1：准备环境

Rocketmq-client Python 基于 rocketmq-client-cpp 进行包装，因此需要先安装 librocketmq 。

> 说明：
>
> 目前Python客户端仅支持 Linux 和 macOS 操作系统，暂不支持 Windows 系统。

1. 安装 librocketmq （版本2.0.0及以上）， 安装教程参见 librocketmq 安装。
2. 执行如下命令安装 rocketmq-client-python。

```
pip install rocketmq-client-python
```

步骤2：生产消息

创建并编译运行生产消息程序。

```
from rocketmq.client import Producer, Message

# 初始化生产者，并设置生产组信息，组名称使用全称，例：rocketmq-xxx|namespace_python%group1
producer = Producer(groupName)
# 设置服务地址
producer.set_name_server_address(nameserver)
# 设置权限（角色名和密钥）
producer.set_session_credentials(
    accessKey,  # 角色密钥
    secretKey,  # 角色名称
    ''
)
# 启动生产者
producer.start()
```

```
# 组装消息   topic名称，在控制台 topic 页面复制。
msg = Message(topicName)
# 设置keys
msg.set_keys(TAGS)
# 设置tags
msg.set_tags(KEYS)
# 消息内容
msg.set_body('This is a new message.')

# 发送同步消息
ret = producer.send_sync(msg)
print(ret.status, ret.msg_id, ret.offset)
# 资源释放
producer.shutdown()
```

参数	说明
groupName	生产者组名称。在控制台集群管理中 Group tab 中获取。
nameserver	集群接入地址，在控制台集群管理页面操作列的获取接入地址获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 
topicName	Topic 的名称，在控制台 Topic 页面复制。
TAGS	用来设置消息的 TAG。
KEYS	设置消息业务 key。

当前开源社区的 Python 客户端生产消息存在一定缺陷，导致同个 Topic 的不同队列间负载不均，详情可参见Github缺陷列表。

步骤3：消费消息

创建并编译运行消费消息程序。

```
import time

from rocketmq.client import PushConsumer, ConsumeStatus

# 消息处理回调
def callback(msg):
    # 模拟业务
    print('Received message. messageId: ', msg.id, ' body: ', msg.body)
    # 消费成功回复CONSUME_SUCCESS
    return ConsumeStatus.CONSUME_SUCCESS
    # 消费成功回复消息状态
    # return ConsumeStatus.RECONSUME_LATER
```

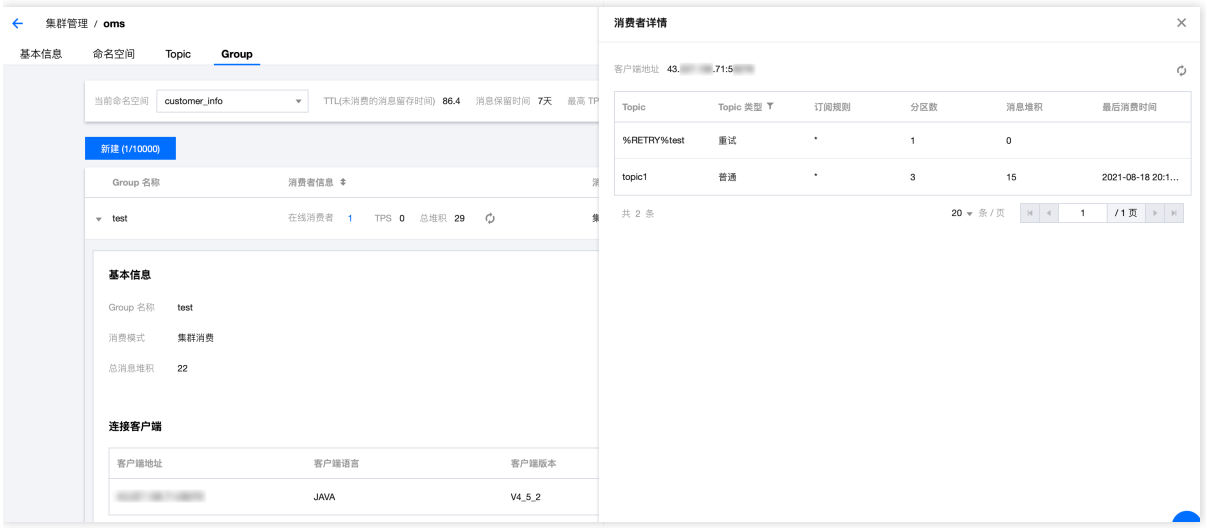
```
# 初始化消费者，并设置消费者组信息
consumer = PushConsumer(groupName)
# 设置服务地址
consumer.set_name_server_address(nameserver)
# 设置权限（角色名和密钥）
consumer.set_session_credentials(
    accessKey,  # 角色密钥
    secretKey,  # 角色名称
    ''
)
# 订阅topic
consumer.subscribe(topicName, callback, TAGS)
print(' [Consumer] Waiting for messages.')
# 启动消费者
consumer.start()

while True:
    time.sleep(3600)
# 资源释放
consumer.shutdown()
```

参数	说明
groupName	消费者 Group 的名称，在控制台 Group 页面复制。
nameserver	集群接入地址，在控制台集群管理页面操作列的获取接入地址获取。
secretKey	角色名称，在 角色管理 页面复制。
accessKey	角色密钥，在 角色管理 页面复制密钥列复制。 
topicName	Topic 的名称，在控制台 Topic 页面复制。
TAGS	设置订阅消息的tag，默认为"*"，表示订阅所有消息

步骤4：查看消费详情

登录 RocketMQ 控制台，在集群管理 > Group 页面，可查看与 Group 连接的客户端列表，单击操作列的查看详情，可查看消费者详情。


集群管理 / oms
基本信息
命名空间
Topic
Group
当前命名空间
customer_info
TTL(未消费的消息留存时间) 86.4
消息保留时间 7天
新建 (1/10000)
Group 名称
消费者信息
test
在线消费者 1
TPS 0
总堆积 29
基本信息
Group 名称 test
消费模式 集群消费
总消息堆积 22
连接客户端
客户端地址
客户端语言
客户端版本
JAVA
V4_5_2
消费者详情
客户端地址
Topic
Topic 类型
订阅规则
分区数
消息堆积
最后消费时间
%RETRY%test
重试
1
0
topic1
普通
3
15
2021-08-18 20:1...
共 2 条
20 条 / 页
1
/ 1 页

C# SDK

操作场景

本文以调用 C# 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。

> 说明：
>
> 此示例代码基于 开源 4.x C# SDK 构建，在使用过程中，出现过无法查看消费轨迹等问题，推荐使用Java SDK。

前提条件

- 安装 DotNet
- 下载 Demo

操作步骤

步骤1：准备环境

根据自己的 IDE 环境，新建一个项目，然后安装 NewLife.RocketMQ 依赖。

```
dotnet add package NewLife.RocketMQ --version 2.0.2022.325-beta0806
```

步骤2：生产消息

创建并编译运行生产消息程序。

```
//生产者
var mq = new Producer
{
    Topic = "TopicTest1", //对于非通用版集群，需要拼接完整的topic，如 MQ_INSTxxx_aaa%TopicTest
    NameServerAddress = "127.0.0.1:9876",// 云平台nameserver接入点
    Log = XTrace.Log,
    AclOptions = new AclOptions
    {
        AccessKey = "页面的ak",
        SecretKey = "页面的sk",
    },
};
mq.Start();
```

```
for (var i = 0; i < 10; i++)
{
    var str = "生产消息测试" + i;
    var sr = mq.Publish(str, "TagA");
}
```

参数	说明
Topic	Topic 的名称，在控制台 Topic 管理页面复制，此处需拼接命名空间名称，格式为 namespace全称%topic名称 ，例如 MQ_INSTxxx_aaa%TopicTest。
NameServerAddress	集群接入地址，控制台集群基本信息页面的接入信息模块获取。
SecretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。
AccessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。
TAGS	用来设置消息的 TAG。

步骤3：消费消息

创建并编译运行消费消息程序。

```
//消费者
var consumer = new Consumer
{
    Topic = "TopicTest1",//对于非通用版集群，需要拼接完整的topic，如 MQ_INSTxxx_aaa%TopicTest
    Group = "test", //对于非通用版集群，需要拼接完整的Goup，如 MQ_INSTxxx_aaa%GroupTest
    NameServerAddress = "127.0.0.1:9876",
    FromLastOffset = true,
    SkipOverStoredMsgCount = 0,
    BatchSize = 20,
    Log = XTrace.Log,
    AclOptions = new AclOptions
    {
        AccessKey = "页面的ak",
        SecretKey = "页面的sk",
    },
};
consumer.OnConsume = (q, ms) =>
{
    XTrace.WriteLine("[{0}@{1}]收到消息 [{2}]", q.BrokerName, q.QueueId, ms.Length);
    foreach (var item in ms.ToList())
    {
        XTrace.WriteLine($"消息 {item.Keys}，发送时间{item.BornTimestamp.ToDateTime()}，内容 {item.Bo
dy.ToStr()}");
    }
```

```
    return true;
};
consumer.Start();
```

参数	说明
Topic	Topic 的名称，在控制台 Topic 管理页面复制。此处需拼接命名空间名称，格式为 namespace全称%topic名称 ，例如 MQ_INSTxxx_aaa%TopicTest。
Group	Group 的名称，在控制台 Group 管理页面复制。此处需拼接命名空间名称，格式为 namespace全称%group名称 ，例如 MQ_INSTxxx_aaa%GroupTest。
NameServerAddress	集群接入地址，控制台集群基本信息页面的接入信息模块获取。
SecretKey	角色名称，在控制台的集群权限页面 SecretKey 列复制。
AccessKey	角色密钥，在控制台的集群权限页面 AccessKey 列复制。

实践教程
RocketMQ 常见概念命名规范

本文介绍在使用 RocketMQ 过程中常见概念的命名规范以及使用规范。

命名规范

topic

- 不能为空，只能包含字母、数字、“%”、“-”及“_”，3-100 字符。
- 建议格式：String.format("tp_%s_%s", "系统名","业务名")。
- 例如：tp_order_check。

tag

- 允许为空，只要是字符串就可以，用来做 topic 下的二级消息过滤。
- 建议格式：String.format（"tag_%s","业务动作或类别"）。
- 例如：tag+业务动作，例如：订单创建的 tag为：tag_create；订单关闭的tag为：tag_close。

keys

- 允许为空，建议设置，只要是字符串或字符串数组都可以，用来在控制台查询消息或查询消息轨迹。
- 建议格式：String.format("%s","业务唯一性 ID")。
- 例如：订单 ID，交易 ID 或流水号，TraceID 等唯一性 ID。

producer group

- 不能为空，3-64个字符，只能包含字母、数字、“-”及“_”。
- 建议格式：String.format("pg_%s_%s", "系统名","业务名")。
- 例如：pg_transfer_check。

consumer group

- 不能为空，3-64个字符，只能包含字母、数字、“-”及“_”。
- 建议格式：String.format("cg_%s_%s", "系统名","业务名")。
- 例如：cg_transfer_check。

role

- 不能为空，只支持数字、大小写字母、分隔符（"_"、"-"），不能超过 32 个字符。
- 不同业务读写权限的标记，建议格式：业务名 + 发送/消费。

- 例如：role_order_send, role_order_consume，role_order_all。

clientId

clientId 表示一个客户端实例 ID，不同客户端不可重复。clientId 不能在客户端直接设置，instanceName 是 clientId 的可选组成部分，可以通过调整instanceName 修改 clientId。

分类	生成策略
生产者	${当前IP}@${instanceName}
集群消费者	${当前IP}@${instanceName}
广播消费者	${当前IP}@${instanceName}

```
public String buildMQClientId() {
    StringBuilder sb = new StringBuilder();
    sb.append(this.getClientIP());

    sb.append("@");
    sb.append(this.getInstanceName());
    if (!UtilAll.isBlank(this.unitName)) {
        sb.append("@");
        sb.append(this.unitName);
    }

    if (enableStreamRequestType) {
        sb.append("@");
        sb.append(RequestType.STREAM);
    }

    return sb.toString();
}
```

instanceName

实例名，默认场景不需要用户特殊设置，默认系统会通过以下代码随机生成一个唯一值。

```
public void changeInstanceNameToPID() {
    if (this.instanceName.equals("DEFAULT")) {
        this.instanceName = UtilAll.getPid() + "#" + System.nanoTime();
    }
}
```

广播消费者在每次启动时，需要保持消费者实例名不变，读取客户端本地的进度文件，需要显式地设置 instanceName，并且广播消费需要保持当前客户端 IP 启动前后不变， 如果容器部署，需要设置固定 pod IP 调度，否则重启期间的广播消息会漏消费。
建议格式：String.format("instance-%s-%s","系统名","业务名")。

使用规范

生产者

- 【强制】一个领域服务只能有一个 topic。
- 【强制】领域服务发送消息时必须根据业务动作设置 tag。
- 【强制】在 producer 发送消息时必须设置 keys。
- 【强制】消息发送成功或者失败要打印消息日志，务必要打印 SendResult 和 key 字段。
- 【建议】消息发送失败后建议将消息存储到 db，然后由定时器类线程进行定时重试，确保消息达到 broker。
- 【建议】对于可靠性要求不高的业务场景可以使用 oneway 消息。
- 【强制】新建生产者时必须指定生产者分组。

消费者

- 【强制】新建消费者时必须指定消费者分组。
- 【强制】消息消费者无法避免消息重复，所以需要业务服务来保证消息消费幂等。
- 【建议】为了提高消费并行度，可以在同一个 ConsumerGroup 下启动多个 Consumer 实例或者通过修改 ConsumeThreadMin 和consumeThreadMax 来提高单个 Consumer 的并行消费能力。
- 【建议】为了增加业务吞吐量，可以通过设置 consumer 的 consumeMessageBatchMaxSize 来批量消费消息。
- 【建议】发生消息堆积时，如果业务对数据要求不高时，可以选择丢弃不重要的消息。
- 【建议】如果消息量较少，建议在消费入口打印消息、消费耗时等，方便后续排查问题。

RocketMQ 客户端实践

发送方实践

选择 Topic 还是 Tags

Topic 是消息主题，用来区分同一类型的业务消息，Tag 是消息的一个特殊属性或者叫子类型，下游可以通过过滤表达式在 Broker 端做高效的消费过滤。
选用 Topic 还是 Tag 可以通过以下几方面判断：

1. 消息类型是否一致：目前普通消息、定时消息、事务消息、顺序消息是不能混用的，如果不一样一定要选用不同的 Topic。
2. 业务场景是否一致：例如订单消息和支付消息，建议选用不同的 Topic，如果订单消息下游有不同省份地域过滤的诉求，可以将城市标记为 Tag。
3. 消息量是否一致：因为 RocketMQ 消费进度按 Topic 维护的，如果不同类型之前量级差异非常大，量小的 Tag 要过滤掉大量无用消息才能消费到，造成无效资源消耗和消费延迟增加。
   综合看，推荐 Topic 类区分没有业务关联的消息，使用 Tag 来区分同一类 topic 中的关键属性，方便不同下游高效的过滤处理部分消息。

Keys 使用

推荐给将每个消息在业务层面的唯一业务标识码设置到 Key 字段，服务器会为每个消息创建索引（哈希索引），用户可以通过 Topic、Key 来查询这条消息内容，以及消息轨迹，方便排查问题。

选择合适的发送方式

RocketMQ 支持三种发送方式：

- 同步发送：适用于可靠性要求比较高的场景，如支付消息、短信通知等。
- 异步发送：适用于对响应时间敏感的业务场景，即发送端不能容忍长时间地等待 Broker 的响应。
- 单向发送：适用于不特别关心发送结果的场景，例如日志发送。
  send(msg)同步发送只有发送成功才会返回结果，但会同步阻塞，增大发送耗时，如果有性能要求，可以使用异步的方式:send(msg, callback)，在回调中检查发送结果，某些应用如果不关注消息是否发送成功，例如日志收集类应用，请直接使用 sendOneWay 方法发送消息。

发送重试策略

对于消息不可丢失应用，务必要有消息重发机制，Producer的send方法本身支持内部重试:

1. 至多重试2次（同步发送为2次，异步发送为0次）。
2. 如果发送失败，则轮转到下一个 Broker。这个方法的总耗时时间不超过 sendMsgTimeout 设置的值，默认3s。

3. 如果本身向 broker 发送消息产生超时异常，就不会再重试。
   如果是关键业务消息，消息发送失败后建议将消息存储到 db，然后由定时器类线程进行定时重试，确保 RocketMQ 服务恢复后消息重新发送到下游消费者。

打印发送消息日志

无论发送成功还是失败，都推荐将发送结果 sendresult、msgid、key、tag 等属性都打印到日志中，方便排查定位问题。

消费方实践教程

使用消费组隔离不同的下游业务

不同的消费业务可以使用消费组独立地消费同样的主题，并且每个消费者都有自己的消费偏移量(offsets)，确保同一组中的每个消费者订阅相同的主题，使用相同的过滤规则。

消费幂等处理

RocketMQ 在原理上保证至少消费一次，无法避免消息重复，例如发送时网络抖动会造成重试，投递时消费异常也会重试，以及消费者重启变化时负载均衡也会造成消息重复，所以如果业务对消费重复非常敏感，务必要在业务层面进行去重处理。可以借助分布式缓存或关系数据库进行去重。
幂等处理时，首先需要确定消息的唯一键，不推荐依赖 msgid，推荐使用设置到 keys 字段的唯一业务标识字段，例如订单Id等，在消费之前判断唯一键是否在分布式缓存或关系数据库中存在，如果不存在则插入，并消费，否则跳过，或者根据业务逻辑做进一步去重处理。

消费失败重试

并发消费：并发消费消费失败采用的是退避重试的机制，消费失败的消息将发回系统延时队列，每一次消费失败，delayLevel 将 +1，默认最大重试次数为 16 次，超过最大次数将进入死信队列。重试 16 次的时间间隔分别对应延时消息的 level3 ~ level18，时间为 10s ~ 2h，详见消息重试。
顺序消费：顺序消费为了保证顺序，不会发回服务端，采用的是本地不断重试的机制，默认重试消费次数为 Integer.MAX_VALUE, 可设置最大重试次数，超过最大次数将进入死信队列。
也可以通过 maxReconsumeTimes 配置最大重试次数。

提升消费并行度

大部分消息消费行为都属于 IO 密集型，即可能在处理消息过程中操作数据库或者调用 RPC，这类消费行为的消费速度在于后端数据库或者外系统的吞吐量，通过增加消费并行度，可以提高总的消费吞吐量。

1. 增加消费节点个数，同一个 ConsumerGroup 下，通过增加 Consumer 实例节点数量来提高并行度（需要注意的是超过订阅队列数的 Consumer 实例无效）。可以通过加机器，或者在已有机器启动多个进程的方式。
2. 增加单个消费节点的线程数，通过修改Consumer 的参数 consumeThreadMin、consumeThreadMax，增加并发线程个数实现更高的并发度。

3. 批量方式消费——某些业务流程如果支持批量方式消费，则可以很大程度上提高消费吞吐量，例如订单扣款类应用，一次处理一个订单耗时 1 s，一次处理 10 个订单可能也只耗时 2 s，这样即可大幅度提高消费的吞吐量，通过设置 consumer的 consumeMessageBatchMaxSize 返个参数，默认是 1，即一次只消费一条消息，例如设置为 N，那么每次消费的消息数小于等于 N。
4. 跳过非关键的消息，发生消息堆积时，如果业务对数据要求不高时，可以通过根据时间过滤跳过过期消息，或根据业务选择丢弃不重要的消息，提升消息处理效率。

订阅关系一致性

一个 ConsumerGroup 通常代表一组消费业务逻辑一致的多个节点，订阅关系一致指的是同一个 Consumer Group 下所有 Consumer 实例的订阅关系一致，否则消息消费的逻辑可能会混乱，甚至导致消息丢失。
尽量保证订阅关系变化的兼容性，建议不要变化订阅 Topic， 过滤表达式只做增量订阅，在同一个 Group 不同的消费节点发布的过程中，就会出现不同消费节点之间过滤规则不一致，Broker 过滤消息时，不同消费节点生效不同的过滤规则，所以同一个 Group 的订阅关系一旦变化，要关注订阅关系的兼容性，避免漏收到消息。
RocketMQ 管控台在消费者详情页面提供了 订阅关系不一致的诊断功能，可以看到哪些节点出现了订阅关系不一致的情况。

打印消费消息日志

无论发送成功还是失败，都推荐收到消息先打印一条日志，包括 msgid、key、tag 等属性都打印到日志中，方便排查定位问题，等消息处理完成，再将msgid、key、tag，重试次数和消费结果也打印到日志中。

其他消费建议

">关于消费者和订阅

第一件需要注意的事情是，不同的消费者组可以独立的消费一些 Topic，并且每个消费者组都有自己的消费偏移量，请确保同一组内的每个消费者订阅信息保持一致。

">关于有序消息

消费者将锁定每个消息队列，以确保他们被逐个消费，虽然这将会导致性能下降，但是当您关心消息顺序的时候会很有用。我们不建议抛出异常，您可以返回 ConsumeOrderlyStatus.SUSPEND_CURRENT_QUEUE_A_MOMENT 作为替代。

">关于并发消费

顾名思义，消费者将并发消费这些消息，建议您使用它来获得良好性能，我们不建议抛出异常，您可以返回 ConsumeConcurrentlyStatus.RECONSUME_LATER 作为替代。

">关于消费状态 Consume Status

对于并发的消费监听器，您可以返回 RECONSUME_LATER 来通知消费者现在不能消费这条消息，并且希望可以稍后重新消费它。然后，您可以继续消费其他消息。对于有序的消息监听器，因为您关心它的顺序，所以不能跳过消息，但是您可以返回SUSPEND_CURRENT_QUEUE_A_MOMENT 告诉消费者等待片刻。

">关于 Blocking

不建议阻塞监听器，因为它会阻塞线程池，并最终可能会终止消费进程。

">关于消费位点

当建立一个新的消费者组时，需要决定是否需要消费已经存在于 Broker 中的历史消息
CONSUME_FROM_LAST_OFFSET 将会忽略历史消息，并消费之后生成的任何消息。
CONSUME_FROM_FIRST_OFFSET 将会消费每个存在于 Broker 中的信息。您也可以使用
CONSUME_FROM_TIMESTAMP 来消费在指定时间戳后产生的消息。

问题排查实践

SDK日志

RocketMQ 客户端与服务端的通信协议是比较复杂的，例如消费队列的分配，主题的寻址，都是客户端 SDK 的实现决定的，这些过程的重要信息都保存在SDK 日志里。因此，当出现生产或消费问题时，SDK 日志是排查问题的最重要手段之一，请务必要保存这些日志。通常情况，SDK 日志不与业务日志打印在同一个文件中，我们给出常用 SDK 的日志默认路径，如下：

SDK 语言	SDK 协议	路径
Java	remoting	~/logs/rocketmqlogs/rocketmq_client.log
Java	grpc	~/logs/rocketmq/rocketmq_client.log
Go	remoting	/tmp/rocketmq-client.log
Go	grpc	~/logs/rocketmqlogs/rocketmq_client_go.log

客户端风险说明和更新指南

1. Java 的 PushConsumer 顺序消息在特定场景下无法保证顺序

客户端列表

5.0.6及以下版本在一些特殊场景下可能会遇到这个问题，在一些网络异常、超时等的场景下，无法保证消息顺序。
建议升级到5.0.7，直接升级依赖版本即可。

2. golang 版本低，路由表可能被清空

客户端列表

v5.0.1及以下在一些网络超时的场景下，由于获取不到路由表，进行了路由表清空。

升级客户端操作指南

建议升级到v5.1.1-rc1，直接升级依赖版本即可。

3. Java 消息轨迹可能不完整

客户端列表

4.9.4及以下在一些特殊场景下可能会遇到。

升级客户端操作指南

建议升级到4.9.5，直接升级依赖版本即可。

4. Go SDK 首次启动可能报错 fetch offset of mq from broker error

客户端列表

master 分支之下都可能遇到，比如 v2.1.2 及以下，启动报错 fetch offset of mq from broker error 。

升级客户端操作指南

建议升级到最新的 master branch 作为 Commit ID 版本即可。

5. Java SDK 消费过程中出现报错 the queue's messages, span too long, so do flow control

客户端列表

4.7.1及以下在一些特殊场景下可能会遇到，在消费过程中，日志中打印如上日志，并且消费会被限流。

升级客户端操作指南

建议升级到4.9.5，直接升级依赖版本即可。

常见问题
控制台常见问题

在消息队列 RocketMQ 版控制台上查询不到消息

- 如果所有消息都查询不到，请联系技术支持处理。
- 如果是部分消息查询不到，消息查询有几秒钟的延迟，且消息查询只支持查询三天内的消息，若是其他的情况请技术支持处理。

客户端接入问题

客户端返回“ No route info of this topic ”错误

详细报错如下：
[org.apache.rocketmq.client.exception.MQClientException: No route info of this topic: generation%rocketmq-gab9bb2877z8|generation%video_task
可能原因如下：
topic_name 不对，检查一下命名空间是否正确，主题名的格式应该是rocketmq-xxx|[ns]%[topicName]

- namespace 为命名空间的名称，在控制台命名空间页面复制。
- topic_name 在控制台集群管理中Topic 页签中复制具体 Topic 名称。
  另外，其他可能的原因是：
- Broker 禁止自动创建 Topic，且用户没有通过手工方式在控制台创建 Topic。
- Broker 没有正确连接到 Name Server，请提交工单处理。
- Producer 没有正确连接到 Name Server，请检查接入点是否输入正确。

客户端异常“ Send [3] times, still failed, cost [334]ms ”

详细报错如下：
com.freesky.inspiration.domain.mq.MQProducerManager:sendMessage``[org.apache.rocketmq.client.exception.MQClientException: Send [3] times, still failed, cost [334]ms, Topic: video_task, BrokersSent: [broker-2, broker-0, broker-1]
可能原因如下：
缺少权限，检查一下命名空间有没有授权。

发送消息报错

客户 Producer 发消息失败，出现 ACL exception 的报错

Caused by: org.apache.rocketmq.client.exception.MQBrokerException: CODE: 1 DESC: org.apache.rocketmq.acl.common.AclException: [PRODUCE] Token authentication failed, please check, org.streamnative.pulsar.handlers.rocketmq.inner.RocketMQBrokerController$3.doBeforeRequest(RocketMQBrokerController.java:475)
For more information, please visit the url, http://rocketmq.apache.org/docs/faq/
at org.apache.rocketmq.client.impl.MQClientAPIImpl.processSendResponse(MQClientAPIImpl.java:665)
at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessageSync(MQClientAPIImpl.java:505)
at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessage(MQClientAPIImpl.java:487)
at org.apache.rocketmq.client.impl.MQClientAPIImpl.sendMessage(MQClientAPIImpl.java:431)
at org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl.sendKernelImpl(DefaultMQProducerImpl.java:854)
at org.apache.rocketmq.client.impl.producer.DefaultMQProducerImpl.sendDefaultImpl(DefaultMQProducerImpl.java:584)
... 3 more

可能原因如下：

AK SK 配置不正确。access-key/secret-key 配置不正确会抛出 AclException 。

- access-key：namespace 配置权限中的角色密钥。
- secret-key：namespace 配置权限中的角色名称。

客户接入的时候报错 RemotingConnectException

Caused by: org.apache.rocketmq.remoting.exception.RemotingConnectException: connect to [rocketmq-vsxssa.rocketmq.ap-gz.qcloud.tdmq.com:5098] failed
at org.apache.rocketmq.remoting.netty.NettyRemotingClient.getAndCreateNameserverChannel(NettyRemotingClient.java:445)
at org.apache.rocketmq.remoting.netty.NettyRemotingClient.getAndCreateChannel(NettyRemotingClient.java:400)
at org.apache.rocketmq.remoting.netty.NettyRemotingClient.invokeSync(NettyRemotingClient.java:369)
at org.apache.rocketmq.client.impl.MQClientAPIImpl.getTopicRouteInfoFromNameServer(MQClientAPIImpl.java:1363)
at org.apache.rocketmq.client.impl.MQClientAPIImpl.getTopicRouteInfoFromNameServer(MQClientAPIImpl.java:1353)
at org.apache.rocketmq.client.impl.factory.MQClientInstance.updateTopicRouteInfoFromNameServer(MQClientInstance.java:622)
... 6 more

大部分时候是接入点填写错误，接入点可以在集群基本信息页面的网络模块获取。

> VPC 内网访问时，需要使用同一地域的 CVM 机器或者其他能成功访问 VPC 的环境。可使用 ping 或者 telnet 测试网络是否成功连接，网络不通时会抛出 RemotingConnectException 错误。

消费消息报错

消费不到消息是什么原因？

消费不到消息有多种原因，常见的如下：

- 生产者启动在消费者启动之前：生产者启动在消费者启动之前，但是消费者 consumeFromWhere 配置错误会导致消费者从最新的位点开始消费。
- 多个生产者使用同一个消费者组：多个生产者使用了同一个消费者组，但是订阅的 TAG 不同，也会引起消息推送错误，从而导致消息消费不到。

消息拉取频率介绍

RocketMQ 有两种常用的消费模式，Pull 和 Push，而 Push 模式的实现是基于 Pull 模式，只不过对 Pull 模式做了个包装，所以 RocketMQ Push 模式并不是真正意义上的推模式，还是拉模式。
可以认为 RocketMQ SDK 会不断的进行消息拉取任务。对于每一次拉取的结果：

- 当从 broker 拉取到消息后，如果消息被过滤掉（例如 tag 过滤），则继续将拉取请求放入阻塞队列中继续循环执行消息拉取任务，否则将消息放入消费者消费线程去执行，而将拉取请求放入阻塞队列中。
- 如果从 broker 端没有可拉取的新消息或者没有匹配到消息，则将 pullRequest 放入阻塞队列中继续循环执行消息拉取任务。
  从这里可以看到，如果用户没有消息生产，只要有消费者在线，那么都会周期性的向服务端请求，这种请求也能保证消息消费的及时性。如果用户想减少拉取次数，

如何处理

- 对于 Pull 模式的消费者，由于用户完全自主控制拉取频率，所以用户只需要自己在拉取一次之后，进行对应的等待，或者使用对应的限流组件来控制。

- 而对于 Push 模式的消费者，那么可以设置 PullInterval 拉长拉取间隔。

```
// 设置拉取间隔 60秒
pushConsumer.setPullInterval(60 * 1000);
```

> 注意：
>
> 但是相应的需要注意，消息消费的延迟可能上升，修改需要谨慎。

- 如果用户生产消息量很少，基本上会命中长轮训的时间，目前 SDK 固定了长轮训是15秒，也就是说，对于一个 queue，一个 broker，每分钟最少会发起4次拉取。如果有10个 Topic，每个 Topic 3个队列，服务端有2个 broker，那么这个拉取数量为10*32*4=240，也就是每秒4个拉取。
- 运行过程中，会随着用户生产消费，可能产生重试 Topic，这时候，因为要拉取重试队列，所以消费拉取频率还会上升。

术语
基本概念

本文列举了 TDMQ RocketMQ 版中常见的概念与定义。

消息（Message）

消息系统所传输信息的物理载体，生产和消费数据的最小单位。生产者将业务数据的负载和拓展属性包装成消息发送到服务端，服务端按照相关语义将消息投递到消费端进行消费。

主题（Topic）

Topic 表示一类消息的集合，每个主题包含若干消息，是 RocketMQ 进行消息订阅的基本单位。

消息标签（MessageTag）

为消息设置的标签，用于将同一个 Topic 下区分不同类型的消息，可以理解为 Topic 是消息的一级分类，Tag 是消息的二级分类。

消息队列（MessageQueue）

存储消息的物理实体，一个 Topic 可以包含多个 Queue，Queue 也叫消息分区，一个 Queue 中的消息只能被一个消费者组中的一个消费者消费，一个 Queue 中的消息不允许同一个消费者组中的多个消费者同时消费。

">消息位点（MessageQueueOffset）

消息是按到达 RocketMQ 服务端的先后顺序存储在指定主题的多个队列中，每条消息在队列中都有一个唯一的 Long 类型坐标，这个坐标被定义为消息位点。

">消费位点（ConsumerOffset）

一条消息被某个消费者消费完成后不会立即从队列中删除， RocketMQ 会基于每个消费者分组记录消费过的最新一条消息的位点，即消费位点。

">消息索引（MessageKey）

消息索引是 RocketMQ 提供的面向消息的索引属性。通过设置的消息索引可以快速查找到对应的消息内容。

">生产者（Producer）

生产者是 RocketMQ 系统中用来构建并传输消息到服务端的运行实体。生产者通常被集成在业务系统中，将业务消息按照要求封装成消息并发送至服务端。

">消费者（Consumer）

消费者是 RocketMQ 中用来接收并处理消息的运行实体。消费者通常被集成在业务系统中，从服务端获取消息，并将消息转化成业务可理解的信息，供业务逻辑处理。

分组（Group）

可分为生产者组和消费者组：

- 生产者组：同一类 Producer 的集合，这类 Producer 发送同一类消息且发送逻辑一致。如果发送的是事务消息，且生产者发送后崩溃，则 Broker 服务器会联系同一个生产者组的其他生产者实例以提交或者回溯消费。
- 消费者组：同一类 Consumer 的集合，这类 Consumer 通常消费同一类消息且消费逻辑一致。消费者组使得在消息消费方面实现了负载均衡和容错。消费者组的消费者实例必须订阅完全相同的 Topic。

">消息类型（MessageType）

按照消息传输特性的不同而定义的分类，用于类型管理和安全校验。RocketMQ 支持的消息类型有普通消息、顺序消息、事务消息和定时/延时消息。

普通消息

普通消息是一种基础的消息类型，由生产投递到指定 Topic 后，被订阅了该 Topic 的消费者所消费。普通消息的 Topic 中无顺序的概念，可以使用多个分区数来提升消息的生产和消费效率，在吞吐量巨大时其性能最好。

顺序消息

顺序消息是消息队列 RocketMQ 提供的一种高级消息类型，对于一个指定的Topic，消息严格按照先进先出（FIFO）的原则进行消息发布和消费，即先发送的消息先消费，后发送的消息后消费。

重试队列

重试队列是一种为了确保消息被正常消费而设计的队列。当某些消息第一次被消费者消费后，没有得到正常的回应，则会进入重试队列，当重试达到一定次数后，停止重试，投递到死信队列中。
由于实际场景中，可能会存在的一些临时短暂的问题（如网络抖动，服务重启等）导致消息无法及时被处理，但短暂时间过后又恢复正常。这种场景下，重试队列的重试机制就可以很好解决此类问题。

死信队列

死信队列是一种特殊的消息队列，用于集中处理无法被正常消费的消息的队列。当消息在重试队列中达到一定重试次数后仍未能被正常消费，TDMQ 会判定这条消息在当前情况下无法被消费，将其投递至死信队列。
实际场景中，消息可能会由于持续一段时间的服务宕机，网络断连而无法被消费。这种场景下，消息不会被立刻丢弃，死信队列会对这种消息进行较为长期的持久化，用户可以在找到对应解决方案后，创建消费者订阅死信队列来完成对当时无法处理消息的处理。

集群消费

集群消费：当使用集群消费模式时，任意一条消息只需要被集群内的任意一个消费者处理即可。适用于每条消息只需要被处理一次的场景。

广播消费

广播消费：当使用广播消费模式时，每条消息会被推送给集群内所有注册过的消费者，保证消息至少被每个消费者消费一次。适用于每条消息需要被集群下每一个消费者处理的场景。

">消息过滤

消费者可以通过订阅指定消息标签（Tag）对消息进行过滤，确保最终只接收被过滤后的消息合集。过滤规则的计算和匹配在 RocketMQ 的服务端完成。

">重置消费位点

以时间轴为坐标，在消息持久化存储的时间范围内，重新设置消费者分组对已订阅主题的消费进度，设置完成后消费者将接收设定时间点之后，由生产者发送到 RocketMQ 服务端的消息。

">消息轨迹

在一条消息从生产者发出到消费者接收并处理过程中，由各个相关节点的时间、地点等数据汇聚而成的完整链路信息。通过消息轨迹，您能清晰定位消息从生产者发出，经由 RocketMQ 服务端，投递给消费者的完整链路，方便定位排查问题。

">消息堆积

生产者已经将消息发送到 RocketMQ 的服务端，但由于消费者的消费能力有限，未能在短时间内将所有消息正确消费掉，此时在服务端保存着未被消费的消息，该状态即消息堆积。

版本说明书
版本说明书

正式发布消息队列 RocketMQ 版首个版本。
消息队列 RocketMQ 版（TDMQ for RocketMQ，简称 TDMQ RocketMQ 版）是云平台基于 Apache RocketMQ 构建的分布式消息中间件，完全兼容 Apache RocketMQ 的各个组件与概念，支持开源社区版本的客户端零改造接入。
消息队列 RocketMQ 版具有低延迟、高性能、高可靠、万亿级消息容量和灵活可扩展等特点，可为分布式应用系统提供异步解耦和削峰填谷的能力，同时也具备互联网应用所需的海量消息堆积、高吞吐、可靠重试等特性。主要产品优势如下：

兼容开源

TDMQ RocketMQ 版兼容开源 RocketMQ 4.3.0及以上版本，支持开源 Java、C/C++、Go 等多语言客户端接入。

资源隔离

多层级的资源结构，不仅基于命名空间做了虚拟隔离，也可以在集群维度做物理隔离。支持在命名空间维度为客户端配置权限校验，区分不同环境的客户端，方便灵活。

丰富的消息类型

支持普通消息、顺序消息、延时消息和事务消息等多种消息类型，支持消息重试和死信机制，满足各类业务场景。

高性能

单机最高可支持上万级别的生产消费吞吐，分布式架构，无状态服务，可以横向扩容来增强整个集群的吞吐。

可观测性

控制台提供丰富的监控指标，消息轨迹可视化展示，对接云平台的监控和告警功能，同时提供完善的云 API，支持集成自助运维系统。

API文档
（tdmq）
版本（2020-02-17）
API 概览

API版本

V3

CMQ管理相关接口

接口名称	接口功能
RewindCmqQueue	回溯cmq队列

RocketMQ 主题相关接口

接口名称	接口功能
CreateRocketMQTopic	创建RocketMQ主题
DeleteRocketMQTopic	删除RocketMQ主题
DescribeRocketMQTopics	获取RocketMQ主题列表
ModifyRocketMQTopic	更新RocketMQ主题信息

RocketMQ 命名空间相关接口

接口名称	接口功能
CreateRocketMQNamespace	创建RocketMQ命名空间
DeleteRocketMQNamespace	删除RocketMQ命名空间

接口名称	接口功能
DescribeRocketMQNamespaces	获取RocketMQ命名空间列表
ModifyRocketMQNamespace	更新RocketMQ命名空间

RocketMQ 平滑迁移相关接口

接口名称	接口功能
ImportRocketMQConsumerGroups	平滑迁移：导入消费者组列表
ImportRocketMQTopics	平滑迁移：导入topic列表

RocketMQ 消息相关接口

接口名称	接口功能
DescribeRocketMQMsg	rocketmq消息详情
DescribeRocketMQMsgTrace	查询RocketMQ消息轨迹
DescribeRocketMQTopicMsgs	rocketmq 消息查询
ExportRocketMQMessageDetail	导出RocketMQ消息详情
ResetRocketMQConsumerOffSet	重置RocketMQ消费位点
SendRocketMQMessage	发送RocketMQ消息
VerifyRocketMQConsume	Rocketmq消费验证

RocketMQ 消费组相关接口

接口名称	接口功能
CreateRocketMQGroup	创建RocketMQ消费组
DeleteRocketMQGroup	删除RocketMQ消费组
DescribeRocketMQConsumerConnectionDetail	获取在线消费端详情

接口名称	接口功能
DescribeRocketMQConsumerConnections	获取指定消费组下当前客户端的连接情况
DescribeRocketMQGroups	获取RocketMQ消费组列表
DescribeRocketMQSubscriptions	获取RocketMQ消费组订阅关系
DescribeRocketMQTopicsByGroup	获取指定消费组下订阅的主题列表
ModifyRocketMQGroup	更新RocketMQ消费组信息

RocketMQ 角色授权相关接口

接口名称	接口功能
CreateRocketMQEnvironmentRole	创建RocketMQ环境角色授权
CreateRocketMQRole	创建RocketMQ角色
DeleteRocketMQEnvironmentRoles	批量删除RocketMQ环境角色授权
DeleteRocketMQRoles	批量删除RocketMQ角色
DescribeRocketMQEnvironmentRoles	获取RocketMQ命名空间角色列表
DescribeRocketMQRoles	获取RocketMQ角色列表
ModifyRocketMQEnvironmentRole	修改RocketMQ环境角色授权
ModifyRocketMQRole	RocketMQ角色修改

RocketMQ 集群相关接口

接口名称	接口功能
CreateRocketMQCluster	创建RocketMQ集群
DeleteRocketMQCluster	删除RocketMQ集群
DescribeRocketMQCluster	获取单个RocketMQ集群信息
DescribeRocketMQClusters	获取RocketMQ的集群列表
DescribeRocketMQPublicAccessPoint	查询RocketMQ实例公网接入点信息

接口名称	接口功能
DescribeRocketMQVipInstances	查询RocketMQ专享实例列表
ModifyRocketMQCluster	更新RocketMQ集群信息

主题相关接口

接口名称	接口功能
CreateTopic	新增主题
DeleteTopics	删除主题
DescribeTopics	查询主题列表
ModifyTopic	修改主题

其他接口

接口名称	接口功能
DeleteRoles	删除角色
DescribeBindVpcs	获取租户VPC绑定关系
SendMsg	发送消息

命名空间相关接口

接口名称	接口功能
CreateEnvironment	创建命名空间
DeleteEnvironments	删除命名空间
DescribeEnvironmentAttributes	获取命名空间属性
DescribeEnvironmentRoles	获取命名空间角色列表
DescribeEnvironments	获取命名空间列表

接口名称	接口功能
ModifyEnvironmentAttributes	修改命名空间属性

消息相关接口

接口名称	接口功能
AcknowledgeMessage	确认消息
ReceiveMessage	接收消息
SendBatchMessages	批量发送消息
SendMessages	发送单条消息

环境角色授权相关接口

接口名称	接口功能
CreateEnvironmentRole	创建环境角色授权
CreateRole	创建角色
DeleteEnvironmentRoles	删除环境角色授权
DescribeRoles	获取角色列表
ModifyEnvironmentRole	修改环境角色授权
ModifyRole	角色修改
TpoModifyResourceProject	修改资源对应的项目（包含新增）

生产消费相关接口

接口名称	接口功能
CreateSubscription	创建订阅关系
DeleteSubscriptions	删除订阅关系

接口名称	接口功能
DescribeSubscriptions	消费订阅列表
ResetMsgSubOffsetByTimestamp	消息回溯

集群相关接口

接口名称	接口功能
CreateCluster	创建集群
DeleteCluster	删除集群
DescribeBindClusters	获取专享集群列表
DescribeClusterDetail	获取集群详情
DescribeClusters	获取集群列表
ModifyCluster	更新集群信息

调用方式
接口签名v1

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（ Signature ）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

参数名称	中文	参数值

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 Action=DescribeInstances 。
注意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.finance.cloud.tencent.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 / 。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。
首先使用 HMAC-SHA1 算法对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的TCloudFinanceZone SDK 3.0 ，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java

- PHP
- Go
- Node

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK 。

最终输出的 url 可能为： https://cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url ，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp 。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.finance.cloud.tencent.com/?");
```

```
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.finance.cloud.tencent.com/?");
        // 实际请求的url中对参数顺序没有要求
        for (String k : params.keySet()) {
            // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
        params.put("Action", "DescribeInstances"); // 公共参数
        params.put("Version", "2017-03-12"); // 公共参数
        params.put("Region", "shjr"); // 公共参数
        params.put("Limit", 20); // 业务参数
        params.put("Offset", 0); // 业务参数
        params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： pip install requests 。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

接口签名v3

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
    HTTPRequestMethod + '\n' +
    CanonicalURI + '\n' +
    CanonicalQueryString + '\n' +
    CanonicalHeaders + '\n' +
    SignedHeaders + '\n' +
    HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- CanonicalURI：URI 参数，API 3.0 固定为正斜杠（/）；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1）头部 key 统一转成小写；2）多个头部 key（小写）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
    Algorithm + \n +
```

```
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

> 1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
> 2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // 注意时区，否则容易出错
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* 步骤 1：拼接规范请求串 *************
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* 步骤 2：拼接待签名字符串 *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* 步骤 4：拼接 Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

```
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)

# ************* 步骤 3：计算签名 *************
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* 步骤 4：拼接 Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

TCloudFinanceZone API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data（仅部分接口支持），必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0， InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

公共参数

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

CMQ消息相关接口
清空cmq消息队列中的消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

清空cmq消息队列中的消息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:35:10。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ClearCmqQueue
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-queue

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

清空cmq订阅者消息标签

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

清空订阅者消息标签

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:35:59。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ClearCmqSubscriptionFilterTags
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线（-）。 示例值： test-topic
SubscriptionName	是	否	String	订阅名字，在单个地域同一帐号的同一主题下唯一。订阅名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-queue

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

发送cmq主题消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送cmq主题消息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-09 14:38:20。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：PublishCmqMsg
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名 示例值： topic123
MsgContent	是	否	String	消息内容 示例值： testmessage
MsgTag	否	否	Array of String	消息标签 示例值： ["a"]

3. 输出参数

参数名称	类型	描述
Result	Bool	true表示发送成功 示例值： true

参数名称	类型	描述
MsgId	String	消息id 示例值： 1EAD5E05003A6767C1FC9160304F98
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

发送cmq消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送cmq消息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:21:23。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SendCmqMsg
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名 示例值： test-queue
MsgContent	是	否	String	消息内容 示例值： testmessage
DelaySeconds	是	否	Int64	延迟时间 示例值： 100

3. 输出参数

参数名称	类型	描述
Result	Bool	true表示发送成功 示例值： true

参数名称	类型	描述
MsgId	String	消息id 示例值： 09A51113003621B8D17C0DF3DC2403D
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

CMQ管理相关接口
创建cmq队列接口

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建cmq队列接口

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:02:18。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateCmqQueue
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过 64 个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： testQueue
MaxMsgHeapNum	否	否	Uint64	最大堆积消息数。取值范围在公测期间为 1,000,000 - 10,000,000，正式上线后范围可达到 1000,000-1000,000,000。默认取值在公测期间为 10,000,000，正式上线后为 100,000,000。 示例值： 1000000

参数名称	必选	允许NULL	类型	描述
PollingWaitSeconds	否	否	Uint64	消息接收长轮询等待时间。取值范围 0-30 秒，默认值 0。 示例值： 3
VisibilityTimeout	否	否	Uint64	消息可见性超时。取值范围 1-43200 秒（即12小时内），默认值 30。 示例值： 30
MaxMsgSize	否	否	Uint64	消息最大长度。取值范围 1024-65536 Byte（即1-64K），默认值 65536。 示例值： 1048576
MsgRetentionSeconds	否	否	Uint64	消息保留周期。取值范围 60-1296000 秒（1min-15天），默认值 345600 (4 天)。 示例值： 3600
RewindSeconds	否	否	Uint64	队列是否开启回溯消息能力，该参数取值范围0-msgRetentionSeconds,即最大的回溯时间为消息在队列中的保留周期，0表示不开启。 示例值： 0
Transaction	否	否	Uint64	1 表示事务队列，0 表示普通队列 示例值： 0
FirstQueryInterval	否	否	Uint64	第一次回查间隔 示例值： 0
MaxQueryCount	否	否	Uint64	最大回查次数 示例值： 0
DeadLetterQueueName	否	否	String	死信队列名称 示例值： test1
Policy	否	否	Uint64	死信策略。0为消息被多次消费未删除，1为Time-To-Live过期 示例值： 0
MaxReceiveCount	否	否	Uint64	最大接收次数 1-1000 示例值： 1
MaxTimeToLive	否	否	Uint64	policy为1时必选。最大未消费过期时间。范围300-43200，单位秒，需要小于消息最大保留时间msgRetentionSeconds 示例值： 300
Trace	否	否	Bool	是否开启消息轨迹追踪，当不设置字段时，默认为不开启，该字段为true表示开启，为false

参数名称	必选	允许NULL	类型	描述
				表示不开启 示例值： false

3. 输出参数

参数名称	类型	描述
QueueId	String	创建成功的queueId 示例值： cmqq-o9p8j9vp
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.CreateTopic	
ResourceUnavailable	
FailedOperation	
FailedOperation.CreateCluster	
InternalError.BrokerService	Broker服务异常。

创建cmq订阅接口

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建cmq订阅接口

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:00:53。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateCmqSubscribe
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线（-）。 示例值： ConnTopic
SubscriptionName	是	否	String	订阅名字，在单个地域同一帐号的同一主题下唯一。订阅名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： ConnSubQueue
Protocol	是	否	String	订阅的协议，目前支持两种协议：http、queue。使用http协议，用户需自己搭建接受消息的web server。使用queue，消息会自动推送到CMQ queue，用户可以并发地拉取消息。 示例值： queue

参数名称	必选	允许NULL	类型	描述
Endpoint	是	否	String	接收通知的Endpoint，根据协议Protocol区分：对于http，Endpoint必须以“ http:// ”开头，host可以是域名或IP；对于Queue，则填QueueName。 请注意，目前推送服务不能推送到私有网络中，因此Endpoint填写为私有网络域名或地址将接收不到推送的消息，目前支持推送到公网和基础网络。 示例值： queue_sub
NotifyStrategy	否	否	String	向Endpoint推送消息出现错误时，CMQ推送服务器的重试策略。取值有：1）BACKOFF_RETRY，退避重试。每隔一定时间重试一次，重试够一定次数后，就把该消息丢弃，继续推送下一条消息；2）EXPONENTIAL_DECAY_RETRY，指数衰退重试。每次重试的间隔是指数递增的，例如开始1s，后面是2s，4s，8s...由于Topic消息的周期是一天，所以最多重试一天就把消息丢弃。默认值是EXPONENTIAL_DECAY_RETRY。 示例值： EXPONENTIAL_DECAY_RETRY
FilterTag	否	否	Array of String	消息正文。消息标签（用于消息过滤)。标签数量不能超过5个，每个标签不超过16个字符。与(Batch)PublishMessage的MsgTag参数配合使用，规则：1）如果FilterTag没有设置，则无论MsgTag是否有设置，订阅接收所有发布到Topic的消息；2）如果FilterTag数组有值，则只有数组中至少有一个值在MsgTag数组中也存在时（即FilterTag和MsgTag有交集），订阅才接收该发布到Topic的消息；3）如果FilterTag数组有值，但MsgTag没设置，则不接收任何发布到Topic的消息，可以认为是2）的一种特例，此时FilterTag和MsgTag没有交集。规则整体的设计思想是以订阅者的意愿为主。 示例值： ["aa"]
BindingKey	否	否	Array of String	BindingKey数量不超过5个， 每个BindingKey长度不超过64字节，该字段表示订阅接收消息的过滤策略，每个BindingKey最多含有15个“.”， 即最多16个词组。 示例值： ["new"]
NotifyContentFormat	否	否	String	推送内容的格式。取值：1）JSON；2）SIMPLIFIED，即raw格式。如果Protocol是queue，则取值必须为SIMPLIFIED。如果Protocol是http，两个值均可以，默认值是

参数名称	必选	允许NULL	类型	描述
				JSON。 示例值： SIMPLIFIED

3. 输出参数

参数名称	类型	描述
SubscriptionId	String	订阅id 示例值： subsc-39gyuuhd
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

创建cmq主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建cmq主题

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:03:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateCmqTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线（-）。 示例值： test-topic
MaxMsgSize	否	否	Uint64	消息最大长度。取值范围 1024-65536 Byte（即1-64K），默认值 65536。 示例值： 65536
FilterType	否	否	Uint64	用于指定主题的消息匹配策略。1：表示标签匹配策略；2：表示路由匹配策略，默认值为标签匹配策略。 示例值： 1
MsgRetentionSeconds	否	否	Uint64	消息保存时间。取值范围60 - 86400 s（即1分钟 - 1天），默认值86400。 示例值： 86400

参数名称	必选	允许NULL	类型	描述
Trace	否	否	Bool	是否开启消息轨迹标识，true表示开启，false表示不开启，不填表示不开启。 示例值： false

3. 输出参数

参数名称	类型	描述
TopicId	String	主题id 示例值： topic-gzz05csc
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
FailedOperation.CreateCluster	

删除cmq队列

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除cmq队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:07:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteCmqQueue
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： testQueue

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.DeleteTopics	
FailedOperation	

删除cmq订阅

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除cmq订阅

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:34:27。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteCmqSubscribe
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-topic
SubscriptionName	是	否	String	订阅名字，在单个地域同一帐号的同一主题下唯一。订阅名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-sub

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

删除cmq主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除cmq主题

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:33:49。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteCmqTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-topic

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

枚举cmq死信队列源队列

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

枚举cmq死信队列源队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-05-17 11:44:59。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqDeadLetterSourceQueues
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
DeadLetterQueueName	是	否	String	死信队列名称 示例值： test
Limit	否	否	Uint64	分页时本页获取主题列表的起始位置。如果填写了该值，必须也要填写 limit 。该值缺省时，后台取默认值 0。 示例值： 10
Offset	否	否	Uint64	分页时本页获取主题的个数，如果不传递该参数，则该参数默认为20，最大值为50。 示例值： 0
SourceQueueName	否	否	String	根据SourceQueueName过滤 示例值： test

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	满足本次条件的队列个数 示例值： 1
QueueSet	Array of CmqDeadLetterSource	死信队列源队列 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询cmq队列详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询cmq队列详情

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:19:51。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqQueueDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	精确匹配QueueName 示例值： test-queue

3. 输出参数

参数名称	类型	描述
QueueDescribe	CmqQueue	队列详情列表。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

查询cmq全量队列

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询cmq全量队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:05:17。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqQueues
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	否	否	Uint64	分页时本页获取队列列表的起始位置。如果填写了该值，必须也要填写 limit 。该值缺省时，后台取默认值 0 示例值： 0
Limit	否	否	Uint64	分页时本页获取队列的个数，如果不传递该参数，则该参数默认为20，最大值为50。 示例值： 1
QueueName	否	否	String	根据QueueName进行过滤 示例值： testqueue
QueueNameList	否	否	Array of String	CMQ 队列名称列表过滤 示例值： testqueue1
IsTagFilter	否	否	Bool	标签过滤查找时，需要设置为 true 示例值： true

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	数量 示例值： 2
QueueList	Array of CmqQueue	队列列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

查询cmq订阅详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询cmq订阅详情

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:06:48。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqSubscriptionDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线（-）。 示例值： Conn
Offset	否	否	Uint64	分页时本页获取主题列表的起始位置。如果填写了该值，必须也要填写 limit 。该值缺省时，后台取默认值 0 示例值： 0
Limit	否	否	Uint64	分页时本页获取主题的个数，如果不传递该参数，则该参数默认为20，最大值为50。 示例值： 0
SubscriptionName	否	否	String	根据SubscriptionName进行模糊搜索 示例值： sub

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 1
SubscriptionSet	Array of CmqSubscription	Subscription属性集合 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询cmq主题详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询cmq主题详情

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:20:34。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqTopicDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	否	否	String	精确匹配TopicName。 示例值： topic-test

3. 输出参数

参数名称	类型	描述
TopicDescribe	CmqTopic	主题详情 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

枚举cmq全量主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

枚举cmq全量主题

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:06:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeCmqTopics
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	否	否	Uint64	分页时本页获取队列列表的起始位置。如果填写了该值，必须也要填写 limit 。该值缺省时，后台取默认值 0 示例值： 0
Limit	否	否	Uint64	分页时本页获取队列的个数，如果不传递该参数，则该参数默认为20，最大值为50。 示例值： 1
TopicName	否	否	String	根据TopicName进行模糊搜索 示例值： topic
TopicNameList	否	否	Array of String	CMQ 主题名称列表过滤 示例值： topic1
IsTagFilter	否	否	Bool	标签过滤查找时，需要设置为 true 示例值： true

3. 输出参数

参数名称	类型	描述
TopicList	Array of CmqTopic	主题列表 示例值：查看
TotalCount	Uint64	全量主题数量 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

修改cmq队列属性

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改cmq队列属性

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 16:58:34。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyCmqQueueAttribute
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过 64 个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： queue1
MaxMsgHeapNum	否	否	Uint64	最大堆积消息数。取值范围在公测期间为 1,000,000 - 10,000,000，正式上线后范围可达到 1000,000-1000,000,000。默认取值在公测期间为 10,000,000，正式上线后为 100,000,000。 示例值： 1000000
PollingWaitSeconds	否	否	Uint64	消息接收长轮询等待时间。取值范围 0-30 秒，默认值 0。 示例值： 3

参数名称	必选	允许NULL	类型	描述
VisibilityTimeout	否	否	Uint64	消息可见性超时。取值范围 1-43200 秒（即12小时内），默认值 30。 示例值： 30
MaxMsgSize	否	否	Uint64	消息最大长度。取值范围 1024-65536 Byte（即1-64K），默认值 65536。 示例值： 1024
MsgRetentionSeconds	否	否	Uint64	消息保留周期。取值范围 60-1296000 秒（1min-15天），默认值 345600 (4 天)。 示例值： 7200
RewindSeconds	否	否	Uint64	消息最长回溯时间，取值范围0-msgRetentionSeconds，消息的最大回溯之间为消息在队列中的保存周期，0表示不开启消息回溯。 示例值： 0
FirstQueryInterval	否	否	Uint64	第一次查询时间 示例值： 0
MaxQueryCount	否	否	Uint64	最大查询次数 示例值： 0
DeadLetterQueueName	否	否	String	死信队列名称 示例值： dead-queue
MaxTimeToLive	否	否	Uint64	MaxTimeToLivepolicy为1时必选。最大未消费过期时间。范围300-43200，单位秒，需要小于消息最大保留时间MsgRetentionSeconds 示例值： 300
MaxReceiveCount	否	否	Uint64	最大接收次数 示例值： 1
Policy	否	否	Uint64	死信队列策略 示例值： 0
Trace	否	否	Bool	是否开启消息轨迹标识，true表示开启，false表示不开启，不填表示不开启。 示例值： false
Transaction	否	否	Uint64	是否开启事务，1开启，0不开启 示例值： 0

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

修改cmq订阅属性

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改cmq订阅属性

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:00:09。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyCmqSubscriptionAttribute
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线（-）。 示例值： test-topic
SubscriptionName	是	否	String	订阅名字，在单个地域同一帐号的同一主题下唯一。订阅名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： tese-sub
NotifyStrategy	否	否	String	向 Endpoint 推送消息出现错误时，CMQ 推送服务器的重试策略。取值如下： （1）BACKOFF_RETRY，退避重试。每隔一定时间重试一次，重试够一定次数后，就把该消息丢弃，继续推送下一条消息。

参数名称	必选	允许NULL	类型	描述
				（2）EXPONENTIAL_DECAY_RETRY，指数衰退重试。每次重试的间隔是指数递增的，例如开始1s，后面是2s，4s，8s…由于 Topic 消息的周期是一天，所以最多重试一天就把消息丢弃。默认值是 EXPONENTIAL_DECAY_RETRY。 示例值： EXPONENTIAL_DECAY_RETRY
NotifyContentFormat	否	否	String	推送内容的格式。取值：（1）JSON；（2）SIMPLIFIED，即 raw 格式。如 果 Protocol 是 queue，则取值必须 为 SIMPLIFIED。如果 Protocol 是 HTTP，两个值均可以，默认值是 JSON。 示例值： SIMPLIFIED
FilterTags	否	否	Array of String	消息正文。消息标签（用于消息过滤)。标签数量不能超过5个，每个标签不超过16个字符。与(Batch)PublishMessage的MsgTag参数配合使用，规则：1）如果FilterTag没有设置，则无论MsgTag是否有设置，订阅接收所有发布到Topic的消息；2）如果FilterTag数组有值，则只有数组中至少有一个值在MsgTag数组中也存在时（即FilterTag和MsgTag有交集），订阅才接收该发布到Topic的消息；3）如果FilterTag数组有值，但MsgTag没设置，则不接收任何发布到Topic的消息，可以认为是2）的一种特例，此时FilterTag和MsgTag没有交集。规则整体的设计思想是以订阅者的意愿为主。 示例值： ["new"]
BindingKey	否	否	Array of String	BindingKey数量不超过5个， 每个BindingKey长度不超过64字节，该字段表示订阅接收消息的过滤策略，每个BindingKey最多含有15个“.”， 即最多16个词组。 示例值： ["ins"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

修改cmq主题属性

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改cmq主题属性

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 17:01:39。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyCmqTopicAttribute
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicName	是	否	String	主题名字，在单个地域同一帐号下唯一。主题名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-topic
MaxMsgSize	否	否	Uint64	消息最大长度。取值范围1024 - 65536 Byte（即1 - 64K），默认值65536。 示例值： 1
MsgRetentionSeconds	否	否	Uint64	消息保存时间。取值范围60 - 86400 s（即1分钟 - 1天），默认值86400。 示例值： 1
Trace	否	否	Bool	是否开启消息轨迹标识，true表示开启，false表示不开启，不填表示不开启。 示例值： false

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

回溯cmq队列

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

回溯cmq队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 16:59:19。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RewindCmqQueue
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueueName	是	否	String	队列名字，在单个地域同一帐号下唯一。队列名称是一个不超过64个字符的字符串，必须以字母为首字符，剩余部分可以包含字母、数字和横划线(-)。 示例值： test-queue
StartConsumeTime	是	否	Uint64	设定该时间，则（Batch）receiveMessage接口，会按照生产消息的先后顺序消费该时间戳以后的消息。 示例值： 1582108595000

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

解绑cmq死信队列

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

解绑cmq死信队列

默认接口请求频率限制：20次/秒。

接口更新时间：2021-09-15 16:57:31。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UnbindCmqDeadLetter
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
SourceQueueName	是	否	String	死信策略源队列名称，调用本接口会清空该队列的死信队列策略。 示例值： test-queue

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

RocketMQ 主题相关接口
创建RocketMQ主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ主题

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Topic	是	是	String	主题名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： order-topic
Namespaces	是	是	Array of String	主题所在的命名空间，目前支持在单个命名空间下创建主题 示例值： ["ns01"]
Type	是	是	String	主题类型，可选值为 Normal, GlobalOrder, PartitionedOrder, Transaction, DelayScheduled。 Transaction仅在专享版支持。 示例值： Normal
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
Remark	否	是	String	主题说明，最大128个字符 示例值： info-mark
PartitionNum	否	是	Int64	分区数，全局顺序无效 示例值： 1

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.CreateTopic	
ResourceInUse	
InvalidParameter	
InternalError	
LimitExceeded	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
ResourceNotFound.Namespace	
InternalError.IllegalMessage	
ResourceUnavailable.FundRequired	
InvalidParameter.Partition	

删除RocketMQ主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ主题

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间名称 示例值： test_namespace
Topic	是	是	String	主题名称 示例值： test_topic

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.DeleteTopics	
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	

获取Topic生产详情列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取Topic生产详情列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQTopicStats
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	实例ID 示例值： rocketmq-7j8m97bqpx5d
NamespaceId	是	是	String	命名空间 示例值： test_namespace
TopicName	是	是	String	主题名 示例值： test_topic

3. 输出参数

参数名称	类型	描述
TopicStatsList	Array of TopicStats	生产详情列表 示例值：查看

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取RocketMQ主题列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQTopics
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
DataOwnerUin	否	是	Int64	查询指定主账号uin的资源 示例值： 7621238456
TagFilters	否	是	Array of TagFilter	标签过滤器 示例值：查看
Limit	是	是	Uint64	查询限制数 示例值： 10
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间 示例值： test_ns
FilterName	否	是	String	按主题名称搜索，支持模糊查询 示例值： test_topic

参数名称	必选	允许NULL	类型	描述
FilterGroup	否	是	String	按订阅消费组名称过滤 示例值： test_group
DataRegion	否	是	String	查询指定region的资源 示例值： ap-beijing
Offset	是	是	Uint64	查询偏移量 示例值： 0
FilterType	否	是	Array of String	按主题类型过滤查询结果，可选择 Normal, GlobalOrder, PartitionedOrder, Transaction 示例值： ["Normal"]

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总记录数 示例值： 100
Topics	Array of RocketMQTopic	主题信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
ResourceNotFound.BrokerCluster	

错误码	描述
ResourceNotFound.Cluster	
InternalError.IllegalMessage	

更新RocketMQ主题信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ主题信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
PartitionNum	否	是	Int64	分区数，全局类型无效，不可小于当前分区数 示例值： 3
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间名称 示例值： test_namespace
Topic	是	是	String	主题名称 示例值： test_topic
Remark	是	是	String	说明信息，最大128个字符 示例值： 测试修改

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	
InvalidParameter.Partition	

RocketMQ 命名空间相关接口
创建RocketMQ命名空间

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQNamespace
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： test_ns
Ttl	否	是	Uint64	已废弃 示例值： 259200000
RetentionTime	否	是	Uint64	已废弃 示例值： 259200000
Remark	否	是	String	说明，最大128个字符 示例值： test remark info.

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceInsufficient	
LimitExceeded	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
FailedOperation.SetTTL	
InternalError.IllegalMessage	
ResourceUnavailable.FundRequired	
FailedOperation.InstanceNotReady	

删除RocketMQ命名空间

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQNamespace
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间名称 示例值： prod

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	
FailedOperation.RoleInUse	

获取RocketMQ命名空间列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQNamespaces
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	是	是	Uint64	偏移量 示例值： 0
Limit	是	是	Uint64	限制数目 示例值： 10
NameKeyword	否	是	String	按名称搜索 示例值： ordername
DataRegion	否	是	String	查询指定region的资源 示例值： ap-beijing
DataOwnerUin	否	是	Int64	查询指定主账号uin的资源 示例值： 7261234356
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq

3. 输出参数

参数名称	类型	描述
Namespaces	Array of RocketMQNamespace	命名空间列表 示例值：查看
TotalCount	Uint64	总条数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
InternalError.IllegalMessage	

更新RocketMQ命名空间

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQNamespace
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
PublicAccessEnabled	否	是	Bool	是否开启公网访问 示例值： false
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： prod
Ttl	否	是	Uint64	已废弃 示例值： 100000
RetentionTime	否	是	Uint64	已废弃 示例值： 100000

参数名称	必选	允许NULL	类型	描述
Remark	否	是	String	说明，最大128个字符 示例值： info-remark

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceInsufficient	
LimitExceeded	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
FailedOperation.SetTTL	
ResourceUnavailable.FundRequired	

RocketMQ 平滑迁移相关接口
平滑迁移：查询Topic迁移状态列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询Topic迁移状态列表，源集群和目标集群客户端数量信息需要配合DescribeRocketMQSmoothMigrationTaskTopicInsNum接口查询

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQMigratingTopicList
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TaskId	是	是	String	迁移任务名称 示例值： 4ed3a07b-36ef-4522-968b-5434dff663xi
Limit	是	是	Int64	分页大小 示例值： 10
Offset	是	是	Int64	偏移量 示例值： 10
Filters	否	是	Array of Filter	查询过滤器，支持topicname、MigrationStatus查询 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总条数 示例值： 20
MigrateTopics	Array of MigrateTopic	迁移topic列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取RocketMQ平滑迁移任务详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于获取RocketMQ平滑迁移任务详情

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQSmoothMigrationTask
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TaskId	是	是	String	任务ID 示例值： a50e97f6-8d2b-4fa8-b1f8-ba47bc50780e

3. 输出参数

参数名称	类型	描述
EnableACL	Bool	是否开启ACL 示例值： false
TopicStageDistribution	Array of RocketMQMigrationTopicDistribution	主题迁移进度分布情况 示例值：查看
TopicTypeDistribution	Array of RocketMQTopicDistribution	主题类型分布情况 示例值：查看
ClusterId	String	目标集群ID 示例值： rocketmq-a52qova7x97b

参数名称	类型	描述
AccessKey	String	源集群AccessKey 示例值： ZlbUZYhhsvYallgZ
TaskError	String	配置源集群时发生的错误 TIMEOUT 连接超时， SERVER_ERROR 服务错误， INTERNAL_ERROR 内部错误， CONNECT_NAMESERVER_ERROR 连接nameserver错误 CONNECT_BROKER_ERROR 连接broker错误 ACL_WRONG ACL信息不正确 示例值： CONNECT_NAMESERVER_ERROR
ConnectionType	String	网络连接类型， PUBLIC 公网 VPC 私有网络 OTHER 其它 示例值： PUBLIC
SecretKey	String	元集群SecretKey 示例值： ZlbUZYuiytYalksZ
TaskId	String	任务ID 示例值： 4ed3a07b-36ef-4522-968b-9845dff663xi
TaskName	String	任务名称 示例值： test_migration
SourceClusterName	String	源集群名称 示例值： test_cluster
VpcId	String	源集群所在私有网络ID 示例值： vpc-1bcglhk2
SubnetId	String	源集群所在子网ID 示例值： subnet-kxj97s89
TaskStatus	String	任务状态 Configuration 迁移配置 SourceConnecting 连接源集群中 SourceConnectionFailure 连接源集群失败 MetaDataImport 元数据导入 EndpointSetup 切换接入点 ServiceMigration 切流中 Completed 已完成 Cancelled 已取消

参数名称	类型	描述
		示例值： Cancelled
SourceClusterNameServer	String	源集群NameServer地址 示例值： http://rocketmq-a52qova7x97b.tdmq-rocketmq.public.tencenttdmq.com:6688
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.MigrationTask	

获取RocketMQ平滑迁移任务列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于查询RocketMQ平滑迁移任务列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQSmoothMigrationTaskList
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	是	是	Int64	查询起始偏移量 示例值： 0
Limit	是	是	Int64	查询最大数量 示例值： 10
Filters	否	是	Array of Filter	查询过滤器， 支持的字段如下 TaskStatus, 支持多选 ConnectionType，支持多选 ClusterId，精确搜索 TaskName，支持模糊搜索 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	任务总数 示例值： 1
Data	Array of RocketMQSmoothMigrationTaskItem	任务列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

平滑迁移：获取源集群的group列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

平滑迁移过程获取源集群group列表接口

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQSourceClusterGroupList
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Limit	是	是	Int64	页大小 示例值： 10
Offset	是	是	Int64	偏移量 示例值： 5
TaskId	是	是	String	迁移任务名称 示例值： a50e97f6-8d2b-4fa8-b1f8-ba47bc50780e
Filters	否	是	Array of Filter	查询过滤器，支持字段groupName，imported 示例值：查看

3. 输出参数

参数名称	类型	描述
Groups	Array of RocketMQGroupConfigOutput	group列表 示例值：查看
TotalCount	Int64	总条数 示例值： 12
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

平滑迁移：获取源集群的topic列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

平滑迁移过程获取源集群topic列表接口

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQSourceClusterTopicList
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Limit	是	是	Int64	分页大小 示例值： 0
Offset	是	是	Int64	偏移量 示例值： 10
TaskId	是	是	String	迁移任务名 示例值： a50e97f6-8d2b-4fa8-b1f8-ba47bc50780e
Filters	否	是	Array of Filter	查询过滤器，支持字段如下 TopicName, Type，Imported 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	总条数 示例值： 9
Topics	Array of RocketMQTopicConfigOutput	topic层列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

平滑迁移：导入消费者组列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

输入迁移任务id和要导入的Group，导入后台

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ImportRocketMQConsumerGroups
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Groups	是	是	Array of RocketMQGroupConfig	导入topic 示例值：查看
TaskId	是	是	String	任务id 示例值： 20157405-8691-4972-b7ec-52fd0839b7a9t

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

平滑迁移：导入topic列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

导入topic列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ImportRocketMQTopics
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Topics	是	是	Array of RocketMQTopicConfig	导入topic 示例值：查看
TaskId	是	是	String	任务ID 示例值： a50e97f6-8d2b-4fa8-b1f8-ba47bc50780e

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

RocketMQ 消息相关接口
rocketmq消息详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

rocketmq消息详情

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQMsg
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	是	String	命名空间 示例值： test_ns
MsgId	是	是	String	消息id 示例值：092BAE5A1656070DEA4E276DF0760089
QueryDlqMsg	否	是	Bool	查询死信时该值为true，只对Rocketmq有效 示例值： false
TopicName	是	是	String	主题，查询死信时传groupId 示例值： test_topic

参数名称	必选	允许NULL	类型	描述
PulsarMsgId	是	是	String	pulsar消息id 示例值： 092BAE5A1656070DEA4E276DF0760089
QueryDeadLetterMessage	否	是	Bool	查询死信时该值为true，只对Rocketmq有效 示例值： false
Offset	否	是	Int64	分页Offset 示例值： 0
Limit	否	是	Int64	分页Limit 示例值： 20
FilterTrackGroup	否	是	String	根据消费组名称过滤消费详情 示例值： test_group
ClusterId	是	是	String	集群id 示例值： rocketmq-4k4orqgq

3. 输出参数

参数名称	类型	描述
MessageTracksCount	Int64	消费组消费情况列表总数 示例值： 100
Body	String	消息体 示例值： order-id-2ksn3b23
Properties	String	详情参数 示例值： {"type":"normal"}
ProduceTime	String	生产时间 示例值： 2021-11-23 21:21:26
MsgId	String	消息id 示例值： 092BAE5A1656070DEA4E276DF0760089
ProducerAddr	String	生产者地址 示例值： 9.43.174.90
MessageTracks	Array of RocketMQMessageTrack	消费组消费情况列表 示例值：查看

参数名称	类型	描述
ShowTopicName	String	详情页展示的topic名称 示例值： test_topic
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	

查询RocketMQ消息轨迹

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询消息轨迹

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQMsgTrace
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群id 示例值： rocketmq-4k4orqgq
EnvironmentId	是	是	String	命名空间 示例值： test_ns
TopicName	是	是	String	主题，rocketmq查询死信时值为groupId 示例值： test_topic
MsgId	是	是	String	消息id 示例值：092BAE5A1656070DEA4E276DF0760089
GroupName	否	是	String	消费组、订阅 示例值： test_group

参数名称	必选	允许NULL	类型	描述
QueryDLQMsg	否	是	Bool	查询死信时该值为true 示例值： false
QueryDeadLetterMessage	否	是	String	查询死信时该值为true 示例值： false

3. 输出参数

参数名称	类型	描述
Result	Array of TraceResult	轨迹详情列表 示例值：查看
ShowTopicName	String	消息轨迹页展示的topic名称 示例值： test_topic
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	

rocketmq 消息查询

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

rocketmq 消息查询

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQTopicMsgs
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
QueryDlqMsg	否	是	Bool	死信查询时该值为true，只对Rocketmq有效 示例值： false
NumOfLatestMsg	否	是	Int64	查询最近N条消息 最大不超过1024，默认-1为其他查询条件 示例值： 20
Tag	否	是	String	TAG表达式 示例值： test_tag
QueryDeadLetterMessage	否	是	Bool	死信查询时该值为true，只对Rocketmq有效 示例值： false
ClusterId	是	是	String	集群 ID 示例值： rocketmq-4k4orqgq

参数名称	必选	允许NULL	类型	描述
EnvironmentId	是	是	String	命名空间 示例值： test_namespace
TopicName	是	是	String	主题名称，查询死信时为groupId 示例值： test_topic
StartTime	是	是	String	开始时间 示例值： 2021-12-22 12:22:22
EndTime	是	是	String	结束时间 示例值： 2021-12-22 12:22:22
Limit	否	是	Uint64	查询限额 示例值： 20
TaskRequestId	否	是	String	标志一次分页事务 示例值： 2e5ecbb3-f2d6-49d3-9c82-8648f8ed8748
MsgId	否	是	String	消息 ID 示例值：092BAE5A1656070DEA4E276DF0760089
MsgKey	否	是	String	消息 key 示例值： test_key
Offset	否	是	Uint64	查询偏移 示例值： 0

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 20
TopicMsgLogSets	Array of RocketMQMsgLog	消息列表 示例值：查看
TaskRequestId	String	标志一次分页事务 示例值： 2e5ecbb3-f2d6-49d3-9c82-8648f8ed8748
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

导出RocketMQ消息详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

导出RocketMQ消息详情

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ExportRocketMQMessageDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
DeadLetterMsg	否	是	Bool	是否死信消息 示例值： false
ClusterId	是	是	String	集群id 示例值： rocketmq-a5nzx52ab7jb
EnvironmentId	是	是	String	应用命名空间 示例值： test_ns
TopicName	是	是	String	Topic名称 如果是死信消息 isDlqMsg=true 示例值： test_topic
MsgId	是	是	String	消息id 示例值： 7F0000013AEC18B4AAC278B640D30002

参数名称	必选	允许NULL	类型	描述
IncludeMsgBody	是	是	Bool	是否包含消息体 示例值： false

3. 输出参数

参数名称	类型	描述
ReConsumeTimes	Uint64	消息重试次数 示例值： 0
MsgBody	String	Base64编码格式字符串 示例值： This is a normal message for Apache RocketMQ
MsgBodyCRC	Int64	消息内容的CRC32 Code 示例值： 0
MsgBodySize	Uint64	消息体大小（单位K） 当大于2048时不返回消息 示例值： 1024
MsgId	String	消息id 示例值： 7F0000013AEC18B4AAC278B640D30002
BornTimestamp	Int64	消息生成时间戳 示例值： 1729478721
Properties	String	消息属性 示例值： {"orderId":"1002328497239845"}
StoreTimestamp	Int64	消息存储时间戳 示例值： 1729478721
BornHost	String	消息生产客户端地址 示例值： 10.2.139.175:8080
MsgTag	String	消息Tag 示例值： test_tag
MsgKey	String	消息Key 示例值： test_key
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	
InternalError.FailedOperation	

重置RocketMQ消费位点

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

重置指定Group的消费位点到指定时间戳

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ResetRocketMQConsumerOffSet
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RetryFlag	否	是	Bool	重置的是否是retry topic 示例值： false
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间名称 示例值： test_namespace
GroupId	是	是	String	消费组名称 示例值： test_group
Type	是	是	Uint64	重置方式，0表示从最新位点开始，1表示从指定时间点开始 示例值： 1

参数名称	必选	允许NULL	类型	描述
Topic	否	是	String	主题名称 示例值： test_topic
ResetTimestamp	否	是	Uint64	重置指定的时间戳，仅在 Type 为1是生效，以毫秒为单位 示例值： 1697945990000

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	

重发RocketMQ死信消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

重发RocketMQ死信消息

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RetryRocketMQDlqMessage
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群id 示例值： rocketmq-a52qova7x97b
NamespaceId	是	是	String	命名空间名称 示例值： test_ns
GroupName	是	是	String	group名称 示例值： test_group
MessageIds	是	是	Array of String	死信消息ID 示例值： ["01963F0B14BAF09F27077D244F00000000"]

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	

发送RocketMQ消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送 RocketMQ 消息，该接口仅用于控制台发送少量测试消息，不保证SLA，且云 API 存在限流，在真实业务场景下，请使用 RocketMQ SDK 发送消息。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SendRocketMQMessage
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
MsgKey	否	是	String	消息key信息 示例值： test_key
MsgTag	否	是	String	消息tag信息 示例值： test_tag
ClusterId	是	是	String	集群id 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间 示例值： test_namespace
TopicName	是	是	String	topic名称 示例值： test_topic
MsgBody	是	是	String	信息内容 示例值：

参数名称	必选	允许NULL	类型	描述
				This is a normal message for Apache RocketMQ

3. 输出参数

参数名称	类型	描述
Result	Bool	发送结果 示例值： true
MsgId	String	消息ID 示例值： 7F0000013AEC18B4AAC278B640D30002
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.Topic	
FailedOperation	
ResourceNotFound.Cluster	
ResourceUnavailable.FundRequired	

Rocketmq消费验证

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

Rocketmq消费验证

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：VerifyRocketMQConsume
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClientId	是	是	String	客户端ID 示例值： 127.0.0.1@25760#13749780392400
TopicName	是	是	String	主题名称 示例值： test_topic
ClusterId	是	是	String	集群id 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间 示例值： test_namespace
GroupId	是	是	String	消费组ID 示例值： test_group
MsgId	是	是	String	消息id 示例值： 092BAE5A1656070DEA4E276DF0760089

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	

RocketMQ 消费组相关接口
创建RocketMQ消费组

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ消费组

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQGroup
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RetryMaxTimes	否	是	Uint64	Group最大重试次数 示例值： 16
GroupId	是	是	String	Group名称，8~64个字符 示例值： test_group
Namespaces	是	是	Array of String	命名空间，目前只支持单个命名空间 示例值： ["test_namespace"]
ReadEnable	是	是	Bool	是否开启消费 示例值： true
BroadcastEnable	是	是	Bool	是否开启广播消费 示例值： true

参数名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
Remark	否	是	String	说明信息，最长128个字符 示例值： 测试创建
GroupType	否	是	String	Group类型（TCP/HTTP） 示例值： TCP

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
LimitExceeded	
FailedOperation	
ResourceNotFound.Cluster	
InternalError.IllegalMessage	

删除RocketMQ消费组

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ消费组

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQGroup
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间名称 示例值： test_namespace
GroupId	是	是	String	消费组名称 示例值： test_group

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
InternalError.IllegalMessage	

获取消费详情列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取消费详情列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQConsumeStats
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	实例ID 示例值： rocketmq-7j8m97bqpx5d
NamespaceId	是	是	String	命名空间 示例值： test_namespace
ConsumerGroup	是	是	String	消费组 示例值： test_group

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
ConsumerStatsList	Array of ConsumerStats	消费详情列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取在线消费端详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取在线消费端详情

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQConsumerConnectionDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
NamespaceId	是	是	String	命名空间名称 示例值： test_ns
GroupId	是	是	String	消费组名称 示例值： test_group
ClientId	是	是	String	消费端实例ID 示例值： local@23645@0@30ey5aef8w
Offset	是	是	Uint64	偏移量 示例值： 0
Limit	是	是	Uint64	限制数目 示例值： 20
FilterType	否	是	Array of String	按主题类型过滤查询结果，可选择 Normal, GlobalOrder, PartitionedOrder, Retry, Transaction, DeadLetter 示例值： ["Normal"]
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总条数 示例值： 100
Details	Array of RocketMQConsumerTopic	消费端主题信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	

获取指定消费组下当前客户端的连接情况

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定消费组下当前客户端的连接情况

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQConsumerConnections
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
SortOrder	否	是	String	查询结果排序规则，ASC为升序，DESC为降序 示例值： ASC
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间名称 示例值： test_ns
GroupId	是	是	String	消费组ID 示例值： test_group
Offset	是	是	Uint64	偏移量 示例值： 0
Limit	是	是	Uint64	限制数目 示例值： 20

参数名称	必选	允许NULL	类型	描述
SortedBy	否	是	String	对查询结果排序，此为排序字段，目前支持Accumulative（消息堆积量） 示例值： Accumulative

3. 输出参数

参数名称	类型	描述
GroupDetail	RocketMQGroup	订阅组信息 示例值：查看
TotalCount	Uint64	总数目 示例值： 100
Connections	Array of RocketMQConsumerConnection	在线消费者信息 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	

获取RocketMQ消费组列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ消费组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQGroups
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TagFilters	否	是	Array of TagFilter	标签过滤器 示例值：查看
NamespaceId	是	是	String	命名空间 示例值： test_ns
Offset	是	是	Uint64	偏移量 示例值： 0
Limit	是	是	Uint64	限制条数 示例值： 20
FilterGroup	否	是	String	按消费组名称查询消费组，支持模糊查询 示例值： test_group
FilterOneGroup	否	是	String	订阅组名称，指定此参数后将只返回该订阅组信息 示例值： test_group

参数名称	必选	允许NULL	类型	描述
Types	否	是	Array of String	group类型 示例值： ["TCP"]
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
FilterTopic	否	是	String	主题名称，输入此参数可查询该主题下所有的订阅组 示例值： test_topic
SortedBy	否	是	String	按照指定字段排序，可选值为 subscribeNum: 订阅 Topic 个数 示例值： subscribeNum
SortOrder	否	是	String	按升序或降序排列，可选值为asc，desc 示例值： asc

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数量 示例值： 100
Groups	Array of RocketMQGroup	订阅组列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	

错误码	描述
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	

获取RocketMQ消费组订阅关系

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于获取RocketMQ消费组订阅关系数据

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQSubscriptions
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Group	是	是	String	消费组名称 示例值： test_group
Offset	是	是	Int64	查询起始位置 示例值： 0
Limit	是	是	Int64	查询限制条数 示例值： 20
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
Namespace	是	是	String	命名空间名称 示例值： test_namespace

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	总条数 示例值： 10
Subscriptions	Array of RocketMQSubscription	订阅关系列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceUnavailable	
FailedOperation	
ResourceNotFound.Cluster	

获取指定消费组下订阅的主题列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定消费组下订阅的主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQTopicsByGroup
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq
NamespaceId	是	是	String	命名空间名称 示例值： test_ns
GroupId	是	是	String	消费组名称 示例值： test_group
Offset	否	是	Uint64	偏移量 示例值： 0
Limit	否	是	Uint64	限制条数 示例值： 10

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总条数 示例值： 100
Topics	Array of String	主题列表 示例值： ["test_topic"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	

更新RocketMQ消费组信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ消费组信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQGroup
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
BroadcastEnable	否	是	Bool	是否开启广播消费 示例值： true
RetryMaxTimes	否	是	Uint64	最大重试次数 示例值： 16
ClusterId	是	是	String	集群ID 示例值： rocketmq-2p9vx3ax9jxg
NamespaceId	是	是	String	命名空间 示例值： test_namespace
GroupId	是	是	String	消费组名称 示例值： test_group
Remark	否	是	String	说明信息，最长128个字符 示例值： 测试消费组

参数名称	必选	允许NULL	类型	描述
ReadEnable	否	是	Bool	是否开启消费 示例值： true

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	

RocketMQ 角色授权相关接口
创建RocketMQ环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建环境角色授权

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQEnvironmentRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	是	是	String	角色名称。 示例值： test_role_name
Permissions	是	是	Array of String	授权项，最多只能包含produce、consume两项的非空字符串数组。 示例值： ["produce"]
ClusterId	是	是	String	必填字段，集群的ID 示例值： rocketmq-2p9vx3ax9jxg
DetailedPerms	否	是	Array of DetailedRolePerm	Topic&Group维度权限配置 示例值：查看

参数名称	必选	允许NULL	类型	描述
EnvironmentId	是	是	String	命名空间 示例值： test_namespace

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： test_namespace
RoleName	String	角色名称。 示例值： test_role_name
Permissions	Array of String	授权项。 示例值： ["produce"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
ResourceInUse.EnvironmentRole	
FailedOperation.CreateEnvironmentRole	
FailedOperation.UpdateEnvironmentRole	
ResourceInUse	

错误码	描述
FailedOperation	
ResourceNotFound.Cluster	

创建RocketMQ角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建角色

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	是	是	String	角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。 示例值： test_role_name
ClusterId	是	是	String	必填字段，集群Id 示例值： rocketmq-2p9vx3ax9jxg
Remark	否	是	String	备注说明，长度必须大等于0且小等于128。 示例值： 测试角色
PermType	否	是	String	角色授权类型（集群：Cluster; 主题或消费组：TopicAndGroup） 示例值： Cluster

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名称 示例值： test_role_name
Token	String	角色token 示例值： eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY
Remark	String	备注说明 示例值： 测试角色

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceInUse.Role	
FailedOperation.CreateRole	
FailedOperation	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	
FailedOperation.CreateSecretKey	
FailedOperation.SaveSecretKey	
ResourceUnavailable.FundRequired	

批量删除RocketMQ环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除环境角色授权。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQEnvironmentRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	是	String	环境（命名空间）名称。 示例值： test_namespace
RoleNames	是	是	Array of String	角色名称数组。 示例值： ["test_role_name"]
ClusterId	是	是	String	集群的ID 示例值： rocketmq-mq-4k4orqgq

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： test_namespace
RoleNames	Array of String	角色名称 示例值： ["test_role_name"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.EnvironmentRole	
FailedOperation.DeleteEnvironmentRoles	
FailedOperation	
ResourceNotFound.Cluster	

批量删除RocketMQ角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除角色，支持批量。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleNames	是	是	Array of String	角色名称数组。 示例值： ["test_role_name"]
ClusterId	是	是	String	必填字段，集群Id 示例值： rocketmq-2p9vx3ax9jxg

3. 输出参数

参数名称	类型	描述
RoleNames	Array of String	成功删除的角色名称数组。 示例值： ["test_role_name"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
FailedOperation.DeleteRoles	
ResourceInUse.EnvironmentRole	
FailedOperation	
ResourceNotFound.Cluster	

获取RocketMQ命名空间角色列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取命名空间角色列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQEnvironmentRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Limit	否	是	Int64	返回数量，不填则默认为10，最大值为20。 示例值： 20
RoleName	否	是	String	角色名称 示例值： test_role
Filters	否	是	Array of Filter	RoleName按照角色名进行过滤，精确查询。类型：String必选：否 示例值：查看
ClusterId	是	是	String	必填字段，RocketMQ集群的ID 示例值： rocketmq-ge3n43v3wj
EnvironmentId	否	是	String	命名空间 示例值： test_ns

参数名称	必选	允许NULL	类型	描述
Offset	否	是	Int64	起始下标，不填默认为0。 示例值： 0

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	记录数。 示例值： 10
EnvironmentRoleSets	Array of EnvironmentRole	命名空间角色集合。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

获取RocketMQ角色列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取角色列表

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Filters	否	是	Array of Filter	RoleName按照角色名进行过滤，精确查询。类型：String必选：否 示例值：查看
Offset	是	是	Int64	起始下标，不填默认为0。 示例值： 0
Limit	是	是	Int64	返回数量，不填则默认为10，最大值为20。 示例值： 20
ClusterId	是	是	String	必填字段，集群Id 示例值： rocketmq-2p9vx3ax9jxg
RoleName	否	是	String	角色名称，模糊查询 示例值： test_role_name

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	记录数。 示例值： 4
RoleSets	Array of Role	角色数组。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

修改RocketMQ环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改环境角色授权。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQEnvironmentRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
DetailedPerms	否	是	Array of DetailedRolePerm	Topic&Group维度权限配置 示例值：查看
EnvironmentId	是	是	String	环境（命名空间）名称。 示例值： test_namespace
RoleName	是	是	String	角色名称。 示例值： test_role_name
Permissions	是	是	Array of String	授权项，最多只能包含produce、consume两项的非空字符串数组。 示例值： ["produce"]
ClusterId	是	是	String	必填字段，集群的ID 示例值： rocketmq-2p9vx3ax9jxg

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： test_namespace
RoleName	String	角色名称。 示例值： test_role_name
Permissions	Array of String	授权项。 示例值： ["produce","consume"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.EnvironmentRole	
FailedOperation.UpdateEnvironmentRole	
FailedOperation	
ResourceNotFound.Cluster	

RocketMQ角色修改

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

角色修改

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	是	是	String	角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。 示例值： test_role_name
ClusterId	是	是	String	必填字段，集群Id 示例值： rocketmq-2p9vx3ax9jxg
Remark	否	是	String	备注说明，长度必须大等于0且小等于128。 示例值： 测试修改
PermType	否	是	String	权限类型，默认按集群授权（Cluster：集群级别；TopicAndGroup：主题&消费组级别） 示例值： Cluster

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名称 示例值： test_role_name
Remark	String	备注说明 示例值： 测试修改
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
FailedOperation.UpdateRole	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	
ResourceUnavailable.FundRequired	

RocketMQ 集群相关接口
创建RocketMQ集群

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

此接口用于创建一个RocketMQ集群

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Name	是	是	String	集群名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： test_name
Remark	否	是	String	集群描述，128个字符以内 示例值： remark info
TagList	否	是	Array of Tag	标签列表 示例值：查看

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
ClusterId	String	集群ID 示例值： rocketmq-a52qova7x97b
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceInsufficient	
LimitExceeded	
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
LimitExceeded.Clusters	
FailedOperation.CreateCluster	
ResourceUnavailable.FundRequired	
ResourceUnavailable.CreateFailed	

创建RocketMQ专享实例

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建RocketMQ专享实例

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRocketMQVipInstance
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TimeSpan	是	是	Int64	购买时长，月为单位 示例值： 12
SupportsMigrateToCloud	否	是	Bool	是否用于迁移上云，默认为false 示例值： false
Bandwidth	否	是	Int64	公网带宽，在开启公网情况下为必传字段 示例值： 0
Spec	是	是	String	集群规格，支持规格有 1.通用型:rocket-vip-basic-0; 2.基础型:rocket-vip-basic-1; 3.标准型:rocket-vip-basic-2; 4.高阶Ⅰ型:rocket-vip-basic-3; 5.高阶Ⅱ

参数名称	必选	允许NULL	类型	描述
				型:rocket-vip-basic-4 示例值： rocket-vip-basic-1
VpcInfo	是	是	VpcInfo	VPC信息 示例值：查看
EnablePublic	否	是	Bool	是否开启公网 示例值： false
IpRules	否	是	Array of PublicAccessRule	公网白名单 示例值：查看
Tags	否	是	Array of Tag	标签 示例值：查看
Name	是	是	String	实例名称 示例值： mq-instance
NodeCount	是	是	Int64	节点数量，最小2，最大20 示例值： 6
StorageSize	是	是	Int64	单节点存储空间，GB为单位，最低200GB 示例值： 500
ZoneIds	是	是	Array of String	节点部署的区域ID列表，如广州一区，则是100001，具体可查询腾讯云官网 示例值： ["100001"]

3. 输出参数

参数名称	类型	描述
ClusterId	String	集群ID 示例值： rocketmq-7drjznvjqzee
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation.CallTrade	

删除RocketMQ集群

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ集群

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	待删除的集群Id。 示例值： rocketmq-xxxxx

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	
FailedOperation.DeleteCluster	
ResourceNotFound.Cluster	

删除RocketMQ专享实例

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除RocketMQ专享实例

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRocketMQVipInstance
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	实例的集群ID 示例值： rocketmq-ajknm838qzg4

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceNotFound.Instance	

获取单个RocketMQ集群信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取单个RocketMQ集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-rd3545bkkj49

3. 输出参数

参数名称	类型	描述
ClusterInfo	RocketMQClusterInfo	集群信息 示例值：查看
ClusterConfig	RocketMQClusterConfig	集群配置 示例值：查看

参数名称	类型	描述
ClusterStats	RocketMQClusterRecentStats	集群最近使用量，即将废弃，请使用腾讯云可观测平台获取相关数据。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
InternalError.IllegalMessage	

获取RocketMQ的集群列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取RocketMQ集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	是	是	Uint64	偏移量 示例值： 0
Limit	是	是	Uint64	限制数目 示例值： 10
IdKeyword	否	是	String	按照集群ID关键字搜索 示例值： rocketmq-a52qova7x97b
NameKeyword	否	是	String	按照集群名称关键字搜索 示例值： test_name
ClusterIdList	否	是	Array of String	集群ID列表过滤 示例值： ["rocketmq-a52qova7x97b"]
IsTagFilter	否	是	Bool	标签过滤查找时，需要设置为true 示例值： true

参数名称	必选	允许NULL	类型	描述
Filters	否	是	Array of Filter	过滤器。目前支持标签过滤。 示例值：查看

3. 输出参数

参数名称	类型	描述
ClusterList	Array of RocketMQClusterDetail	集群信息 示例值：查看
TotalCount	Uint64	总条数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
InternalError	
ResourceNotFound	
FailedOperation	

拉取公网指标监控数据

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

从腾讯云可观测平台拉取公网指标监控数据，目前仅支持客户端到 LB 的入带宽和出宽带指标。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQPublicAccessMonitorData
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
InstanceId	是	是	String	专享集群ID 示例值： rocketmq-xxx
MetricName	是	是	String	指标名称，仅支持单指标拉取。目前仅支持：ClientIntraffic; ClientOuttraffic 示例值： ClientOuttraffic
StartTime	否	是	String	起始时间 示例值： 2018-09-22T19:51:23+08:00
EndTime	否	是	String	结束时间，默认为当前时间 示例值： 2018-09-22T20:51:23+08:00
Period	否	是	Int64	监控统计周期，如60。默认为取值为300，单位为s。 示例值： 60

3. 输出参数

参数名称	类型	描述
MetricName	String	指标名 示例值： ClientOuttraffic
Period	Int64	监控统计周期，如60。默认为取值为300，单位为s。 示例值： 60
StartTime	String	起始时间，如2018-09-22T19:51:23+08:00 示例值： 2018-09-22T19:51:23+08:00
EndTime	String	结束时间，如2018-09-22T20:51:23+08:00，默认为当前时间 示例值： 2018-09-22T20:51:23+08:00
DataPoints	Array of RocketMQDataPoint	数据点数组 示例值：查看
Msg	String	返回信息 示例值： succ
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询RocketMQ实例公网接入点信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

接口用于查询RocketMQ实例的公网接入信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQPublicAccessPoint
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
InstanceId	是	是	String	集群ID，当前只支持专享集群 示例值： rocketmq-a52qova7x97b

3. 输出参数

参数名称	类型	描述
Status	Int64	公网接入点状态： 0， 已开启 1， 已关闭 2，开启中 3，关闭中 4，修改中 示例值： 0

参数名称	类型	描述
PayStatus	Int64	支付状态： 0, 未知 1，正常 2，欠费 示例值： 1
AccessUrl	String	接入点地址 示例值： tdmq-rocketmq.ap-gz.public.tencenttdmq.com:9876
Rules	Array of PublicAccessRule	安全访问规则列表 示例值：查看
Bandwidth	Int64	带宽 示例值： 2
PayMode	Int64	付费模式 示例值： 0
BillingFlow	Bool	公网是否按流量计费 示例值： false
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取 RocketMQ 集群相关指标排序列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于获取RocketMQ指标排序列表，比如集群实例下占用存储空间最多的主题排序。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQTopUsages
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-vvqb9emabapx
MetricName	是	是	String	指标名称，支持以下： consumeLag，消费组堆积数量 deadLetterCount，死信数量 topicRateIn， Topic生产速率 topicRateOut，Topic消费速率 topicStorageSize，Topic存储空间 topicApiCalls，Topic API调用次数 示例值： consumeLag
Limit	是	是	Uint64	排序数量，最大20 示例值： 10

3. 输出参数

参数名称	类型	描述
Values	Array of Int64	指标值列表 示例值： [120,88]
Dimensions	Array of DimensionInstance	指标值对应的维度组合，本接口存在以下几个维度：tenant，namespace，group，topic 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceNotFound.Cluster	

获取单个RocketMQ专享集群信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取单个RocketMQ专享集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQVipInstanceDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	集群ID 示例值： rocketmq-4k4orqgq

3. 输出参数

参数名称	类型	描述
ClusterInfo	RocketMQClusterInfo	集群信息 示例值：查看
InstanceConfig	RocketMQInstanceConfig	集群配置 示例值：查看

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter	
ResourceUnavailable	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
InternalError.IllegalMessage	
ResourceNotFound.Instance	

查询RocketMQ专享实例列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询用户已购的RocketMQ专享实例列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRocketMQVipInstances
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Filters	否	是	Array of Filter	查询条件过滤器，支持的查询条件如下： instanceIds - 实例ID instanceName - 实例名称 status - 实例状态 示例值：查看
Limit	否	是	Uint64	查询数目上限，默认20 示例值： 20
Offset	否	是	Uint64	查询起始位置 示例值： 1

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
TotalCount	Uint64	未分页的总数目 示例值： 10
Instances	Array of RocketMQVipInstance	实例信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

更新RocketMQ集群信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新RocketMQ集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	是	String	RocketMQ集群ID 示例值： rocketmq-a52qova7x97b
ClusterName	否	是	String	3-64个字符，只能包含字母、数字、“-”及“_” 示例值： test_name
Remark	否	是	String	说明信息，不超过128个字符 示例值： remark info
PublicAccessEnabled	否	是	Bool	是否开启HTTP公网访问 示例值： false

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
InvalidParameter	
InternalError	
ResourceInsufficient	
LimitExceeded	
ResourceNotFound	
FailedOperation	
ResourceNotFound.Cluster	
ResourceUnavailable.FundRequired	

修改RocketMQ专享实例

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改RocketMQ专享实例

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQInstance
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
InstanceId	是	是	String	专享实例ID 示例值： rocketmq-4k4orqgq
Name	否	是	String	实例名称 示例值： order-service-instance
Remark	否	是	String	实例备注信息 示例值： info-mark
MessageRetention	否	是	Int64	实例消息保留时间，小时为单位 示例值： 72
EnableDeletionProtection	否	是	Bool	是否开启删除保护 示例值： false

参数名称	必选	允许NULL	类型	描述
SendReceiveRatio	否	是	Float	控制生产和消费消息的 TPS 占比，取值范围0～1，默认值为0.5 示例值： 0.6

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	

修改RocketMQ专享实例配置

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

本API用于修改RocketMQ专享实例配置，可以支持实例规格、节点数和存储的升配和实例规格的降配。本API发起订单并成功支付后进入实例配置变更的流程，可通过DescribeRocketMQVipInstances查询实例是否已变更完成。

默认接口请求频率限制：10000次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRocketMQInstanceSpec
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
InstanceId	是	是	String	专享实例ID 示例值： rocketmq-4k4orqgq
Specification	否	是	String	实例规格， rocket-vip-basic-1 基础型 rocket-vip-basic-2 标准型 rocket-vip-basic-3 高阶Ⅰ型 rocket-vip-basic-4 高阶Ⅱ型 示例值： rocket-vip-basic-1
NodeCount	否	是	Uint64	节点数量 示例值： 2
StorageSize	否	是	Uint64	存储空间，GB为单位 示例值： 1000

3. 输出参数

参数名称	类型	描述
OrderId	String	订单号 示例值： 2023072007301760809412 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.AtLeastOne	
ResourceNotFound.Instance	
FailedOperation.InstanceNotReady	
UnsupportedOperation.InstanceDowngrade	
FailedOperation.CallTrade	

设置RocketMQ实例公网访问接入点

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

该接口用于开启关闭公网访问、设置安全访问策略

默认接口请求频率限制：20次/秒。

接口更新时间：2025-11-18 22:21:43。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SetRocketMQPublicAccessPoint
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Enabled	是	是	Bool	开启或关闭访问 示例值： true
Bandwidth	否	是	Uint64	带宽大小，开启或者调整公网时必须指定，Mbps为单位 示例值： 2
PayMode	否	是	Uint64	付费模式，开启公网时必须指定，0为按小时计费，1为包年包月，当前只支持按小时计费 示例值： 0
Rules	否	是	Array of PublicAccessRule	公网访问安全规则列表，Enabled为true时必须传入 示例值：查看
BillingFlow	否	是	Bool	公网是否按流量计费 示例值： false

参数名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	集群ID，当前只支持专享集群 示例值： rocketmq-a52qova7x97b

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceInUse	
ResourceNotFound	
FailedOperation	

主题相关接口
新增主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

新增指定分区、类型的消息主题

默认接口请求频率限制：200次/秒。

接口更新时间：2021-07-19 19:21:10。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
TopicName	是	否	String	主题名，不支持中字以及除了短线和下划线外的特殊字符且不超过64个字符。 示例值： test_topic
Partitions	是	否	Uint64	0：非分区topic，无分区；非0：具体分区topic的分区数，最大不允许超过128。 示例值： 2
Remark	否	否	String	备注，128字符以内。 示例值： 2个分区的普通消息队列
TopicType	是	否	Uint64	0： 普通消息； 1 ：全局顺序消息；

参数名称	必选	允许NULL	类型	描述
				2 ：局部顺序消息； 3 ：重试队列； 4 ：死信队列。 示例值： TopicType 和 PulsarTopicType 两者不能同时为空，有一个必选
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-5r59xd4vnx

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： default
TopicName	String	主题名。 示例值： test_topic
Partitions	Uint64	0：非分区topic，无分区；非0：具体分区topic的分区数。 示例值： 2
Remark	String	备注，128字符以内。 示例值： 2个分区的普通消息队列
TopicType	Uint64	0： 普通消息； 1 ：全局顺序消息； 2 ：局部顺序消息； 3 ：重试队列； 4 ：死信队列； 5 ：事务消息。 示例值： 0
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	

错误码	描述
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
ResourceInUse.Topic	重名，主题已存在。
FailedOperation.CreateTopic	
MissingParameter.NeedMoreParams	
LimitExceeded.Topics	实例下主题数量超过限制。
InvalidParameterValue	参数取值错误。
FailedOperation	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

删除主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量删除topics

默认接口请求频率限制：20次/秒。

接口更新时间：2021-08-04 10:41:16。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteTopics
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TopicSets	是	否	Array of TopicRecord	主题集合，每次最多删除20个。 示例值：查看
ClusterId	否	否	String	pulsar集群Id。 示例值： pulsar-xk3ne8k2qkp8
EnvironmentId	否	否	String	环境（命名空间）名称。 示例值： devNs
Force	否	否	Bool	是否强制删除，默认为false 示例值： false

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
TopicSets	Array of TopicRecord	被删除的主题数组。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
ResourceNotFound.Topic	
FailedOperation.DeleteTopics	
OperationDenied.DefaultEnvironment	默认环境不允许操作。
MissingParameter.NeedMoreParams	
FailedOperation	
ResourceNotFound.Cluster	

查询跨地域复制目标集群

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询跨地域复制目标集群

默认接口请求频率限制：20次/秒。

接口更新时间：2023-03-16 11:54:56。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeReplicationClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
TargetRegionId	是	否	Uint64	目标集群所在地域Id 示例值： gz

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	目标集群总数 示例值： 10
ClusterSets	Array of ReplicationClusterSets	目标集群列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询topic跨地域复制绑定关系

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询topic跨地域复制绑定关系

默认接口请求频率限制：20次/秒。

接口更新时间：2023-03-16 11:57:01。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeTopicReplicationClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	否	String	Pulsar 集群的ID 示例值： pulsar-4k32ozmb4z
EnvironmentId	是	否	String	环境（命名空间）名称 示例值： default
TopicName	是	否	String	主题名 示例值： global4

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
TotalCount	Uint64	绑定的目标集群总数 示例值： 10
ClusterSets	Array of ReplicationClusterSets	绑定的目标集群列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询主题列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取环境下主题列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 15:22:05。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeTopics
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
TopicName	否	否	String	主题名模糊匹配。 示例值： topic1
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 20
TopicType	否	否	Uint64	topic类型描述： 0：普通消息； 1：全局顺序消息； 2：局部顺序消息； 3：重试队列； 4：死信队列；

参数名称	必选	允许NULL	类型	描述
				5：事务消息。 示例值： 1
ClusterId	否	否	String	Pulsar 集群的ID 示例值： cluster1
Filters	否	否	Array of Filter	* TopicName 按照主题名字查询，精确查询。 类型：String 必选：否 示例值：查看
InternalReqSource	否	否	String	内部业务标识 示例值： rocketmq-xxxxx

3. 输出参数

参数名称	类型	描述
TopicSets	Array of Topic	主题集合数组。 示例值：查看
TotalCount	Uint64	主题数量。 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
MissingParameter.NeedMoreParams	

错误码	描述
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

修改主题

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改主题备注和分区数

默认接口请求频率限制：50次/秒。

接口更新时间：2021-06-16 18:02:21。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyTopic
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
TopicName	是	否	String	主题名。 示例值： test_topic
Partitions	是	否	Uint64	分区数，必须大于或者等于原分区数，若想维持原分区数请输入原数目，修改分区数仅对非全局顺序消息起效果，不允许超过128个分区。 示例值： 3
Remark	否	否	String	备注，128字符以内。 示例值： 修改分区为3
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-5r59xd4vnx

3. 输出参数

参数名称	类型	描述
Partitions	Uint64	分区数 示例值： 3
Remark	String	备注，128字符以内。 示例值： 修改分区为3
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Topic	
FailedOperation.UpdateTopic	主题更新失败。
MissingParameter.NeedMoreParams	
FailedOperation.GetTopicPartitionsFailed	获取主题分区数失败。
FailedOperation	

其他接口
删除角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除角色，支持批量。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:07:08。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleNames	是	否	Array of String	角色名称数组。 示例值： ["test_role_1","test_role_2"]
ClusterId	否	否	String	必填字段，集群Id 示例值： pulsar-xk3ne8k2qkp8

3. 输出参数

参数名称	类型	描述
RoleNames	Array of String	成功删除的角色名称数组。 示例值： ["test_role_1","test_role_2"]

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
FailedOperation.DeleteRoles	
ResourceInUse.EnvironmentRole	
FailedOperation	

获取租户VPC绑定关系

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取租户VPC绑定关系

默认接口请求频率限制：20次/秒。

接口更新时间：2021-01-13 15:38:57。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeBindVpcs
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 1
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 1
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-xk3ne8k2qkp8

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	记录数。 示例值： 1

参数名称	类型	描述
VpcSets	Array of VpcBindRecord	Vpc集合。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.NeedMoreParams	必要参数没有传递。
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.BrokerCluster	

运营端获取命名空间bundle列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

运营端获取命名空间bundle列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-21 11:41:44。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeNamespaceBundlesOpt
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterName	是	否	String	物理集群名 示例值： cluster-s3hklt5n
TenantId	是	否	String	虚拟集群（租户）ID 示例值： pulsar-nzw8q5b3g7b2
NamespaceName	是	否	String	命名空间名 示例值： default
NeedMetrics	是	否	Bool	是否需要监控指标，若传false，则不需要传Limit和Offset分页参数 示例值： true
Limit	否	否	Int64	查询限制条数 示例值： 10

参数名称	必选	允许NULL	类型	描述
Offset	否	否	Int64	查询偏移量 示例值： 0

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	记录条数 示例值： 10
BundleSet	Array of BundleSetOpt	bundle列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

运营端获节点健康状态

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

运营端获节点健康状态

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-21 11:41:51。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeNodeHealthOpt
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
InstanceId	是	否	String	节点实例ID 示例值： ins-pkqqy3e2

3. 输出参数

参数名称	类型	描述
NodeState	Int64	0-异常；1-正常 示例值： 1
LatestHealthCheckTime	String	最近一次健康检查的时间 示例值： 2021-07-06 00:00:00
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

发送消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

此接口仅用于测试发生消息，不能作为现网正式生产使用

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-21 17:11:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SendMsg
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： userNamespace
TopicName	是	否	String	主题名称，如果是分区topic需要指定具体分区，如果没有指定则默认发到0分区，例如：my_topic-partition-0。 示例值： cloud_test
MsgContent	是	否	String	消息内容，不能为空且大小不得大于5242880个byte。 示例值： one_msg
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-k3ex2kxanuol

3. 输出参数

参数名称	类型	描述
Result	Bool	结果。 示例值： true
MsgId	String	消息ID。 示例值： 50169152:0:0:-1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Topic	
MissingParameter.NeedMoreParams	
FailedOperation.GetTopicPartitionsFailed	获取主题分区数失败。
FailedOperation.SendMsgFailed	发送消息失败。
FailedOperation	

命名空间相关接口
创建命名空间

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

用于在用户账户下创建消息队列 Tdmq 命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-27 16:20:00。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateEnvironment
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称，不支持中字以及除了短线和下划线外的特殊字符且不超过16个字符。 示例值： devNamespace
MsgTTL	是	否	Uint64	未消费消息过期时间，单位：秒，最小60，最大1296000，（15天）。 示例值： 60
Remark	否	否	String	说明，128个字符以内。 示例值： demo
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj

参数名称	必选	允许NULL	类型	描述
RetentionPolicy	否	否	RetentionPolicy	消息保留策略 示例值：查看

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： devNamespace
MsgTTL	Uint64	未消费消息过期时间，单位：秒。 示例值： 60
Remark	String	说明，128个字符以内。 示例值： demo
NamespaceId	String	命名空间ID 示例值： namespace-ak7jexmrz2
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
FailedOperation.CreateEnvironment	环境创建失败。
OperationDenied.DefaultEnvironment	默认环境不允许操作。
MissingParameter.NeedMoreParams	
LimitExceeded.Environments	实例下环境数量超过限制。
FailedOperation	

错误码	描述
ResourceNotFound.Cluster	
InvalidParameterValue.TTL	无效的消息TTL值。
LimitExceeded.Namespaces	实例下命名空间数量超过限额。
ResourceInUse.Namespace	重名，命名空间已存在。
FailedOperation.CreateNamespace	创建命名空间失败。

删除命名空间

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量删除租户下的命名空间

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 18:37:26。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteEnvironments
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentIds	是	否	Array of String	环境（命名空间）数组，每次最多删除20个。 示例值： [test1,test2]
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj

3. 输出参数

参数名称	类型	描述
EnvironmentIds	Array of String	成功删除的环境（命名空间）数组。 示例值： [test1,test2]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
FailedOperation.DeleteEnvironments	环境删除失败。
OperationDenied.DefaultEnvironment	默认环境不允许操作。
MissingParameter.NeedMoreParams	
InvalidParameter	
FailedOperation.RoleInUse	
FailedOperation.TopicInUse	必须先清除关联主题数据才能继续操作。
FailedOperation.DeleteNamespace	删除命名空间失败。
ResourceNotFound.Cluster	

获取命名空间属性

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取指定命名空间的属性

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-12 16:00:36。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeEnvironmentAttributes
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj

3. 输出参数

参数名称	类型	描述
MsgTTL	Uint64	未消费消息过期时间，单位：秒，最大1296000（15天）。 示例值： 60

参数名称	类型	描述
RateInByte	Uint64	消费速率限制，单位：byte/秒，0：不限速。 示例值： 0
RateInSize	Uint64	消费速率限制，单位：个数/秒，0：不限速。 示例值： 0
RetentionHours	Uint64	已消费消息保存策略，单位：小时，0：消费完马上删除。 示例值： 0
RetentionSize	Uint64	已消费消息保存策略，单位：G，0：消费完马上删除。 示例值： 0
EnvironmentId	String	环境（命名空间）名称。 示例值： default
Replicas	Uint64	副本数。 示例值： 2
Remark	String	备注。 示例值： demo
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError.GetAttributesFailed	获取属性失败。
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
FailedOperation.GetEnvironmentAttributesFailed	获取环境属性失败。
MissingParameter.NeedMoreParams	
InternalError.BrokerService	Broker服务异常。

获取命名空间角色列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取命名空间角色列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:06:38。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeEnvironmentRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	否	否	String	必填字段，环境（命名空间）名称。 示例值： default
Offset	否	否	Int64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Int64	返回数量，不填则默认为10，最大值为20。 示例值： 10
ClusterId	否	否	String	必填字段，Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj
RoleName	否	否	String	角色名称 示例值： role
Filters	否	否	Array of Filter	* RoleName 按照角色名进行过滤，精确查询。 类型：String

参数名称	必选	允许NULL	类型	描述
				必选：否 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	记录数。 示例值： 1
EnvironmentRoleSets	Array of EnvironmentRole	命名空间角色集合。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

获取命名空间列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取租户下命名空间列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-20 15:35:11。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeEnvironments
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	否	否	String	命名空间名称，模糊搜索。 示例值： default
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 5
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj
Filters	否	否	Array of Filter	* EnvironmentId 按照名称空间进行过滤，精确查询。 类型：String 必选：否 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	命名空间记录数。 示例值： 1
EnvironmentSet	Array of Environment	命名空间集合数组。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.Cluster	
FailedOperation	

修改命名空间属性

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改指定命名空间的属性值

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-28 11:01:03。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyEnvironmentAttributes
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	命名空间名称。 示例值： default
MsgTTL	是	否	Uint64	未消费消息过期时间，单位：秒，最大1296000。 示例值： 60
Remark	否	否	String	备注，字符串最长不超过128。 示例值： demo
ClusterId	否	否	String	集群ID 示例值： pulsar-ge3n43v3wj
RetentionPolicy	否	否	RetentionPolicy	消息保留策略 示例值：查看

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	命名空间名称。 示例值： default
MsgTTL	Uint64	未消费消息过期时间，单位：秒。 示例值： 60
Remark	String	备注，字符串最长不超过128。 示例值： demo
NamespaceId	String	命名空间ID 示例值： namespace-ak27qjs68x
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
FailedOperation.UpdateEnvironment	环境更新失败。
OperationDenied.DefaultEnvironment	默认环境不允许操作。
MissingParameter.NeedMoreParams	
ResourceNotFound.Cluster	
InvalidParameterValue.TTL	无效的消息TTL值。
FailedOperation.SetTTL	
ResourceNotFound.Namespace	

消息相关接口
确认消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

根据提供的 MessageID 确认指定 topic 中的消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:51:05。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AcknowledgeMessage
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
MessageId	是	否	String	用作标识消息的唯一的ID（可从 receiveMessage 的返回值中获得） 示例值： 5:7:-1
AckTopic	是	否	String	Topic 名字（可从 receiveMessage 的返回值中获得）这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default 示例值： persistent://tenant/namespace/my-topic
SubName	是	否	String	订阅者的名字，可以从receiveMessage的返回值中获取到。这里尽量与receiveMessage中的订阅者保持一致，否则没办法正确ack 接收回来的消息。 示例值： my-sub

3. 输出参数

参数名称	类型	描述
ErrorMsg	String	如果为“”，则说明没有错误返回 示例值： "ack error"
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.TopicNotFound	上传的topic name错误。
InvalidParameter.TenantNotFound	上传的 tenant name 错误。

接收消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

接收发送到指定 topic 中的消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:50:21。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ReceiveMessage
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Topic	是	否	String	接收消息的topic的名字, 这里尽量需要使用topic的全路径，如果不指定，即：tenant/namespace/topic。默认使用的是：public/default 示例值： persistent://tenant/namespace/my-topic
SubscriptionName	是	否	String	订阅者的名字 示例值： my-sub
ReceiverQueueSize	否	否	Int64	默认值为1000，consumer接收的消息会首先存储到receiverQueueSize这个队列中，用作调优接收消息的速率 示例值： 1000
SubInitialPosition	否	否	String	默认值为：Latest。用作判定consumer初始接收消息的位置，可选参数为：Earliest, Latest 示例值： Latest

3. 输出参数

参数名称	类型	描述
MessageID	String	用作标识消息的唯一主键 示例值： {"Response":{"MessageId":"26365:1475:0"}}
MessagePayload	String	接收消息的内容 示例值： my-payload
AckTopic	String	提供给 Ack 接口，用来Ack哪一个topic中的消息 示例值： my-topic
ErrorMsg	String	返回的错误信息，如果为空，说明没有错误 示例值： sssss
SubName	String	返回订阅者的名字，用来创建 ack consumer时使用 示例值： my-sub
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.TopicNotFound	上传的topic name错误。
InvalidParameter.TenantNotFound	上传的 tenant name 错误。
InvalidParameter.TokenNotFound	没有获取到正确的 token。
FailedOperation.ReceiveTimeout	接收消息超时，请重试。
FailedOperation.ReceiveError	接收消息出错。

批量发送消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量发送消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:46:50。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SendBatchMessages
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Topic	是	否	String	消息要发送的topic的名字, 这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default 示例值： persistent://tenant/namespace/my-topic
Payload	是	否	String	需要发送消息的内容 示例值： hello tdmq
StringToken	否	否	String	String 类型的 token，可以不填，系统会自动获取 示例值： abcdefghijklmnopqrstovwxyz
ProducerName	否	否	String	producer 的名字，要求全局是唯一的，如果不设置，系统会自动生成

参数名称	必选	允许NULL	类型	描述
				示例值： producer1
SendTimeout	否	否	Int64	单位：s。消息发送的超时时间。默认值为：30s 示例值： 30
MaxPendingMessages	否	否	Int64	内存中允许缓存的生产消息的最大数量，默认值：1000条 示例值： 1000
BatchingMaxMessages	否	否	Int64	每一个batch中消息的最大数量，默认值：1000条/batch 示例值： 1000
BatchingMaxPublishDelay	否	否	Int64	每一个batch最大等待的时间，超过这个时间，不管是否达到指定的batch中消息的数量和大小，都会将该batch发送出去，默认：10ms 示例值： 10L
BatchingMaxBytes	否	否	Int64	每一个batch中最大允许的消息的大小，默认：128KB 示例值： 128*1024

3. 输出参数

参数名称	类型	描述
MessageId	String	消息的唯一标识 示例值： {"Response":{"MessageId":"26365:1475:0"}}
ErrorMsg	String	错误消息，返回为 ""，代表没有错误 示例值： "error"
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述

错误码	描述
InvalidParameterValue.TopicNotFound	上传的topic name错误。
InvalidParameter.TenantNotFound	上传的 tenant name 错误。
InvalidParameter.TokenNotFound	没有获取到正确的 token。
FailedOperation.CreateProducerError	创建producer出错。
FailedOperation.CreatePulsarClientError	创建TDMQ client的出错。

发送单条消息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

发送单条消息

默认接口请求频率限制：1000次/秒。

接口更新时间：2021-06-24 14:46:14。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SendMessages
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
StringToken	否	否	String	Token 是用来做鉴权使用的，可以不填，系统会自动获取 示例值： abcdefghijklmnopqrstovwxyz
Topic	是	否	String	消息要发送的topic的名字, 这里尽量需要使用topic的全路径，即：tenant/namespace/topic。如果不指定，默认使用的是：public/default 示例值： persistent://tenant/namespaces/my-topic
Payload	是	否	String	要发送的消息的内容 示例值： hello tdmq
ProducerName	否	否	String	设置 producer 的名字，要求全局唯一，用户不配置，系统会随机生成 示例值： producer

参数名称	必选	允许NULL	类型	描述
SendTimeout	否	否	Int64	设置消息发送的超时时间，默认为30s 示例值： 30
MaxPendingMessages	否	否	Int64	内存中缓存的最大的生产消息的数量，默认为1000条 示例值： 1000

3. 输出参数

参数名称	类型	描述
MessageId	String	消息的messageID, 是全局唯一的，用来标识消息的元数据信息 示例值： {"Response":{"MessageId":"26365:1475:0"}}
ErrorMsg	String	返回的错误消息，如果返回为“”，说明没有错误 示例值： "error"
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.TopicNotFound	上传的topic name错误。
InvalidParameter.TenantNotFound	上传的 tenant name 错误。
InvalidParameter.TokenNotFound	没有获取到正确的 token。
FailedOperation.CreateProducerError	创建producer出错。
FailedOperation.CreatePulsarClientError	创建TDMQ client的出错。

环境角色授权相关接口
检验批量导入角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

检验批量导入角色

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-06 15:21:03。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CheckBatchRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterScope	否	否	String	指定集群id校验 示例值： pulsar-jmw4o94a3z82
BatchRoleBindData	是	否	Array of RoleBindData	批量绑定数据 示例值：查看

3. 输出参数

参数名称	类型	描述
CheckCode	Uint64	0,检测通过，1 检测不通过

参数名称	类型	描述
		示例值： 0
CheckMessage	String	枚举 原因 示例值： Success
AbnormalResources	Array of String	异常资源列表 示例值： ["pulsar-xxxx"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

批量导入角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量导入角色

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 15:36:59。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateBatchRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterScope	是	否	String	指定集群id校验 示例值： pulsar-jmw4o94a3z82
BatchRoleBindData	是	否	Array of RoleBindData	批量绑定数据 示例值：查看

3. 输出参数

参数名称	类型	描述
BatchRoleBindData	Array of RoleBindData	批量绑定数据 示例值：查看

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

创建环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建环境角色授权

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 18:38:36。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateEnvironmentRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devDemo
RoleName	是	否	String	角色名称。 示例值： test_role_1
Permissions	是	否	Array of String	授权项，最多只能包含produce、consume两项的非空字符串数组。 示例值： ["produce"]
ClusterId	否	否	String	必填字段，集群的ID 示例值： pulsar-xk3ne8k2qkp8

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： devDemo
RoleName	String	角色名称。 示例值： test_role_1
Permissions	Array of String	授权项。 示例值： ["produce"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
ResourceInUse.EnvironmentRole	
FailedOperation.CreateEnvironmentRole	
FailedOperation	
ResourceNotFound.Cluster	
FailedOperation.UpdateEnvironmentRole	

创建角色

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建角色

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 15:54:23。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	是	否	String	角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。 示例值： test_role_123
Remark	否	否	String	备注说明，长度必须大等于0且小等于128。 示例值： devDemo
ClusterId	否	否	String	必填字段，集群Id 示例值： pulsar-xk3ne8k2qkp8
EnvironmentRoleSets	否	否	Array of EnvironmentRoleSet	批量操作环境 示例值：查看

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名称 示例值： test_role_123
Token	String	角色token 示例值： eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY
Remark	String	备注说明 示例值： devDemo
EnvironmentRoleSets	Array of EnvironmentRoleSet	批量操作环境信息 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceInUse.Role	
FailedOperation.CreateRole	
FailedOperation	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	
FailedOperation.CreateSecretKey	
FailedOperation.SaveSecretKey	

删除环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除环境角色授权。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:09:02。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteEnvironmentRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
RoleNames	是	否	Array of String	角色名称数组。 示例值： ["test_role"]
ClusterId	否	否	String	必填字段，集群的ID 示例值： pulsar-xk3ne8k2qkp8

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： devNs

参数名称	类型	描述
RoleNames	Array of String	角色名称数组。 示例值： ["test_role"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.EnvironmentRole	
FailedOperation.DeleteEnvironmentRoles	

DescribeBatchRole

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

DescribeBatchRole

默认接口请求频率限制：20次/秒。

接口更新时间：2023-03-27 21:34:59。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeBatchRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	否	String	集群Id 示例值： pulsar-jmw4o94a3z82

3. 输出参数

参数名称	类型	描述
BatchRoleBindData	Array of RoleBindData	role绑定关系 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取角色列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取角色列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-08 16:05:57。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRoles
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	否	否	String	角色名称，模糊查询 示例值： devRole1
Offset	否	否	Int64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Int64	返回数量，不填则默认为10，最大值为20。 示例值： 10
ClusterId	否	否	String	必填字段，集群Id 示例值： pulsar-xk3ne8k2qkp8
Filters	否	否	Array of Filter	* RoleName 按照角色名进行过滤，精确查询。 类型：String 必选：否 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	记录数。 示例值： 4
RoleSets	Array of Role	角色数组。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

修改环境角色授权

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

修改环境角色授权。

默认接口请求频率限制：20次/秒。

接口更新时间：2021-10-09 10:09:49。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyEnvironmentRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
RoleName	是	否	String	角色名称。 示例值： test_role
Permissions	是	否	Array of String	授权项，最多只能包含produce、consume两项的非空字符串数组。 示例值： ["produce"]
ClusterId	否	否	String	必填字段，集群的ID 示例值： pulsar-xk3ne8k2qkp8

3. 输出参数

参数名称	类型	描述
EnvironmentId	String	环境（命名空间）名称。 示例值： default
RoleName	String	角色名称。 示例值： test_role
Permissions	Array of String	授权项。 示例值： ["produce","consume"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.EnvironmentRole	
FailedOperation.UpdateEnvironmentRole	

角色修改

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

角色修改

默认接口请求频率限制：20次/秒。

接口更新时间：2025-10-15 16:57:42。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
RoleName	是	否	String	角色名称，不支持中字以及除了短线和下划线外的特殊字符且长度必须大于0且小等于32。 示例值： test_role
Remark	否	否	String	备注说明，长度必须大等于0且小等于128。 示例值： 更新角色
ClusterId	否	否	String	必填字段，集群Id 示例值： pulsar-xk3ne8k2qkp8
EnvironmentRoleSets	否	否	Array of EnvironmentRoleSet	批量操作环境 示例值：查看

参数名称	必选	允许NULL	类型	描述
UnbindAllEnvironment	否	否	Bool	是否全部解绑环境 示例值： false

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名称 示例值： test_role
Remark	String	备注说明 示例值： Remark1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
AuthFailure.UnauthorizedOperation	
MissingParameter.NeedMoreParams	
ResourceNotFound.Role	
FailedOperation.UpdateRole	
ResourceNotFound.BrokerCluster	

获取用户项目权限列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

通过获取项目权限列表，用于用户集群项目归属的编辑（添加到某个项目，或者从某个项目变更到另一个项目）

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-10 21:08:31。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：TpoDescribeResourcesAdminProjects
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	项目总数量 示例值： 100
ProjectSet	ResourceAdminProject	项目列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InternalError	
InternalError.CloudAPIError	云API错误。

修改资源对应的项目（包含新增）

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

新增或修改资源对应的项目，新增时 oldProjectId 为空

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 22:10:21。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：TpoModifyResourceProject
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ResourceProjectInfos	是	否	Array of ResourceProjectInfo	ResourceProjectInfo 数组，用于新增或修改资源对应的权限 示例值：查看

3. 输出参数

参数名称	类型	描述
ResourceProjectInfos	Array of ResourceProjectInfo	ResourceProjectInfo 数组，新增或修改成功的资源 示例值：查看

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
InternalError	
InternalError.CloudAPIError	云API错误。

批量导入角色预校验

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

批量导入角色预校验

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-26 21:09:39。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：VerifyBatchRole
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterScope	否	否	String	指定集群id校验 示例值： pulsar-jmw4o94a3z82
BatchRoleBindData	是	否	Array of RoleBindData	批量绑定数据 示例值：查看

3. 输出参数

参数名称	类型	描述
CheckCode	Uint64	0,检测通过，1 检测不通过 示例值： 0

参数名称	类型	描述
CheckMessage	String	枚举 原因 示例值： Success
AbnormalResources	Array of String	异常资源列表 示例值： pulsar-xxxx
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

生产消费相关接口
创建订阅关系

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建一个主题的订阅关系

默认接口请求频率限制：200次/秒。

接口更新时间：2021-07-27 11:43:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateSubscription
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
TopicName	是	否	String	主题名称。 示例值： cloud_test
SubscriptionName	是	否	String	订阅者名称，不支持中字以及除了短线和下划线外的特殊字符且不超过150个字符。 示例值： cloud_sub
Remark	否	否	String	备注，128个字符以内。 示例值： 创建订阅关系
IsIdempotent	是	否	Bool	是否幂等创建，若否不允许创建同名的订阅关系。 示例值： true

参数名称	必选	允许NULL	类型	描述
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-5r59xd4vnx
AutoCreatePolicyTopic	否	否	Bool	是否自动创建死信和重试主题，True 表示创建，False表示不创建，默认自动创建死信和重试主题。 示例值： true
PostFixPattern	否	否	String	指定死信和重试主题名称规范，LEGACY表示历史命名规则，COMMUNITY表示Pulsar社区命名规范 示例值： LEGACY

3. 输出参数

参数名称	类型	描述
Result	Bool	创建结果。 示例值： true
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
InternalError.Retry	重试可以成功。
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Topic	
ResourceInUse.Subscription	重名，订阅关系已存在。
FailedOperation.GetTopicPartitionsFailed	获取主题分区数失败。

错误码	描述
FailedOperation.CreateSubscription	创建订阅关系失败。
LimitExceeded.Topics	实例下主题数量超过限制。
LimitExceeded.Subscriptions	实例下订阅者数量超过限制。
FailedOperation	
ResourceNotFound.BrokerCluster	

删除订阅关系

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除订阅关系

默认接口请求频率限制：20次/秒。

接口更新时间：2021-08-04 10:44:23。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteSubscriptions
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
SubscriptionTopicSets	是	否	Array of SubscriptionTopic	订阅关系集合，每次最多删除20个。 示例值：查看
ClusterId	否	否	String	pulsar集群Id。 示例值： pulsar-xk3ne8k2qkp8
EnvironmentId	否	否	String	环境（命名空间）名称。 示例值： devNs
Force	否	否	Bool	是否强制删除，默认为false 示例值： false

3. 输出参数

参数名称	类型	描述
SubscriptionTopicSets	Array of SubscriptionTopic	成功删除的订阅关系数组。 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Topic	
ResourceNotFound.Subscription	订阅关系不存在。
ResourceInUse.Subscription	重名，订阅关系已存在。
MissingParameter.NeedMoreParams	
FailedOperation.GetTopicPartitionsFailed	获取主题分区数失败。
FailedOperation.DeleteSubscriptions	删除订阅关系失败。
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	
OperationDenied.ConsumerRunning	订阅仍在消费中。

生产者列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取生产者列表，仅显示在线的生产者

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-17 17:25:25。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProducers
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： default
TopicName	是	否	String	主题名。 示例值： U_TOPIC_nfa_game_push_retry_kb
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 10
ProducerName	否	否	String	生产者名称，模糊匹配。 示例值： test
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-xxxx

3. 输出参数

参数名称	类型	描述
ProducerSets	Array of Producer	生产者集合数组。 示例值：查看
TotalCount	Uint64	记录总数。 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
ResourceNotFound.Topic	
MissingParameter.NeedMoreParams	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	
FailedOperation.DescribeProducers	查询生产者数据失败。

消费订阅列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

查询指定环境和主题下的订阅者列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-29 16:20:55。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeSubscriptions
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
TopicName	是	否	String	主题名称。 示例值： devtopic
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 10
SubscriptionName	否	否	String	订阅者名称，模糊匹配。 示例值： devSub

参数名称	必选	允许NULL	类型	描述
Filters	否	否	Array of FilterSubscription	数据过滤条件。 示例值：查看
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-b843bz38z5mz

3. 输出参数

参数名称	类型	描述
SubscriptionSets	Array of Subscription	订阅者集合数组。 示例值：查看
TotalCount	Uint64	数量。 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Environment	
ResourceNotFound.Topic	
MissingParameter.NeedMoreParams	
FailedOperation	
ResourceNotFound.BrokerCluster	
ResourceNotFound.Cluster	

错误码	描述
FailedOperation.DescribeSubscription	查询订阅数据失败。

消息回溯

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

根据时间戳进行消息回溯，精确到毫秒

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-21 11:31:34。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ResetMsgSubOffsetByTimestamp
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
EnvironmentId	是	否	String	命名空间名称。 示例值： default
TopicName	是	否	String	主题名称。 示例值： topic
Subscription	是	否	String	订阅者名称。 示例值： subscription
ToTimestamp	是	否	Uint64	时间戳，精确到毫秒。 示例值： 1730690396152
ClusterId	否	否	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj

3. 输出参数

参数名称	类型	描述
Result	Bool	结果。 示例值： true
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameterValue.InvalidParams	
InternalError.SystemError	
AuthFailure.UnauthorizedOperation	
ResourceNotFound.Topic	
ResourceNotFound.Subscription	订阅关系不存在。
MissingParameter.NeedMoreParams	
FailedOperation.ResetMsgSubOffsetByTimestampFailed	消息回溯设置失败。
FailedOperation	
ResourceNotFound.BrokerCluster	

集群相关接口
创建集群

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

创建用户的集群

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-19 11:29:14。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterName	是	否	String	集群名称，不支持中字以及除了短线和下划线外的特殊字符且不超过16个字符。 示例值： devName
BindClusterId	否	否	Uint64	用户专享物理集群ID，如果不传，则默认在公共集群上创建用户集群资源。 示例值： sh_cluster_1
Remark	否	否	String	说明，128个字符以内。 示例值： demo
Tags	否	否	Array of Tag	集群的标签列表 示例值：查看
ProjectId	否	否	String	集群对应的项目id 示例值： 0

参数名称	必选	允许NULL	类型	描述
BindClusterName	否	否	String	绑定集群名 示例值： sh_cluster_1

3. 输出参数

参数名称	类型	描述
ClusterId	String	集群ID 示例值： pulsar-ge3n43v3wj
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
FailedOperation	
ResourceInUse.Cluster	集群已存在。
InternalError.SystemError	
FailedOperation.CreateCluster	
ResourceUnavailable	
AuthFailure.UnauthorizedOperation	
LimitExceeded.Clusters	
MissingParameter.NeedMoreParams	
ResourceUnavailable.CreateFailed	

删除集群

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

删除集群

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-20 14:46:19。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	否	String	集群Id，传入需要删除的集群Id。 示例值： pulsar-ge3n43v3wj

3. 输出参数

参数名称	类型	描述
ClusterId	String	集群的ID 示例值： pulsar-ge3n43v3wj
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
OperationDenied.DefaultEnvironment	默认环境不允许操作。
FailedOperation	
FailedOperation.NamespaceInUse	必须先清除关联命名空间才能继续操作。
FailedOperation.RoleInUse	
FailedOperation.VpcInUse	必须先清除关联VPC路由数据才能继续操作。
FailedOperation.DeleteCluster	
ResourceNotFound.Cluster	

获取专享集群列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取用户绑定的专享集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-09 15:35:05。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeBindClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	专享集群的数量 示例值： 1
ClusterSet	Array of BindCluster	专享集群的列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
ResourceUnavailable.SystemUpgrade	系统升级。
UnsupportedOperation	
UnauthorizedOperation	未授权操作。
ResourceUnavailable	
ResourcesSoldOut	资源售罄。
FailedOperation	
UnknownParameter	未知参数错误。

获取集群详情

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取集群的详细信息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-09 15:51:04。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeClusterDetail
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	否	String	集群的ID 示例值： pulsar-ge3n43v3wj

3. 输出参数

参数名称	类型	描述
ClusterSet	Cluster	集群的详细信息 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取集群列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 17:18:12。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Offset	否	否	Uint64	起始下标，不填默认为0。 示例值： 0
Limit	否	否	Uint64	返回数量，不填则默认为10，最大值为20。 示例值： 1
ClusterName	否	否	String	Pulsar 集群的名称 示例值： devCluster
ClusterIdList	否	否	Array of String	集群ID列表过滤 示例值： ["cluster001"]

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
TotalCount	Int64	集群列表数量 示例值： 1
ClusterSet	Array of Cluster	集群信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

获取用户可以使用物理集群列表

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

获取用户可以使用物理集群列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 17:34:19。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribePClusters
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
Type	是	否	Int64	0表示Pulsar，1表示rocketmq，2表示AMQP，3表示CMQ 示例值： 1

3. 输出参数

参数名称	类型	描述
Total	Uint64	集群总数 示例值： 10
Pclusters	Array of String	集群列表 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

更新集群信息

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

更新集群信息

默认接口请求频率限制：20次/秒。

接口更新时间：2021-07-20 14:45:34。

接口既验签名又鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyCluster
Version	是	否	String	公共参数，本接口取值：2020-02-17
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
ClusterId	是	否	String	Pulsar 集群的ID，需要更新的集群Id。 示例值： pulsar-ge3n43v3wj
ClusterName	否	否	String	更新后的集群名称。 示例值： devName
Remark	否	否	String	说明信息。 示例值： demo

3. 输出参数

参数名称	类型	描述
ClusterId	String	Pulsar 集群的ID 示例值： pulsar-ge3n43v3wj

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
MissingParameter	缺少参数错误。
FailedOperation	
InvalidParameterValue.ClusterNameDuplication	与现有集群名称重复。

数据结构

RocketMQRoleConfig

RocketMQ角色配置信息

被如下接口引用：CreateRocketMQMigrationTask

名称	必选	允许NULL	类型	描述
Remark	否	否	String	备注 示例值： remark info
PermType	否	否	String	权限类型，默认按集群授权（Cluster：集群级别；TopicAndGroup：主题&消费组级别） 示例值： Cluster
DetailedRolePerms	否	否	Array of DetailedRolePerm	Topic和Group维度权限配置 示例值：查看
RoleName	否	否	String	角色名，对应SecretKey 示例值： test_role
RoleToken	否	否	String	accessKey 示例值： akTd0eozCSH78BYp1E
EnvironmentId	否	否	String	命名空间 示例值： test_ns
Permissions	否	否	Array of String	角色权限 示例值：["produce", "consume"]

ServerLog

服务方信息

被如下接口引用：DescribeMsgTrace

名称	必选	允许NULL	类型	描述
SaveTime	是	否	String	存储时间。 示例值： 2024-10-08 11:38:48,068
Status	是	否	String	状态。 示例值： success

RouterTencentMQTTSource

待复制的 MQTT 源集群信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
SourceConsumeConcurrentThreadCount	否	是	Int64	消费者并发数，0 ~ 100 示例值： 0
FilterFromTimestampMs	否	是	Int64	过滤时间，毫秒时间戳 示例值： 0
Endpoint	否	是	String	MQTT集群接入点信息 示例值： http://mqtt-sk3bsed7.com

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	MQTT 集群ID 示例值： mqtt-demo123
Topic	是	是	String	主题名 示例值： topic_test
SourceRegion	是	是	String	地域 示例值： ap-guangzhou
UserName	是	是	String	用户名 示例值： admin
Password	否	是	String	用户密码 示例值： test-qwe

EnvironmentRole

环境角色集合

被如下接口引用：DescribeEnvironmentRoles、DescribeRocketMQEnvironmentRoles

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Datetime	创建时间。 示例值： 2023-07-18 17:06:57
UpdateTime	是	否	Datetime	更新时间。 示例值： 2023-07-18 17:06:57
EnvironmentId	是	否	String	环境（命名空间）。 示例值： test_namespace
RoleName	是	否	String	角色名称。 示例值： test_role_name
Permissions	是	否	Array of String	授权项，最多只能包含produce、consume两项的非空字符串数组。 示例值： ["produce"]
RoleDescribe	是	否	String	角色描述。 示例值： 测试角色

DeliveryInfoOpt

配置下发状态信息

被如下接口引用：DescribeDeliveryStatusOpt

名称	必选	允许NULL	类型	描述
DeliveryIp	是	否	String	代理节点IP 示例值： 10.73.13.97
DeliveryTimes	是	是	Int64	下发时间 示例值： 1
DeliveryStatus	是	否	Int64	下发状态（实际大小1字节） 0 下发中 1 陈工 2失败 示例值： 2
ConfigVersion	是	否	String	版本 示例值： 1
AddTime	是	否	Int64	新增时间 示例值： 1111

RocketMQClusterConfig

RocketMQ集群配置

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

名称	必选	允许NULL	类型	描述
MaxNamespaceNum	是	否	Uint64	最大命名空间数量 示例值： 10
UsedNamespaceNum	是	否	Uint64	已使用命名空间数量 示例值： 3
MaxTopicNum	是	否	Uint64	最大Topic数量 示例值： 1000
UsedTopicNum	是	否	Uint64	已使用Topic数量 示例值： 8
UsedGroupNum	是	否	Uint64	已使用Group数量 示例值： 10
MaxRetentionTime	是	否	Uint64	消息最大保留时间，以毫秒为单位 示例值： 100000
MaxTpsLimit	否	否	Int64	TPS限额 示例值： 4000
MaxTpsPerNamespace	是	否	Uint64	单命名空间TPS上限 示例值： 8000
MaxGroupNum	是	否	Uint64	最大Group数量 示例值： 10000
MaxLatencyTime	是	否	Uint64	消息最长延时，以毫秒为单位 示例值： 100000
MaxQueuesPerTopic	是	是	Uint64	单个主题最大队列数 示例值： 16
TopicDistribution	是	是	Array of RocketMQTopicDistribution	topic分布 示例值：查看
MaxRoleNum	否	否	Int64	最大角色数量 示例值： 100

RocketMQTopicConfig

RocketMQ主题配置信息

被如下接口引用：CreateRocketMQMigrationTask、ImportRocketMQTopics

名称	必选	允许NULL	类型	描述
Namespace	是	否	String	命名空间 示例值： test_ns
TopicName	是	否	String	主题名称 示例值： test_topic
Type	是	否	String	主题类型： Normal，普通 PartitionedOrder, 分区顺序 Transaction，事务消息 DelayScheduled，延迟/定时消息 示例值： Normal

名称	必选	允许NULL	类型	描述
Partitions	是	否	Int64	分区个数 示例值： 3
Remark	否	否	String	备注信息 示例值： remark info

RocketMQGroupConfig

RocketMQ消费组配置信息

被如下接口引用：CreateRocketMQMigrationTask、ImportRocketMQConsumerGroups

名称	必选	允许NULL	类型	描述
ConsumeBroadcastEnable	是	否	Bool	是否开启广播消费 示例值： false
ConsumeEnable	是	否	Bool	是否开启消费 示例值： false
Remark	否	否	String	备注信息 示例值： remark info
ConsumerGroupType	否	否	String	协议类型，支持以下枚举值 TCP; HTTP; 示例值： TCP
Namespace	是	否	String	命名空间 示例值： test_ns
GroupName	是	否	String	消费组名称 示例值： test_group

ProducerLog

消息生产信息

被如下接口引用：DescribeMsgTrace

名称	必选	允许NULL	类型	描述
MsgId	是	否	String	消息ID。 示例值： 6054156:22067:16
ProducerName	否	否	String	生产者名称。 示例值： pulsar-7dqvnqwjgzm-2156-9
ProduceTime	是	否	String	消息生产时间。 示例值： 2024-10-08 14:45:52,636
ProduceUseTime	是	否	Uint64	生产耗时（秒）。 示例值： 2
Status	是	否	String	状态。 示例值： success

RocketMQSubscription

RocketMQ消费组订阅信息

被如下接口引用：DescribeRocketMQSubscriptions、DescribeRocketMQTopic、DescribeRocketMQTopics

名称	必选	允许NULL	类型	描述
ConsumerGroup	否	是	String	消费组名称 示例值： test_group
ClientProtocol	否	是	String	协议类型 示例值： TCP/HTTP
Type	是	是	String	主题类型： Normal 普通, GlobalOrder 全局顺序, PartitionedOrder 局部顺序, Transaction 事务消息, DelayScheduled 延时消息, Retry 重试, DeadLetter 死信 示例值： DeadLetter
IsOnline	否	是	Bool	是否在线 示例值： false
ConsumeType	否	是	Int64	消费类型 0: 广播消费 1: 集群消费 示例值： 1
Topic	是	否	String	主题名称 示例值： test_topic
ConsumerLag	是	是	Int64	消费堆积数量 示例值： 260
ClusterId	否	是	String	实例ID 示例值： rocketmq-2p9vx3ax9jxg
ClientSubscriptionInfos	否	是	Array of ClientSubscriptionInfo	客户端订阅详情 示例值：查看
Consistency	否	是	Int64	订阅一致性 示例值： 1
LastUpdateTime	否	是	Int64	最后消费进度更新时间，秒为单位 示例值： 1727420571
MaxRetryTimes	否	是	Int64	最大重试次数 示例值： 10
PartitionNum	是	是	Int64	分区数 示例值： 3
ExpressionType	是	是	String	过滤模式，TAG，SQL 示例值： TAG
SubString	是	是	String	过滤表达式 示例值： test_tag
Status	是	是	Int64	订阅状态： 0，订阅关系一致 1，订阅关系不一致 2，未知 示例值： 0

TopicRecord

主题关键信息

被如下接口引用：DeleteTopics

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
TopicName	是	否	String	主题名称。 示例值： devTopic

AMQPVHostConnection

Vhost客户端连接情况

被如下接口引用：DescribeAMQPVHostConnections

名称	必选	允许NULL	类型	描述
ClientAddr	是	否	String	客户端实例的地址和端口 示例值： 192.168.0.1:5555
Protocol	是	否	String	客户端协议 示例值： AMQP 0.9.1
Security	是	否	Bool	是否支持SSL/TLS 示例值： true
ChannelNum	是	否	Uint64	信道数 示例值： 2
ReceiveRate	是	否	Uint64	接受速率 示例值： 102
SendRate	是	否	Uint64	发送速率 示例值： 97

BusinessSystem

单元化客户映射服务地址

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
BusinessSystemName	否	否	String	业务系统名称 示例值： 无
BusinessSystemType	否	否	String	业务系统类型 示例值： 无
BusinessSystemId	否	否	String	业务系统 ID 示例值： 无

MsgLog

消息日志

被如下接口引用：DescribeTopicMsgs

名称	必选	允许NULL	类型	描述
MsgId	否	是	String	消息ID。 示例值： 931087:81:0
ProducerName	否	是	String	生产者名称。 示例值： devProducerName

名称	必选	允许NULL	类型	描述
ProduceTime	否	是	String	生产时间。 示例值： 2024-10-08 08:04:13,809
ProducerAddr	否	是	String	生产客户端地址。 示例值： 10.0.0.120:38168

RouterMessageServiceTarget

rocketmq router mns 目标信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
SecretKey	是	是	String	secret key 示例值： -
TargetProduceConcurrentThreadCount	否	是	Int64	发送mns的并发数 示例值： 1
Topic	否	是	String	topic名字。可以在https://mns.console.aliyun.com/region/cn-shenzhen/topics 看到 示例值： -
AccessKey	是	是	String	access key 示例值： -
Endpoint	是	是	String	mns接入点，需要和router内网联通，一般是公网接入点 示例值： http://1111111.mns.cn-shenzhen.aliyuncs.com
Queue	否	是	String	mns queue名字。可以在https://mns.console.aliyun.com/region/cn-shenzhen/queues 看到 示例值： -
SendWithBase64	否	是	String	是否用base64编码发送mns消息。 ON：发送base64编码的消息体到mns OFF：将原始消息体发送到mns 示例值： OFF

RocketMQTopicRouteAccessInfo

访问主题路由的信息

被如下接口引用：DescribeRocketMQMigrateTopicAccessInfo

名称	必选	允许NULL	类型	描述
LastAccessTimestamp	否	是	Uint64	最近访问的时间戳，毫秒为单位 示例值： 无
ClientIP	否	是	String	客户端地址 示例值： 无

AMQPExchange

AMQP Exchange信息

被如下接口引用：DescribeAMQPExchanges

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
Name	是	否	String	Exchange名称 示例值： test
Type	是	否	String	Exchange的类别，为枚举类型:Direct, Fanout, Topic 示例值： Default
SourceBindedNum	是	否	Uint64	主绑定数 示例值： 1
Remark	是	是	String	说明 示例值： 备注
DestBindedNum	是	否	Uint64	被绑定数 示例值： 1
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000
UpdateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000
Internal	是	否	Bool	是否为内部Exchange(以amq.前缀开头的) 示例值： false

RouterTencentMQTTTarget

MQTT 目标集群信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
Endpoint	否	是	String	MQTT集群接入点信息 示例值： http://mqtt-sk3bsed7.com
ClusterId	是	是	String	MQTT 集群ID 示例值： mqtt-demo123
Topic	是	是	String	主题名 示例值： topic_test
SourceRegion	是	是	String	地域 示例值： ap-guangzhou
UserName	是	是	String	用户名 示例值： admin
Password	否	是	String	用户密码 示例值： text-qwe
TargetProduceConcurrentThreadCount	否	是	Int64	消费者并发数，0 ~ 100 示例值： 0
FilterFromTimestampMs	否	是	Int64	过滤时间，毫秒时间戳 示例值： 0

SDUTrafficRule

单元化流量路径

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
SDUId	否	否	String	SDU ID 示例值： 无
CloudSpaceId	否	否	String	云空间 ID 示例值： 无
GDUId	否	否	String	GDU ID 示例值： 无

RocketMQConsumerTopic

消费者详情中的主题信息

被如下接口引用：DescribeRocketMQConsumerConnectionDetail

名称	必选	允许NULL	类型	描述
Accumulative	是	否	Int64	消息堆积数 示例值： 100
LastConsumptionTime	是	否	Uint64	最后消费时间，以毫秒为单位 示例值： 1697945990000
SubRule	是	是	String	订阅规则 示例值： test_tag
Topic	是	否	String	主题名称 示例值： test_topic
Type	是	否	String	主题类型，Normal表示普通，GlobalOrder表示全局顺序，PartitionedOrder表示局部顺序，Transaction表示事务，Retry表示重试，DeadLetter表示死信 示例值： Normal
PartitionNum	是	否	Uint64	分区数 示例值： 3

BindCluster

用户专享集群信息

被如下接口引用：DescribeBindClusters

名称	必选	允许NULL	类型	描述
ClusterId	是	否	Int64	物理集群的ID 示例值： pulsar-mw4qmgaq298g
ClusterName	是	否	String	物理集群的名称 示例值： devClusterName

ConsumersSchedule

消费进度详情

被如下接口引用：DescribeSubscriptions

名称	必选	允许NULL	类型	描述
Partitions	是	是	Uint64	当前分区id。 示例值： 1
NumberOfEntries	是	是	Uint64	消息数量。 示例值： 1000

名称	必选	允许NULL	类型	描述
MsgBacklog	是	是	Uint64	消息积压数量。 示例值： 2
MsgRateOut	是	否	String	消费者每秒分发消息的数量之和。 示例值： 0.0
MsgThroughputOut	是	否	String	消费者每秒消息的byte。 示例值： 0.0
MsgRateExpired	是	是	String	超时丢弃比例。 示例值： 0.0

TagFilter

标签过滤器

被如下接口引用：DescribeRocketMQGroups、DescribeRocketMQTopics

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键名称 示例值： test_key
TagValues	是	否	Array of String	标签值列表 示例值： test_value

AMQPClusterConfig

AMQP集群配置

被如下接口引用：DescribeAMQPCluster、DescribeAMQPClusters

名称	必选	允许NULL	类型	描述
MaxTpsPerVHost	是	否	Uint64	单Vhost TPS上限 示例值： 8000
MaxConnNumPerVHost	是	否	Uint64	单Vhost客户端连接数上限 示例值： 8000
MaxVHostNum	是	否	Uint64	最大Vhost数量 示例值： 10
MaxExchangeNum	是	否	Uint64	最大exchange数量 示例值： 100
MaxQueueNum	是	否	Uint64	最大Queue数量 示例值： 100
MaxRetentionTime	是	否	Uint64	消息最大保留时间，以毫秒为单位 示例值： 100000
UsedVHostNum	是	否	Uint64	已使用Vhost数量 示例值： 3
UsedExchangeNum	是	否	Uint64	已使用exchange数量 示例值： 3
UsedQueueNum	是	否	Uint64	已使用queue数量 示例值： 8

VpcConfig

VPC配置信息

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

名称	必选	允许NULL	类型	描述
VpcId	是	否	String	vpc的id 示例值： vpc-9dlrd5h1
SubnetId	是	否	String	子网id 示例值： subnet-jadmas

RocketMQRouterRuleInfo

rocketmq router规则基本信息

被如下接口引用：CreateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
SourceType	是	否	String	源类型。 OPEN_SOURCE_ROCKETMQ：开源rocketmq ALI_ROCKETMQ：阿里云rocketmq TENCENT_ROCKETMQ：腾讯云rocketmq TENCENT_MQTT：腾讯云MQTT ALI_MNS：阿里云mns 示例值： TENCENT_ROCKETMQ
TargetType	是	否	String	目标类型。 枚举和SourceType字段一样 示例值： TENCENT_ROCKETMQ
AliMessageServiceSource	否	否	RouterMessageServiceSource	阿里云mns源信息 示例值：查看
AliMessageServiceTarget	否	否	RouterMessageServiceTarget	阿里云mns目标信息 示例值：查看
TenRocketMQTarget	否	否	RouterTencentRocketMQTarget	腾讯云rocketmq目标信息 示例值：查看
AliasName	否	否	String	别名 示例值： -
TenMQTTTarget	否	否	RouterTencentMQTTTarget	腾讯云 MQTT 目标集群信息 示例值：查看
RemarkName	否	否	String	规则备注，创建后任务后可以修改 示例值： -
AliRocketMQSource	否	否	RouterRocketMQSource	阿里云rocketmq源信息 示例值：查看
AliRocketMQTarget	否	否	RouterRocketMQTarget	阿里云rocketmq目标信息 示例值：查看
TenRocketMQSource	否	否	RouterTencentRocketMQSource	腾讯云rocketmq源信息 示例值：查看
TenMQTTSource	否	否	RouterTencentMQTTSource	腾讯云 MQTT 源集群信息 示例值：查看

RocketMQTopicDistribution

RocketMQtopic分布情况

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

名称	必选	允许NULL	类型	描述
TopicType	是	否	String	topic类型 示例值： Normal
Count	是	否	Uint64	topic数量 示例值： 10

RouteRecord

路由记录

被如下接口引用：DescribeRoute

名称	必选	允许NULL	类型	描述
VpcId	是	是	String	私有网络 示例值： xxx
SubnetId	是	是	String	子网id 示例值： xxx
RouterId	是	否	String	路由id 示例值： pulsar-xxx
Vip	是	否	String	vip地址 示例值： 10.10.10.10
Vport	是	否	Uint64	vport记录 示例值： 6000
Remark	是	是	String	备注信息 示例值： xxx
NetType	是	否	Uint64	路由类型 1: 公网 2：vpc 4：支撑网 示例值： 1
Id	否	否	Uint64	路由Id（int 类型） 示例值： route-ID

RocketMQVipInstance

RocketMQ专享实例信息

被如下接口引用：DescribeRocketMQVipInstances

名称	必选	允许NULL	类型	描述
Retention	是	否	Int64	实例消息保留时间，小时为单位 示例值： 72
InstanceId	是	否	String	实例id 示例值： rmq-4k4orqgq1
ConfigDisplay	是	否	String	实例配置规格名称 示例值： 基础型
MaxStorage	是	否	Uint64	存储容量，GB为单位 示例值： 100
Remark	是	否	String	备注信息 示例值： remark-info
SpecName	是	否	String	实例配置ID 示例值： vip-basic-1

名称	必选	允许NULL	类型	描述
CanVerticalChange	是	否	Bool	是否可以垂直变配 示例值： false
InstanceName	是	否	String	实例名称 示例值： order-instance
MaxBandWidth	是	否	Uint64	峰值带宽，Mbps为单位 示例值： 1200
AclEnabled	是	否	Bool	是否开启ACL鉴权 示例值： true
Status	是	否	Uint64	实例状态，0表示创建中，1表示正常，2表示隔离中，3表示已销毁，4 - 异常, 5 - 发货失败，6 - 变配中，7 - 变配失败 示例值： 0
ExpireTime	是	否	Uint64	实例到期时间，毫秒为单位 示例值： 100000
AutoRenewFlag	是	否	Uint64	自动续费标记，0表示默认状态(用户未设置，即初始状态即手动续费)， 1表示自动续费，2表示明确不自动续费(用户设置) 示例值： 0
PayMode	是	否	Uint64	0-后付费，1-预付费 示例值： 0
DestroyTime	否	否	Uint64	销毁时间 示例值： 369182637452345
InstanceVersion	是	否	String	实例版本 示例值： 4.9.3
NodeCount	是	否	Uint64	节点数量 示例值： 5
MaxTps	是	否	Uint64	峰值TPS 示例值： 10000
MaxRetention	是	否	Int64	最大可设置消息保留时间，小时为单位 示例值： 168
MinRetention	是	否	Int64	最小可设置消息保留时间，小时为单位 示例值： 24

ConsumerLogs

消费信息

被如下接口引用：DescribeMsgTrace

名称	必选	允许NULL	类型	描述
TotalCount	是	是	Uint64	记录数。 示例值： 1
ConsumerLogSets	是	是	Array of ConsumerLog	消费日志。 示例值：查看

TemplateItemInfoOpt

模板配置项

被如下接口引用：DescribeTemplateItemOpt

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ItemKey	是	否	String	配置属性名 示例值： key
ItemDefaultValue	是	是	String	配置项默认值 示例值： value
ItemValue	是	是	String	模板配置项值 示例值： value
ItemDesc	是	是	String	描述 示例值： desc
IsDynamic	是	否	Bool	是否动态配置，0需要重启，1不需要重启 示例值： 0

AMQPClusterInfo

AMQP集群基本信息

被如下接口引用：DescribeAMQPCluster、DescribeAMQPClusters

名称	必选	允许NULL	类型	描述
ClusterId	是	否	String	集群ID 示例值： amqp-xxxxx
ClusterName	是	否	String	集群名称 示例值： test
Region	是	否	String	地域信息 示例值： 某地区
CreateTime	是	否	Uint64	创建时间，毫秒为单位 示例值： 100000000
Remark	是	是	String	集群说明信息 示例值： 备注
PublicEndPoint	是	是	String	公网接入地址 示例值： xxxx
VpcEndPoint	是	是	String	VPC接入地址 示例值： xxxxx

Environment

命名空间信息

被如下接口引用：DescribeEnvironments

名称	必选	允许NULL	类型	描述
EnvironmentId	是	否	String	命名空间名称 示例值： devNs
Remark	是	否	String	说明 示例值： devRemark
MsgTTL	是	否	Int64	未消费消息过期时间，单位：秒，最大1296000（15天） 示例值： 2
CreateTime	是	否	Datetime	创建时间 示例值： 2020-02-17 00:00:00

名称	必选	允许NULL	类型	描述
UpdateTime	是	否	Datetime	最近修改时间 示例值： 2020-02-17 00:00:00
NamespaceId	是	否	String	命名空间ID 示例值： devName1
NamespaceName	是	否	String	命名空间名称 示例值： devName1
TopicNum	是	是	Int64	Topic数量 示例值： 2
RetentionPolicy	是	是	RetentionPolicy	消息保留策略 示例值：查看

ConsumerLog

消费日志

被如下接口引用：DescribeMsgTrace

名称	必选	允许NULL	类型	描述
MsgId	是	否	String	消息ID。 示例值： 1:1:1
ConsumerGroup	是	否	String	消费组。 示例值： group
ConsumerName	是	否	String	消费组名称。 示例值： name
ConsumeTime	是	否	String	消费时间。 示例值： 2021-11-02 17:53:00,12,654
ConsumerAddr	是	否	String	消费者客户端地址。 示例值： 1.1.1.1
ConsumeUseTime	是	否	Uint64	消费耗时（毫秒）。 示例值： 12
Status	是	否	String	消费状态。 示例值： 0

Topic

主题实例

被如下接口引用：DescribeTopics

名称	必选	允许NULL	类型	描述
AverageMsgSize	是	是	String	最后一次间隔内发布消息的平均byte大小。 示例值： 10
ConsumerCount	是	是	String	消费者数量。 示例值： 120
LastConfirmedEntry	是	是	String	被记录下来的消息总数。 示例值： 100
LastLedgerCreatedTimestamp	是	是	String	最后一个ledger创建的时间。 示例值： 2024-06-19 14:15:54

名称	必选	允许NULL	类型	描述
MsgRateIn	是	是	String	本地和复制的发布者每秒发布消息的速率。 示例值： 130
MsgRateOut	是	是	String	本地和复制的消费者每秒分发消息的数量之和。 示例值： 140
MsgThroughputIn	是	是	String	本地和复制的发布者每秒发布消息的byte。 示例值： 160
MsgThroughputOut	是	是	String	本地和复制的消费者每秒分发消息的byte。 示例值： 120
NumberOfEntries	是	是	String	被记录下来的消息总数。 示例值： 100
Partitions	是	是	Int64	分区数<=0：topic下无子分区。 示例值： 2
ProducerCount	是	是	String	生产者数量。 示例值： 2
TotalSize	是	是	String	以byte计算的所有消息存储总量。 示例值： 2
SubTopicSets	是	是	Array of PartitionsTopic	分区topic里面的子分区。 示例值：查看
TopicType	是	是	Uint64	topic类型描述： 0：普通消息； 1：全局顺序消息； 2：局部顺序消息； 3：重试队列； 4：死信队列； 5：事务消息。 示例值： 0
EnvironmentId	是	是	String	环境（命名空间）名称。 示例值： devNs
TopicName	是	是	String	主题名称。 示例值： devTopic
Remark	是	是	String	说明，128个字符以内。 示例值： devRemark
CreateTime	是	是	Datetime	创建时间。 示例值： 2024-06-19 14:15:54
UpdateTime	是	是	Datetime	最近修改时间。 示例值： 2024-06-19 14:15:54
ProducerLimit	是	是	String	生产者上限。 示例值： 5000
ConsumerLimit	是	是	String	消费者上限。 示例值： 5000
PulsarTopicType	否	是	Int64	0: 非持久非分区 1: 非持久分区 2: 持久非分区 3: 持久分区 示例值： 1
ReplicationStatus	否	是	String	topic所在集群的消息同步状态OriginReplicationOpen 源集群开启消息同步OriginReplicationClose 源集群关闭消息同步TargetReplicationOpen 目标集群开启消息同步TargetReplicationClose 目标集群关闭消息同步 示例值： success
MsgTTL	否	是	Uint64	未消费消息过期时间，单位：秒 示例值： 60

RouterRocketMQSource

rocketmq router的rocketmq源信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
Namespace	否	是	String	4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如： MQ_INST_rocketmxxx57d53rnn5_tiger_namespace 示例值： -
Topic	是	是	String	topic名字 示例值： -
SecretKey	是	是	String	secret key 示例值： -
SourceConsumeConcurrentThreadCount	否	是	Int64	消费者并发数，0 ~ 100 示例值： 1
AccessKey	是	是	String	access key 示例值： -
FilterExpression	是	是	String	过滤表达式, 参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java 示例值： *
FilterFromTimestampMs	是	是	Int64	过滤时间，毫秒时间戳 示例值： 1691983398481
FilterType	是	是	String	过滤类型，参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java 示例值： TAG
Group	是	是	String	消费者组名。会使用该消费者来消费消息 示例值： xxx-group
NameServerAddressList	是	是	String	namesrv地址列表，也可以是云厂商提供的接入点信息 示例值： 127.0.0.1:9876

RouterRocketMQTarget

rocketmq router任务的目标信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
TargetProduceConcurrentThreadCount	否	是	Int64	生产者并发数，可以不用填，可以在规则创建后修改 示例值： 1
AccessKey	是	是	String	access key 示例值： -
NameServerAddressList	是	是	String	namesrv地址列表，也可以是云厂商提供的接入点信息 示例值： 127.0.0.1:9876
Namespace	否	是	String	4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如：MQ_INST_rocketmqka57d53rnn5_tiger_namespace 示例值： -

名称	必选	允许NULL	类型	描述
SecretKey	是	是	String	secret key 示例值： -
Topic	是	是	String	topic名字 示例值： -

AMQPQueueDetail

AMQP 队列信息

被如下接口引用：DescribeAMQPQueues

名称	必选	允许NULL	类型	描述
Name	是	否	String	Queue名称 示例值： test
Remark	是	是	String	说明 示例值： 备注
DestBindedNum	是	是	Uint64	被绑定数 示例值： 1
CreateTime	是	是	Uint64	创建时间，以毫秒为单位 示例值： 100000
UpdateTime	是	是	Uint64	创建时间，以毫秒为单位 示例值： 100000
OnlineConsumerNum	是	是	Uint64	在线消费者数 示例值： 20
Tps	是	是	Uint64	每秒钟的事务数 示例值： 120
AccumulativeMsgNum	是	是	Uint64	消息堆积数 示例值： 90
AutoDelete	是	是	Bool	是否自动删除 示例值： true
DeadLetterExchange	是	是	String	死信交换机 示例值： test
DeadLetterRoutingKey	是	是	String	死信交换机路由键 示例值： test

AMQPVHost

vhostd信息

被如下接口引用：DescribeAMQPVHosts

名称	必选	允许NULL	类型	描述
VHostId	是	否	String	命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： prod
MsgTtl	是	否	Uint64	未消费消息的保留时间，以毫秒单位，范围60秒到15天 示例值： 100000
Remark	是	是	String	备注 示例值： 备注

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000
UpdateTime	是	否	Uint64	更新时间，以毫秒为单位 示例值： 100000
Username	是	否	String	用户名 示例值： xxxxx
Password	是	否	String	密码 示例值： xxxxxx

RocketMQGroup

RocketMQ消费组信息

被如下接口引用：DescribeRocketMQConsumerConnections、DescribeRocketMQGroups

名称	必选	允许NULL	类型	描述
Name	是	否	String	消费组名称 示例值： test_group
TotalAccumulative	是	否	Int64	总堆积数量 示例值： 380
ClientProtocol	是	否	String	客户端协议 示例值： tcp
RetryMaxTimes	是	是	Uint64	重试次数 示例值： 16
SubscribeTopicNum	否	否	Int64	订阅的主题个数 示例值： 10
ConsumerNum	是	否	Uint64	在线消费者数量 示例值： 2
TPS	是	否	Uint64	消费TPS 示例值： 100
ConsumptionMode	是	否	Int64	0表示集群消费模式，1表示广播消费模式，-1表示未知 示例值： 0
RetryPartitionNum	是	是	Uint64	重试队列分区数 示例值： 2
UpdateTime	是	否	Uint64	修改时间，以毫秒为单位 示例值： 1729478721
Remark	是	是	String	说明信息 示例值： 备注信息
ConsumerType	是	是	String	消费者类型，枚举值ACTIVELY, PASSIVELY 示例值： PUSH
BroadcastEnabled	是	否	Bool	是否开启广播消费 示例值： false
ReadEnabled	是	否	Bool	是否允许消费 示例值： true
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 1729478721
GroupType	是	是	String	Group类型 示例值： TCP

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID 示例值： rocketmq-4k4orqgq
Namespace	是	是	String	命名空间 示例值： test_namespace
TagList	否	否	Array of Tag	绑定的标签列表 示例值：查看

TraceResult

消息轨迹结果

被如下接口引用：DescribeRocketMQMsgTrace

名称	必选	允许NULL	类型	描述
Stage	是	否	String	阶段 示例值： produce
Data	是	否	String	内容详情 示例值： {"", ""}

Tag

标签的key/value的类型

被如下接口引用：CreateCluster、CreateRocketMQCluster、CreateRocketMQGroupV2、DescribeClusterDetail、DescribeClusters、DescribeRocketMQClusters、DescribeRocketMQConsumerConnections、DescribeRocketMQGroups、DescribeRocketMQTopic、DescribeRocketMQTopics

名称	必选	允许NULL	类型	描述
TagKey	是	是	String	标签的key的值 示例值： test_tag_key
TagValue	是	是	String	标签的Value的值 示例值： test_tag_value1

SubscriptionTopic

订阅关系

被如下接口引用：DeleteSubscriptions

名称	必选	允许NULL	类型	描述
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
TopicName	是	否	String	主题名称。 示例值： devTopic
SubscriptionName	是	否	String	订阅名称。 示例值： devSub

DetailedRolePerm

Topic&Group维度的权限配置

被如下接口引用：CreateRocketMQEnvironmentRole、CreateRocketMQGeneralRole、CreateRocketMQMigrationTask、ModifyRocketMQEnvironmentRole

名称	必选	允许NULL	类型	描述
Remark	否	否	String	资源备注 示例值： test remark info.
Resource	是	否	String	权限对应的资源 示例值： test_topic
PermWrite	是	否	Bool	是否开启生产权限 示例值： true
PermRead	是	否	Bool	是否开启消费权限 示例值： false
ResourceType	是	否	String	授权资源类型，枚举值如下： - Topic：主题维度 - Group：消费组维度 - Cluster：集群维度（默认值） 示例值： Topic

ConfigModuleInfoOpt

配置模块

被如下接口引用：DescribeModuleListOpt

名称	必选	允许NULL	类型	描述
ModuleName	是	否	String	模块名 示例值： broker
ComponentName	是	否	String	组件名 示例值： BROKER
ConfigName	是	否	String	配置文件名 示例值： broker.conf
IsShell	是	否	Bool	是否脚本文件 示例值： false

RocketMQMsgLog

rocketmq消息日志

被如下接口引用：DescribeRocketMQTopicMsgs

名称	必选	允许NULL	类型	描述
DeadLetterResendTimes	是	是	Int64	死信重发次数 示例值： 10
ResendSuccessCount	是	是	Int64	死信重发成功次数 示例值： 10
MsgId	是	否	String	消息id 示例值： 092BAE5A1656070DEA4E276DF0760089
MsgTag	是	是	String	消息tag 示例值： test_tag
MsgKey	是	是	String	消息key 示例值： test_key
ProducerAddr	是	否	String	客户端地址 示例值： 9.43.174.90

名称	必选	允许NULL	类型	描述
ProduceTime	是	否	String	消息发送时间 示例值： 2023-10-08 15:45:16
PulsarMsgId	是	否	String	pulsar消息id 示例值： 092BAE5A1656070DEA4E276DF0760089

AMQPClusterRecentStats

AMQP集群近期使用量

被如下接口引用：DescribeAMQPCluster

名称	必选	允许NULL	类型	描述
QueueNum	是	否	Uint64	Queue数量 示例值： 20
ProducedMsgNum	是	否	Uint64	消息生产数 示例值： 11000
AccumulativeMsgNum	是	否	Uint64	消息堆积数 示例值： 22000
ExchangeNum	是	否	Uint64	Exchange数量 示例值： 20

AgentNodeOpt

代理节点

被如下接口引用：DescribeAgentNodesOpt

名称	必选	允许NULL	类型	描述
ClusterName	是	否	String	集群名称 示例值： tdmq_txy_gz_01
ModuleName	是	否	String	模块名称 示例值： ZOOKEEPER_BOOKIE
AgentId	是	否	String	代理ID 示例值： 10.73.13.46-fqLzhQ6B
AgentIp	是	否	String	代理IP 示例值： 10.73.13.46
ProcId	是	否	Int64	协议ID 示例值： 8451
AgentVersion	是	否	String	代理版本 示例值： 0.0.1-SNAPSHOT
JavaVersion	是	否	String	JDK版本 示例值： 1.8.0_275

DeploymentUnitHealthStatusResult

DeploymentUnitHealthStatus返回值

被如下接口引用：DescribeDeploymentUnitHealthStatus

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
Enabled	否	是	Bool	是否启用单元化 示例值： 无
Data	否	是	String	扩展信息 示例值： 无

Filter

过滤参数

被如下接口引用：DescribeEnvironmentRoles、DescribeEnvironments、DescribeInternalRocketMQInstances、DescribeInternalRocketMQNamespaces、DescribeRocketMQClusters、DescribeRocketMQEnvironmentRoles、DescribeRocketMQMigrateTopicAccessInfo、DescribeRocketMQMigrationTaskDetail、DescribeRocketMQMigrationTasks、DescribeRocketMQRoles、DescribeRocketMQVipInstances、DescribeRoles、DescribeTopics

名称	必选	允许NULL	类型	描述
Name	否	否	String	过滤参数的名字 示例值： paramName
Values	否	否	Array of String	数值 示例值： ["a"]

RocketMQTopic

RocketMQ主题信息

被如下接口引用：DescribeRocketMQTopic、DescribeRocketMQTopics

名称	必选	允许NULL	类型	描述
InstanceId	是	是	String	实例ID 示例值： rocketmq-2p9vx3ax9jxg
Namespace	是	是	String	命名空间 示例值： test_namespace
LastUpdateTime	否	是	Int64	最后写入时间，单位为秒 示例值： 1727420571
SubscriptionCount	否	是	Int64	订阅数量 示例值： 10
Type	是	否	String	主题的类别，为枚举类型，Normal，GlobalOrder，PartitionedOrder，Transaction，Retry及DeadLetter 示例值： Normal
GroupNum	是	否	Uint64	订阅组数量 示例值： 5
Remark	是	是	String	说明 示例值： 备注信息
SubscriptionData	否	是	Array of RocketMQSubscription	订阅关系列表 示例值：查看
TagList	否	否	Array of Tag	绑定的标签列表 示例值：查看
Name	是	否	String	主题名称 示例值： test_topic
PartitionNum	是	否	Uint64	读写分区数 示例值： 3

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000
UpdateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000

PublicAccessRule

公网访问安全规则

被如下接口引用：DescribeRocketMQPublicAccessPoint

名称	必选	允许NULL	类型	描述
IpRule	是	否	String	ip网段信息 示例值： 0.0.0.0/0
Allow	是	否	Bool	允许或者拒绝 示例值： true
Remark	否	是	String	备注信息 示例值： test remark info

AMQPQueueConsumer

AMQP队列消费者信息

被如下接口引用：DescribeAMQPQueueConsumers

名称	必选	允许NULL	类型	描述
ClientAddr	是	否	String	消费者实例的地址和端口 示例值： 192.168.0.1:5555
ClientTag	是	否	String	消费者标签 示例值： consumer1
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 100000

RocketMQClusterInfo

RocketMQ集群基本信息

被如下接口引用：DescribeRocketMQCluster、DescribeRocketMQClusters

名称	必选	允许NULL	类型	描述
PublicEndPoint	是	否	String	公网接入地址 示例值： rocketmq.access.public.com:9867
InternalEndpoint	是	是	String	TCP内部接入地址 示例值： namespace.internal:port
HttpInternalEndpoint	是	是	String	HTTP协议内部接入地址 示例值： http://ip:port
InstanceStatus	否	是	Int64	实例状态，0表示创建中，1表示正常，2表示隔离中，3表示已销毁，4 - 异常, 5 - 发货失败，6 - 变配中，7 - 变配失败 示例值： 1
AdminSecretKey	否	是	String	Admin SK 示例值： sk4ea181d5da

名称	必选	允许NULL	类型	描述
RocketMQFlag	是	是	Bool	Rocketmq集群标识 示例值： false
PublicClbId	否	是	String	公网CLB实例ID 示例值： clbId
Remark	是	是	String	集群说明信息 示例值： info-remark
HttpPublicEndpoint	是	是	String	HTTP协议公网接入地址 示例值： http://ip:port
AdminFeatureEnabled	否	否	Bool	是否开启集群Admin能力 示例值： false
ClusterName	是	否	String	集群名称 示例值： rocket-order-instance
IsolateTime	是	是	Int64	欠费停服时间，毫秒为单位 示例值： 1729478721
ZoneIds	否	是	Array of Int64	集群节点所在的可用区，若该集群为跨可用区集群，则包含该集群节点所在的多个可用区。 示例值： [100001]
NamePattern	否	是	String	名称格式 示例值： STARAND
IsFrozen	否	否	Bool	是否已冻结 示例值： false
SupportNamespaceEndpoint	是	是	Bool	是否支持命名空间接入点 示例值： true
HttpVpcEndpoint	是	是	String	HTTP协议VPC接入地址 示例值： http://ip:port
AdminAccessKey	否	是	String	Admin AK 示例值： ak5pe5ae0cca
Vip	否	是	String	vip 示例值： vip
AutoCreateTopicEnabled	否	否	Bool	是否开启自动创建主题 示例值： false
EnableDeletionProtection	否	否	Bool	是否开启删除保护 示例值： false
Vpcs	是	是	Array of VpcConfig	VPC信息 示例值：查看
IsVip	是	是	Bool	是否为专享实例 示例值： false
Status	是	是	Int64	计费状态，1表示正常，2表示已停服，3表示已销毁 示例值： 1
AclEnabled	是	是	Bool	是否开启ACL鉴权，专享实例支持关闭 示例值： true
SupportMigration	否	是	Bool	是否支持迁移 示例值： false
ZoneId	否	是	Int64	集群所属可用区，表明集群归属的可用区 示例值： 100001
ClusterId	是	否	String	集群ID 示例值： rocketmq-4k4orqgq

名称	必选	允许NULL	类型	描述
Region	是	否	String	地域信息 示例值： 华南地区
CreateTime	是	否	Uint64	创建时间，毫秒为单位 示例值： 1729478721
VpcEndPoint	是	否	String	VPC接入地址 示例值： rocketmq.resource.vpc.com:5010
VpcId	否	是	String	所属VPC 示例值： mqvpc296

ResourceProjectInfo

新增或修改资源对应的项目，新增时 oldProjectid 为空，修改时oldProjectId 为当前资源对应的 projectId

被如下接口引用：TpoModifyResourceProject

名称	必选	允许NULL	类型	描述
ResourceId	是	否	String	资源id 示例值： pulasr-11123
OldProjectId	否	否	String	旧项目id 示例值： pr-106602ce
NewProjectId	是	否	String	新项目id 示例值： pr-106602ce7

Gateway

单元化网关

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
BusinessSystemId	否	否	String	业务系统 ID 示例值： 无
GatewayId	否	否	String	网关 ID 示例值： 无
Vip	否	否	String	网关 vip 示例值： 无
Port	否	否	String	网关端口 示例值： 无

RocketMQNamespace

RocketMQ命名空间信息

被如下接口引用：DescribeInternalRocketMQNamespaces、DescribeRocketMQNamespaces

名称	必选	允许NULL	类型	描述
Ttl	是	否	Uint64	已废弃，未消费消息的保留时间，以毫秒单位，范围60秒到15天 示例值： 100000
RetentionTime	是	否	Uint64	消息持久化后保留的时间，以毫秒单位 示例值： 10000

名称	必选	允许NULL	类型	描述
Remark	是	是	String	说明 示例值： 备注
PublicEndpoint	是	是	String	公网接入点地址 示例值： http://ip:port
VpcEndpoint	是	是	String	VPC接入点地址 示例值： http://ip:port
InternalEndpoint	是	是	String	内部接入点地址 示例值： http://ip:port
UnitType	否	否	String	单元类型 示例值： SDU
UnitStatus	否	否	String	容灾状态 示例值： NOMORL
UnitEnv	否	否	String	环境类型 示例值： prod
NamespaceId	是	否	String	命名空间名称，3-64个字符，只能包含字母、数字、“-”及“_” 示例值： prod

SDU

单元化SDU

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
ZoneId	否	否	String	可用区 示例值： 无
ZoneName	否	否	String	可用去名 示例值： 无
RemoteDisasterRecoveryCloudSpaceId	否	否	String	异地灾备云空间 id 示例值： 无
UnitEnvType	否	否	String	单元环境类型- prod，gray 示例值： 无
SDUId	否	否	String	SDU ID 示例值： 无
SDUName	否	否	String	SDU 名称 示例值： 无
LocalDisasterRecoveryUnitId	否	否	String	同城灾备单元 id 示例值： 无
RemoteDisasterRecoveryUnitId	否	否	String	异地灾备单元 id 示例值： 无
Status	否	否	String	当前容灾状态 示例值： 无

RocketMQMigrationSubTask

RocketMQ数据迁移子任务信息

被如下接口引用：DescribeRocketMQMigrationTaskDetail

名称	必选	允许NULL	类型	描述
Name	是	否	String	主题或消费组名称 示例值： xxxx
Code	是	否	Int64	执行结果码： 0，成功 1，已存在，跳过 2，额度已满 3，名称不符合规范 4，系统错误 5，不合法的分区数 示例值： 0
Type	是	否	Int64	元数据类型： 0，主题 1，消费组 示例值： 0
Namespace	是	否	String	命名空间 示例值： xxxx

ConfigTemplateItemOpt

模板配置项

被如下接口引用：CreateTemplateItemsOpt

名称	必选	允许NULL	类型	描述
ItemKey	是	否	String	配置属性 示例值： key
ItemValue	是	否	String	属性值 示例值： value
ItemDesc	否	否	String	配置项备注 示例值： desc
IsDynamic	否	否	Bool	是否动态配置 示例值： true

RocketMQClusterRecentStats

RocketMQ近期使用量

被如下接口引用：DescribeRocketMQCluster

名称	必选	允许NULL	类型	描述
TopicNum	是	否	Uint64	Topic数量 示例值： 20
ApiRequestNum	是	否	Uint64	API请求数 示例值： 538000
ProducedMsgNum	是	否	Uint64	消息生产数 示例值： 11000
ConsumedMsgNum	是	否	Uint64	消息消费数 示例值： 11200
AccumulativeMsgNum	是	否	Uint64	消息堆积数 示例值： 22000

RetentionPolicy

消息保留策略

被如下接口引用：CreateEnvironment、DescribeEnvironments、ModifyEnvironmentAttributes

名称	必选	允许NULL	类型	描述
TimeInMinutes	是	否	Int64	消息保留时长 示例值： 0
SizeInMB	是	否	Int64	消息保留大小 示例值： 0

ClientSubscriptionInfo

客户端订阅详情，可用于辅助判断哪些客户端订阅关系不一致

被如下接口引用：DescribeRocketMQSubscriptions、DescribeRocketMQTopic、DescribeRocketMQTopics

名称	必选	允许NULL	类型	描述
SubString	是	是	String	订阅表达式 示例值： test_tag
ExpressionType	是	是	String	订阅方式 示例值： TAG
ClientId	是	否	String	客户端ID 示例值： rocketmq-2p9vx3ax9jxg
ClientAddr	是	是	String	客户端IP 示例值： 1.1.1.12
Topic	是	是	String	订阅主题 示例值： test_topic

EditableSource

RocketMQ Router可编辑的规则源属性

被如下接口引用：UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
AccessKey	否	否	String	新的access key 示例值： -
SecretKey	否	否	String	新的secret key 示例值： -
NewFilterExpression	否	否	String	新的RocketMQ过滤规则表达式 示例值： -
NewFilterFromTimestampMs	否	否	String	新的过滤时间，毫秒级时间戳 示例值： -

FilterSubscription

过滤订阅列表

被如下接口引用：DescribeSubscriptions

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
ConsumerHasCount	否	否	Bool	是否仅展示包含真实消费者的订阅。 示例值： false
ConsumerHasBacklog	否	否	Bool	是否仅展示消息堆积的订阅。 示例值： false
ConsumerHasExpired	否	否	Bool	是否仅展示存在消息超期丢弃的订阅。 示例值： false
SubscriptionNames	否	否	Array of String	按照订阅名过滤，精确查询。 示例值： false

AMQPClusterDetail

租户AMQP集群详细信息

被如下接口引用：DescribeAMQPClusters

名称	必选	允许NULL	类型	描述
Info	是	否	AMQPClusterInfo	集群基本信息 示例值：查看
Config	是	否	AMQPClusterConfig	集群配置信息 示例值：查看

ConfigDictItemOpt

配置项字典

被如下接口引用：DescribeConfigDictOpt

名称	必选	允许NULL	类型	描述
ItemKey	是	否	String	配置项Key 示例值： clientPort
ItemDefaultValue	是	否	String	配置项默认值 示例值： 2181
ItemDesc	是	否	String	配置项描述 示例值： Item
IsDynamic	是	否	Bool	是否动态生成 示例值： false

RocketMQMigrationTask

RocketMQ迁移任务信息

被如下接口引用：DescribeRocketMQMigrationTasks

名称	必选	允许NULL	类型	描述
TopicNum	是	否	Int64	主题总数 示例值： 10
GroupNum	是	否	Int64	消费组总数 示例值： 10
CreateTime	是	否	Uint64	创建时间，单位毫秒 示例值： 1000

名称	必选	允许NULL	类型	描述
Status	是	否	Int64	任务状态： 0，迁移中 1，迁移成功 2，迁移完成，只有部分数据完成迁移 示例值： 0
TaskId	是	否	String	任务id 示例值： xxxx
Namespace	是	是	String	导入的命名空间 示例值： xxxx
ClusterId	是	否	String	集群id 示例值： rocketmq-xxx
Type	是	否	Int64	任务类型： 0，集群迁移 1，导入指定的命名空间 示例值： 0

RocketMQMessageTrack

Rocketmq消息消费track信息

被如下接口引用：DescribeRocketMQMsg

名称	必选	允许NULL	类型	描述
ConsumeStatus	是	否	String	消费状态, CONSUMED: 已消费 CONSUMED_BUT_FILTERED: 已过滤 NOT_CONSUME: 未消费 ENTER_RETRY: 进入重试队列 ENTER_DLQ: 进入死信队列 UNKNOWN: 查询不到消费状态 示例值： CONSUMED
TrackType	是	否	String	消息track类型 示例值： CONSUMED_BUT_FILTERED
ExceptionDesc	是	是	String	异常信息 示例值： NO Exception
TraceConsumeSource	否	否	String	消费状态来源，枚举值如下： - DIFF_OFFSET：通过服务端offset计算 - TRACE_REPORT：通过上报的轨迹判断 示例值： TRACE_REPORT
Group	是	否	String	消费者组 示例值： test_group

RouterMessageServiceSource

rocketmq router mns 源信息

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
SourceConsumeConcurrentThreadCount	否	是	Int64	mns消费并发数 示例值： 1

名称	必选	允许NULL	类型	描述
FilterFromTimestampMs	否	否	Int64	过滤时间戳，毫秒级时间戳 示例值： 1694055357311
AccessKey	是	是	String	access key 示例值： -
EnableDecodeBody	否	是	String	是否自动解码mns body。 ON：自动解码，复制后的消息内容为解码后的明文。 OFF：不用解码，保持Mns消息体原始状态 示例值： OFF
Endpoint	是	是	String	mns接入点。 需要和router内网联通，一般是公网接入点 示例值： http://1111111.mns.cn-shenzhen.aliyuncs.com
Queue	是	是	String	mns queue名字。 可以在https://mns.console.aliyun.com/region/cn-shenzhen/queues 看到 示例值： -
SecretKey	是	是	String	secret key 示例值： -

RocketMQRouterRuleInfoItem

RocketMQ Router输出的规则信息

被如下接口引用：DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
TenMQTTTarget	否	是	RouterTencentMQTTTarget	腾讯云MQTT目标集群信息 示例值：查看
RemarkName	否	是	String	规则备注 示例值： -
SourceType	否	是	String	源类型。 OPEN_SOURCE_ROCKETMQ：开源rocketmq ALI_ROCKETMQ：阿里云rocketmq TENCENT_ROCKETMQ：腾讯云rocketmq ALI_MNS：阿里云mns 示例值： TENCENT_ROCKETMQ
TenRocketMQSource	否	是	RouterTencentRocketMQSource	腾讯云rocketmq源信息 示例值：查看
Status	否	是	String	规则状态。 UNKNOWN：未知 START_FAIL：启动/重启失败 JUST_BE_CREATED：已创建 JUST_BE_RE_CREATING：正在重建 PAUSED：已暂停 RUNNING：运行中 SHUTDOWN：已关闭 示例值： RUNNING
CreateTime	否	是	Int64	创建时间, 毫秒级时间戳 示例值： 1694056404492
AliRocketMQTarget	否	是	RouterRocketMQTarget	阿里云rocketmq目标信息 示例值：查看
AliMessageServiceSource	否	是	RouterMessageServiceSource	阿里云mns源信息 示例值：查看

名称	必选	允许NULL	类型	描述
TenRocketMQTarget	否	是	RouterTencentRocketMQTarget	腾讯云rocketmq目标信息 示例值：查看
AliasName	否	是	String	别名 示例值： -
RuleId	否	是	String	规则id 示例值： router-xxxx
AliMessageServiceTarget	否	是	RouterMessageServiceTarget	阿里云mns目标信息 示例值：查看
ApacheRocketMQTarget	否	是	RouterRocketMQTarget	开源rocketmq目标信息 示例值：查看
TaskStatusReason	否	是	String	任务状态产生原因. UNKNOWN: 未知 CleanTaskBySystem: 自动被系统清理 CleanTaskByRouterOffline: 自动被系统清理 CreateTask: 创建任务失败 RestartTask : 重启任务失败 CheckBeforeStartTask: 启动任务前参数检查失败 DeleteByUser: 用户删除了任务 PausedByUser：用户暂停了任务 ParamCheckFail：启动/重启时参数检查失败 ProducerFail: 生产失败 ProducerError: 生产异常 示例值： PausedByUser
UpdateTime	否	是	Int64	更新时间，毫秒级时间戳 示例值： 1694056404492
TenMQTTSource	否	是	RouterTencentMQTTSource	腾讯云MQTT源集群信息 示例值：查看
TargetType	否	是	String	目标类型，和 SourceType 字段一样 示例值： TENCENT_ROCKETMQ
AliRocketMQSource	否	是	RouterRocketMQSource	阿里云rocketmq源信息 示例值：查看
ApacheRocketMQSource	否	是	RouterRocketMQSource	开源rocketmq源信息 示例值：查看
SyncType	否	是	String	数据同步类型。 Topic: 按topic同步 示例值： Topic

EditableTarget

RocketMQ Router可编辑的目标信息

被如下接口引用：UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
AccessKey	否	否	String	新的access key 示例值： -
SecretKey	否	否	String	新的secret key 示例值： -

RocketMQConsumerClient

消费者客户端信息

被如下接口引用：DescribeRocketMQConsumerClients

名称	必选	允许NULL	类型	描述
ClientId	是	是	String	客户端ID 示例值： 127.0.0.1@25760#13749780392400
ClientAddr	是	是	String	客户端地址 示例值： 127.0.0.1:6133

AMQPRouteRelation

AMQP路由关系

被如下接口引用：DescribeAMQPRouteRelations

名称	必选	允许NULL	类型	描述
RouteRelationId	是	否	String	路由关系ID 示例值： test
SourceExchange	是	否	String	源Exchange 示例值： xxx
DestType	是	否	String	目标类型:Queue
DestValue	是	否	String	目标值 示例值： queue1
RoutingKey	是	否	String	绑定key 示例值： xxxc
SourceExchangeType	是	否	String	源路由类型:Direct
CreateTime	是	否	Uint64	创建时间，以毫秒为单位 示例值： 10000
UpdateTime	是	否	Uint64	修改时间，以毫秒为单位 示例值： 修改时间
Remark	是	是	String	说明信息 示例值： 备注

CloudSpace

单元化云空间

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
CloudSpaceName	否	否	String	云空间名称 示例值： 无
CloudId	否	否	String	TCS ID 示例值： 无
CloudName	否	否	String	TCS 名称 示例值： 无
RegionId	否	否	String	地域 示例值： 无
GDUs	否	否	Array of GDU	GDU 示例值：查看
Gateways	否	否	Array of Gateway	单元化网关 示例值：查看

名称	必选	允许NULL	类型	描述
Extensions	否	否	Array of Extension	组件扩展信息 示例值：查看
CustomerIdMappingServiceURL	否	否	String	客户号映射服务地址 示例值： 无
CloudSpaceId	否	否	String	云空间 ID 示例值： 无
SDUs	否	否	Array of SDU	SDU 示例值：查看
SDUTrafficRules	否	否	Array of SDUTrafficRule	流量路径 示例值：查看
RegionName	否	否	String	地域名 示例值： 无
UnbindGatewayIds	否	否	Array of String	待解绑的网关 Id， rocketmq目前没有使用这个字段 示例值： 无

EnvironmentRoleSet

角色批量绑定命名空间信息

被如下接口引用：CreateRole、ModifyRole

名称	必选	允许NULL	类型	描述
EnvironmentId	是	否	String	命名空间Id 示例值： devNs
Permissions	是	否	Array of String	权限枚举值 示例值： ["produce"]
ClusterId	否	是	String	绑定的集群Id 示例值： pulsar-jmw4o94a3z82
RoleName	否	是	String	角色名字 示例值： role-a

InternalRocketMQInstance

RocketMQ实例信息内部展示

被如下接口引用：DescribeInternalRocketMQInstances

名称	必选	允许NULL	类型	描述
AdminFeatureEnabled	否	否	Bool	是否开启Admin功能 示例值： true
AppId	是	否	Uint64	实例所属用户APPID 示例值： 100000
Uin	是	否	String	实例所属用户UIN 示例值： 12343444
RateLimit	是	否	Uint64	限流值 示例值： 4000
MaxRetention	是	否	Int64	最大可调整消息保留时间，小时为单位 示例值： 168

名称	必选	允许NULL	类型	描述
ClusterId	是	否	String	实例所属物理集群 示例值： rocketmq-n5degomgrwja
MaxQueuesPerTopic	是	否	Uint64	单节点主题最大可调整队列数 示例值： 16
IsVIP	是	否	Bool	是否专享实例 示例值： true
MinRetention	是	否	Int64	最小可调整消息保留时间，小时为单位 示例值： 24
AclEnabled	是	否	Bool	是否开启ACL鉴权 示例值： true
InstanceId	是	否	String	实例ID 示例值： rocketmq-n5degomgrwja
PayStatus	是	否	String	实例付费状态，OK正常，ISOLATED隔离中，DESTROYED已销毁 示例值： OK
MaxNamespaces	是	否	Uint64	最大可创建命名空间数量 示例值： 1
MaxGroups	是	否	Uint64	最大可创建消费组数量 示例值： 100
Retention	是	否	Int64	消息保留时间，小时为单位 示例值： 72
AutoCreateTopicEnabled	否	否	Bool	是否开启自动创建Topic 示例值： true
Name	是	否	String	实例名称 示例值： test_name
MaxTopics	是	否	Uint64	最大可创建主题数量 示例值： 10

ConfigBaseInfoOpt

集群配置基础信息

被如下接口引用：CreateConfigOpt、DescribeAllVersionConfigsOpt、DescribeClusterConfigOpt

名称	必选	允许NULL	类型	描述
ClusterName	是	否	String	集群名称 示例值： tdmq_txy_gz_01
ModuleName	是	否	String	模块名称 示例值： BROKER
ConfigVersion	是	否	String	配置版本 示例值： 1
RootPath	是	是	String	程序路径 示例值： /usr
DataPath	是	是	String	数据路径 示例值： /usr
ConfigPath	是	是	String	配置路径 示例值： /usr
ConfigName	是	是	String	配置文件名 示例值： broker.conf

ConfigTypeInfoOpt

集群模块配置类型描述

被如下接口引用：DescribeClusterConfigTypeOpt

名称	必选	允许NULL	类型	描述
ComponentName	是	否	String	模块组件名称 示例值： BROKER
ConfigName	是	否	String	模块配置文件名称 示例值： broker.conf
ModuleName	是	否	String	配置模块名 示例值： BROKER

DisasterRecoveryUnit

容灾单元

被如下接口引用：NotifyDisasterRecoveryUnit

名称	必选	允许NULL	类型	描述
RemoteDisasterRecoveryUnitId	否	是	String	异地灾备单元 id 示例值： unit-01
Status	否	是	String	容灾状态NORMAL—正常(恢复)LOCAL_DISASTER_RECOVERY—同城容灾 REMOTE_DISASTER_RECOVERY—异地容灾 示例值： NORMAL
CloudSpaceId	否	是	String	云空间 ID(非必填) 示例值： cloud-01
RemoteDisasterRecoveryCloudSpaceId	否	是	String	异地灾备云空间 id 示例值： cloud-01
CloudId	否	是	String	云空间名称 示例值： clould-01
RegionId	否	是	String	地域 ID 示例值： ap-chengdu
UnitId	否	是	String	单元 ID 示例值： unit-01
LocalDisasterRecoveryUnitId	否	是	String	同城灾备单元 id 示例值： unit-01
CurrentDisasterRecoveryUnitId	否	是	String	当前容灾单元 Id 示例值： unit-01
ExcludeZones	否	是	Array of String	排除流量的可用区 ID 示例值： [az1Id,az2Id]

VpcBindRecord

vcp绑定记录

被如下接口引用：DescribeBindVpcs

名称	必选	允许NULL	类型	描述
UniqueVpcId	是	否	String	租户Vpc Id 示例值： vpc-8jjiuayds

名称	必选	允许NULL	类型	描述
UniqueSubnetId	是	否	String	租户Vpc子网Id 示例值： subnet-1i12ofyz
RouterId	是	否	String	路由Id 示例值： 124323
Ip	是	否	String	Vpc的Id 示例值： 10.0.0.120
Port	是	否	Uint64	Vpc的Port 示例值： 8080
Remark	是	是	String	说明，128个字符以内 示例值： devRemark

Extension

单元化组件扩展信息

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
Data	否	否	String	扩展数据 示例值： 无
Service	否	否	String	组件名称，例如 tsf 示例值： 无

RouterTencentRocketMQTarget

rocketmq router腾讯云目标

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
ClusterId	是	是	String	rocketmq实例id 示例值： rocketmq-xxxxx
TargetRegion	否	是	String	目标region 示例值： ap-beijing
Namespace	否	是	String	4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST那部分. 比如：MQ_INST_rocket2mq3ka57d53rnn5_tiger_namespace 示例值： -
Topic	是	是	String	topic名字 示例值： -
TargetProduceConcurrentThreadCount	否	是	Int64	消费者并发数 示例值： 1

RocketMQClusterDetail

租户RocketMQ集群详细信息

被如下接口引用：DescribeRocketMQClusters

名称	必选	允许NULL	类型	描述
Status	是	是	Int64	集群状态，0:创建中，1:正常，2:销毁中，3:已删除，4: 隔离中，5:创建失败，6: 删除失败

名称	必选	允许NULL	类型	描述
				示例值： 0
Info	是	否	RocketMQClusterInfo	集群基本信息 示例值：查看
Config	是	否	RocketMQClusterConfig	集群配置信息 示例值：查看
Tags	是	是	Array of Tag	标签 示例值：查看

SendConfigInfoOpt

下发配置结果

被如下接口引用：SendAllModulesConfigOpt、SendConfigOpt

名称	必选	允许NULL	类型	描述
NodeIp	是	否	String	节点ip地址 示例值： 192.168.1.1
ClusterName	是	否	String	集群名 示例值： test-cluster
ModuleName	是	是	String	模块名 示例值： BROKER
ConfigVersion	是	是	String	配置版本 示例值： 1
Status	是	否	Bool	下发状态 示例值： true

Cluster

集群信息集合

被如下接口引用：DescribeClusterDetail、DescribeClusters

名称	必选	允许NULL	类型	描述
ClusterId	是	否	String	集群Id。 示例值： 1000
ClusterName	是	否	String	集群名称。 示例值： pulsar-47emajijmazk
Remark	是	否	String	说明信息。 示例值： devInfo
EndPointNum	是	否	Int64	接入点数量 示例值： 0
CreateTime	是	否	String	创建时间 示例值： 2021-04-21 19:00:00
Healthy	是	否	Int64	集群是否健康，1表示健康，0表示异常 示例值： 1
HealthyInfo	是	是	String	集群健康信息 示例值： healthy
Status	是	否	Int64	集群状态，0:创建中，1:正常，2:删除中，3:已删除，5:创建失败，6: 删除失败 示例值： 1

名称	必选	允许NULL	类型	描述
MaxNamespaceNum	是	否	Int64	最大命名空间数量 示例值： 100
MaxTopicNum	是	否	Int64	最大Topic数量 示例值： 8000
MaxQps	是	否	Int64	最大QPS 示例值： 1000
MessageRetentionTime	是	否	Int64	消息保留时间 示例值： 100000000
MaxStorageCapacity	是	否	Int64	最大存储容量 示例值： 1024
Version	是	是	String	集群版本 示例值： 2.5.1
PublicEndPoint	是	是	String	公网访问接入点 示例值： http://pulsar-45epqe3j7jvz.sap-8iutas0k.tdmq.ap-gz.qcloud.xxx.com:8080
VpcEndPoint	是	是	String	VPC访问接入点 示例值： http://pulsar-45epqe3j7jvz.sap-8iutas0k.tdmq.ap-gz.qcloud.xxx.com:8080
NamespaceNum	是	是	Int64	命名空间数量 示例值： 2
UsedStorageBudget	是	是	Int64	已使用存储限制，MB为单位 示例值： 0
ProjectId	否	是	String	资源对应项目id 示例值： 251198839
ProjectName	否	是	String	资源对应项目名称 示例值： product01
PayMode	是	否	Int64	计费模型0:按量计费，1：包年包月 示例值： 0
MaxPublishRateInMessages	否	是	Int64	最大生产消息速率，以条数为单位 注意：此字段可能返回 null，表示取不到有效值。 示例值： 1
PclusterName	否	是	String	集群名字 示例值： test-Id
PublicAccessEnabled	否	是	Bool	是否开启公网访问，不填时默认开启 注意：此字段可能返回 null，表示取不到有效值。 示例值： true
MaxDispatchRateInBytes	否	是	Int64	命名空间最大消费带宽，byte为单位 注意：此字段可能返回 null，表示取不到有效值。 示例值： 1
Tags	否	是	Array of Tag	标签 示例值：查看
TopicNum	否	否	Int64	topic数量 示例值： 1
InternalEndPoint	否	是	String	内部接入点 示例值： pulsar://1.1.1.1:6650
MaxDispatchRateInMessages	否	是	Int64	最大推送消息速率，以条数为单位 注意：此字段可能返回 null，表示取不到有效值。 示例值： 1
InternalPulsarEndPoint	否	是	String	内部pulsar-sdk接入地址 示例值： pulsar://tdmq.ap-gz.xxx.com:6650
MaxPublishRateInBytes	否	是	Int64	最大生产消息速率，以字节为单位 注意：此字段可能返回 null，表示取不到有效值。 示例值： 1

名称	必选	允许NULL	类型	描述
MaxMessageDelayInSeconds	否	是	Int64	最长消息延时，以秒为单位 注意：此字段可能返回 null，表示取不到有效值。 示例值： 1
InternalHttpEndPoint	否	是	String	内部http接入地址 示例值： http://tdmq.ap-gz.xxx.com:8080

PartitionsTopic

分区topic

被如下接口引用：DescribeTopics

名称	必选	允许NULL	类型	描述
AverageMsgSize	是	是	String	最后一次间隔内发布消息的平均byte大小。 示例值： 10
ConsumerCount	是	是	String	消费者数量。 示例值： 10
LastConfirmedEntry	是	是	String	被记录下来的消息总数。 示例值： 10
LastLedgerCreatedTimestamp	是	是	String	最后一个ledger创建的时间。 示例值： 2024-10-08 08:04:13,809
MsgRateIn	是	是	String	本地和复制的发布者每秒发布消息的速率。 示例值： 10
MsgRateOut	是	是	String	本地和复制的消费者每秒分发消息的数量之和。 示例值： 10
MsgThroughputIn	是	是	String	本地和复制的发布者每秒发布消息的byte。 示例值： 10
MsgThroughputOut	是	是	String	本地和复制的消费者每秒分发消息的byte。 示例值： 10
NumberOfEntries	是	是	String	被记录下来的消息总数。 示例值： 10
Partitions	是	是	Int64	子分区id。 示例值： 2
ProducerCount	是	是	String	生产者数量。 示例值： 2
TotalSize	是	是	String	以byte计算的所有消息存储总量。 示例值： 2
TopicType	是	是	Uint64	topic类型描述。 示例值： 0

RocketMQConsumerConnection

在线消费者情况

被如下接口引用：DescribeRocketMQConsumerConnections

名称	必选	允许NULL	类型	描述
Language	是	否	String	消费者应用的语言版本 示例值： JAVA

名称	必选	允许NULL	类型	描述
Accumulative	是	否	Int64	消息堆积量 示例值： 100
Version	是	否	String	消费端版本 示例值： V4.9.7
ClientId	是	否	String	消费者实例ID 示例值： rocketmq-2p9vx3ax9jxg
ClientAddr	是	否	String	消费者实例的地址和端口 示例值： 192.168.0.1:5555

Consumer

消费者

被如下接口引用：DescribeSubscriptions

名称	必选	允许NULL	类型	描述
ConnectedSince	是	是	String	消费者开始连接的时间。 示例值： 2024-04-11T23:03:47.328+08:00
ConsumerAddr	是	是	String	消费者地址。 示例值： 30.11.19.46:40940
ConsumerName	是	是	String	消费者名称。 示例值： devName1
ClientVersion	是	是	String	消费者版本。 示例值： 2.7.1
Partition	否	是	Int64	Partition数量 示例值： 0

Role

角色实例

被如下接口引用：DescribeRocketMQRoles、DescribeRoles

名称	必选	允许NULL	类型	描述
Secretkey	否	是	String	secret key 示例值： Zl34bUZYqqsdsdfYaxlrZ
PermType	否	否	String	授权类型（Cluster：集群；TopicAndGroup：主题或消费组） 示例值： Cluster
RoleName	是	否	String	角色名称。 示例值： test_role_name
Token	是	否	String	角色token值。 示例值： eyJrZXlJZCI6InN1bmdvxxxxx0X3JvbGVfMyJ9.dbHR8m6gc4L4vZUrodhW_O9bDulZQ6lraNswNLtcUcY
Remark	是	否	String	备注说明。 示例值： 测试账号
CreateTime	是	否	Datetime	创建时间。 示例值： 2020-09-22 00:00:00
UpdateTime	是	否	Datetime	更新时间。 示例值： 2020-09-22 00:00:00

Subscription

订阅者

被如下接口引用：DescribeSubscriptions

名称	必选	允许NULL	类型	描述
TopicName	是	否	String	主题名称。 示例值： devTopic
EnvironmentId	是	否	String	环境（命名空间）名称。 示例值： devNs
ConnectedSince	是	是	String	消费者开始连接的时间。 示例值： 2024-09-12T16:01:59.677+08:00
ConsumerAddr	是	是	String	消费者地址。 示例值： 127.0.0.1
ConsumerCount	是	是	String	消费者数量。 示例值： 11
ConsumerName	是	是	String	消费者名称。 示例值： devConsumerName
MsgBacklog	是	是	String	堆积的消息数量。 示例值： 1
MsgRateExpired	是	是	String	于TTL，此订阅下没有被发送而是被丢弃的比例。 示例值： 1
MsgRateOut	是	是	String	消费者每秒分发消息的数量之和。 示例值： 1
MsgThroughputOut	是	是	String	消费者每秒消息的byte。 示例值： 1
SubscriptionName	是	是	String	订阅名称。 示例值： devSubscriptionName
ConsumerSets	是	是	Array of Consumer	消费者集合。 示例值：查看
IsOnline	是	是	Bool	是否在线。 示例值： true
ConsumersScheduleSets	是	是	Array of ConsumersSchedule	消费进度集合。 示例值：查看
Remark	是	是	String	备注。 示例值： devRemark
CreateTime	是	是	Datetime	创建时间。 示例值： 2023-10-05 14:30:00
UpdateTime	是	是	Datetime	最近修改时间。 示例值： 2023-12-05 14:30:00
BlockedSubscriptionOnUnackedMsgs	否	是	Bool	是否由于未 ack 数到达上限而被 block 注意：此字段可能返回 null，表示取不到有效值。 示例值： true
SubType	否	是	String	订阅类型，Exclusive，Shared，Failover， Key_Shared，空或NULL表示未知， 注意：此字段可能返回 null，表示取不到有效值。 示例值： Shared
MaxUnackedMsgNum	否	是	Int64	未 ack 消息数上限 注意：此字段可能返回 null，表示取不到有效值。 示例值： 5000

GDU

单元化GDU

被如下接口引用：NotifyDeploymentUnitArchChanged

名称	必选	允许NULL	类型	描述
Status	否	否	String	当前容灾状态 示例值： 无
RemoteDisasterRecoveryCloudSpaceId	否	否	String	异地灾备云空间 id 示例值： 无
GDUId	否	否	String	GDU ID 示例值： 无
ZoneId	否	否	String	可用区 示例值： 无
LocalDisasterRecoveryUnitId	否	否	String	同城灾备单元 id 示例值： 无
UnitEnvType	否	否	String	单元环境类型- prod，gray 示例值： 无
ExcludeZones	否	否	Array of String	排除的可用区Id信息, V4新增 示例值： 无
GDUName	否	否	String	GDU 名称 示例值： 无
ZoneName	否	否	String	可用区名 示例值： 无
RemoteDisasterRecoveryUnitId	否	否	String	异地灾备单元 id 示例值： 无

RouterTencentRocketMQSource

rocketmq router腾讯云源

被如下接口引用：CreateRocketMQRouterRule、DeleteRocketMQRouterRule、DescribeRocketMQRouterRuleDetail、DescribeRocketMQRouterRules、UpdateRocketMQRouterRule

名称	必选	允许NULL	类型	描述
SourceConsumeConcurrentThreadCount	否	是	Int64	消费者并发数 示例值： 1
ClusterId	是	是	String	rocketmq实例id 示例值： rocketmq-xxxxx
SourceRegion	否	是	String	源地域 示例值： ap-beijing
FilterExpression	是	是	String	过滤表达式, 参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java 示例值： -
FilterFromTimestampMs	是	是	Int64	过滤时间，毫秒时间戳 示例值： 1691983398481
FilterType	是	是	String	过滤类型，参考：https://github.com/apache/rocketmq/blob/develop/common/src/main/java/org/apache/rocketmq/common/filter/ExpressionType.java 示例值： TAG

名称	必选	允许NULL	类型	描述
Namespace	否	是	String	4.X的命名空间，没有使用则留空，5.X也留空。 naemspace 需要包含MQ_INST 那部分. 比如：MQ_INST_rocketmqka57d53rnn5_tiger_namespace 示例值： -
Topic	是	是	String	topic名字 示例值： -

错误码

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。 请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。 签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
InternalError.IllegalMessage	
InternalError.VpcService	VPC服务异常。
FailedOperation.CreateNamespace	创建命名空间失败。
MissingParameter	缺少参数错误。

错误码	说明
FailedOperation.DeleteEnvironments	环境删除失败。
ResourceInsufficient	
ResourceInUse.EnvironmentRole	
FailedOperation.DeleteNamespace	删除命名空间失败。
ResourceNotFound.BindVpc	vpc绑定关系不存在。
FailedOperation.UpdateEnvironment	环境更新失败。
FailedOperation.CreateEnvironmentRole	
InvalidParameterValue.ClusterNameDuplication	与现有集群名称重复。
LimitExceeded.Topics	实例下主题数量超过限制。
ResourceInUse.Topic	重名，主题已存在。
FailedOperation.DeleteTopics	
InvalidParameter.TenantNotFound	上传的 tenant name 错误。
ResourceInUse.VpcRoute	vpc路由已存在。
ResourceNotFound.Topic	
FailedOperation.CreateSecretKey	
ResourceNotFound.Msg	找不到指定消息。
FailedOperation.CreateCluster	
FailedOperation.VpcInUse	必须先清除关联VPC路由数据才能继续操作。
FailedOperation.CreateRole	
FailedOperation.UpdateRole	
FailedOperation.UpdateEnvironmentRole	
InvalidParameterValue.InvalidParams	
ResourceInUse.Cluster	集群已存在。
ResourceNotFound.Namspace	
ResourcesSoldOut	资源售罄。

错误码	说明
ResourceInUse	
FailedOperation.CreateBindVpc	创建vpc绑定关系失败。
ResourceUnavailable.FundRequired	
ResourceNotFound	
FailedOperation.SaveSecretKey	
LimitExceeded.Clusters	
FailedOperation.GetEnvironmentAttributesFailed	获取环境属性失败。
FailedOperation.DeleteRoles	
AuthFailure.UnauthorizedOperation	
LimitExceeded	
FailedOperation.CreateTopic	
InvalidParameterValue.TopicNotFound	上传的topic name错误。
LimitExceeded.Subscriptions	实例下订阅者数量超过限制。
ResourceNotFound.Environment	
ResourceNotFound.Instance	
FailedOperation	
ResourceUnavailable.CreateFailed	
FailedOperation.CreateSubscription	创建订阅关系失败。
ResourceUnavailable	
ResourceNotFound.MsgProduceLog	消息生产日志不存在。
FailedOperation.SetTTL	
ResourceInUse.Subscription	重名，订阅关系已存在。
FailedOperation.InstanceNotReady	
ResourceInUse.Role	
InternalError.GetAttributesFailed	获取属性失败。

错误码	说明
InternalError.CloudAPIError	云API错误。
UnknownParameter	未知参数错误。
LimitExceeded.Environments	实例下环境数量超过限制。
FailedOperation.GetTopicPartitionsFailed	获取主题分区数失败。
FailedOperation.DeleteSubscriptions	删除订阅关系失败。
FailedOperation.NamespaceInUse	必须先清除关联命名空间才能继续操作。
InvalidParameterValue.TTL	无效的消息TTL值。
OperationDenied.DefaultEnvironment	默认环境不允许操作。
InvalidParameterValue.NeedMoreParams	必要参数没有传递。
FailedOperation.CreateProducerError	创建producer出错。
InvalidParameter	
InternalError.Retry	重试可以成功。
FailedOperation.ResetMsgSubOffsetByTimestampFailed	消息回溯设置失败。
FailedOperation.DescribeSubscription	查询订阅数据失败。
ResourceInUse.SupportRoute	支撑路由已存在。
InvalidParameterValue	参数取值错误。
ResourceInUse.BindVpc	vpc绑定关系已存在。
FailedOperation.UpdateTopic	主题更新失败。
FailedOperation.DeleteBindVpc	删除vpc绑定关系失败。
FailedOperation.CallTrade	
FailedOperation.ReceiveError	接收消息出错。
FailedOperation.ReceiveTimeout	接收消息超时，请重试。
UnauthorizedOperation	未授权操作。
InternalError	
ResourceNotFound.Role	

错误码	说明
FailedOperation.DescribeProducers	查询生产者数据失败。
InvalidParameter.Partition	
InternalError.FailedOperation	
FailedOperation.DeleteCluster	
InternalError.SystemError	
FailedOperation.TopicInUse	必须先清除关联主题数据才能继续操作。
InternalError.BrokerService	Broker服务异常。
UnsupportedOperation	
FailedOperation.RouterRule	
FailedOperation.CreatePulsarClientError	创建TDMQ client的出错。
InvalidParameterValue.AtLeastOne	
ResourceNotFound.BrokerCluster	
ResourceNotFound.EnvironmentRole	
OperationDenied.ConsumerRunning	订阅仍在消费中。
ResourceNotFound.Namespace	
LimitExceeded.Namespaces	实例下命名空间数量超过限额。
InvalidParameter.TokenNotFound	没有获取到正确的 token。
ResourceNotFound.Cluster	
FailedOperation.RoleInUse	
ResourceNotFound.MigrationTask	
FailedOperation.DeleteEnvironmentRoles	
ResourceNotFound.Subscription	订阅关系不存在。
ResourceUnavailable.SystemUpgrade	系统升级。
FailedOperation.CreateEnvironment	环境创建失败。
UnsupportedOperation.InstanceDowngrade	

错误码	说明
MissingParameter.NeedMoreParams	
ResourceInUse.Namespace	重名，命名空间已存在。
FailedOperation.SendMsgFailed	发送消息失败。

版本（2021-06-21）
API 概览

API版本

V3

调用方式
接口签名v1

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（ Signature ）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. 生成签名串

有了安全凭证SecretId 和 SecretKey后，就可以生成签名串了。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下:

参数名称	中文	参数值

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 php 中的 ksort 函数。上述示例参数的排序结果如下:

```
{
    'Action' : 'DescribeInstances',
    'InstanceIds.0' : 'ins-09dx96dg',
    'Limit' : 20,
    'Nonce' : 11886,
    'Offset' : 0,
    'Region' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    'Timestamp' : 1465185768,
    'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。
将把上一步排序好的请求参数格式化成“参数名称”=“参数值”的形式，如对 Action 参数，其参数名称为 "Action" ，参数值为 "DescribeInstances" ，因此格式化后就为 Action=DescribeInstances 。
注意：“参数值”为原始值而非url编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为:

```
Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。
签名原文字符串由以下几个参数构成:

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机:查看实例列表(DescribeInstances)的请求域名为：cvm.finance.cloud.tencent.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 / 。
4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 +请求路径 + ? + 请求字符串

示例的拼接结果为：

```
GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。
首先使用 HMAC-SHA1 算法对上一步中获得的签名原文字符串进行签名，然后将生成的签名串使用 Base64 进行编码，即可获得最终的签名串。

具体代码如下，以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

使用其它程序设计语言开发时，可用上面示例中的原文进行签名验证，得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%3d，它将用于生成最终的请求 URL。

注意：如果用户的请求方法是 GET，或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded，则发送请求时所有请求参数的值均需要做 URL 编码，参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的 http 库会自动为所有参数进行 urlencode，在这种情况下，就不需要对签名串进行 URL 编码了，否则两次 URL 编码会导致签名失败。

注意：其他参数值也需要进行编码，编码采用 RFC 3986。使用 %XY 对特殊字符例如汉字进行百分比编码，其中“X”和“Y”为十六进制字符（0-9 和大写字母 A-F），使用小写将引发错误。

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的TCloudFinanceZone SDK 3.0 ，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 SDK 中心。当前支持的编程语言有：

- Python
- Java

- PHP
- Go
- Node

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK 。

最终输出的 url 可能为： https://cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url ，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp 。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API ，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class CloudAPIDemo {
    private final static String CHARSET = "UTF-8";

    public static String sign(String s, String key, String method) throws Exception {
        Mac mac = Mac.getInstance(method);
        SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
        mac.init(secretKeySpec);
        byte[] hash = mac.doFinal(s.getBytes(CHARSET));
        return DatatypeConverter.printBase64Binary(hash);
    }

    public static String getStringToSign(TreeMap<String, Object> params) {
        StringBuilder s2s = new StringBuilder("GETcvm.finance.cloud.tencent.com/?");
```

```
        // 签名时要求对参数进行字典排序，此处用TreeMap保证顺序
        for (String k : params.keySet()) {
            s2s.append(k).append("=").append(params.get(k).toString()).append("&");
        }
        return s2s.toString().substring(0, s2s.length() - 1);
    }

    public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException
{
        StringBuilder url = new StringBuilder("https://cvm.finance.cloud.tencent.com/?");
        // 实际请求的url中对参数顺序没有要求
        for (String k : params.keySet()) {
            // 需要对请求串进行urlencode，由于key都是英文字母，故此处仅对其value进行urlencode
            url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).app
end("&");
        }
        return url.toString().substring(0, url.length() - 1);
    }

    public static void main(String[] args) throws Exception {
        TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
        // 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 公共参数
        params.put("Action", "DescribeInstances"); // 公共参数
        params.put("Version", "2017-03-12"); // 公共参数
        params.put("Region", "shjr"); // 公共参数
        params.put("Limit", 20); // 业务参数
        params.put("Offset", 0); // 业务参数
        params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

注意：如果是在 Python 2 环境中运行，需要先安装 requests 依赖包： pip install requests 。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/?"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        'SecretId' : secret_id,
        'Timestamp' : 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

接口签名v3

TCloudFinanceZone API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。
签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往云API密钥页面申请，否则无法调用云API接口。

1. 申请安全凭证

在第一次使用云API之前，请前往云API密钥页面申请安全凭证。
安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

1. 登录TCloudFinanceZone管理中心控制台。
2. 前往云API密钥的控制台页面
3. 在云API密钥页面，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：对于GET方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于POST方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
    HTTPRequestMethod + '\n' +
    CanonicalURI + '\n' +
    CanonicalQueryString + '\n' +
    CanonicalHeaders + '\n' +
    SignedHeaders + '\n' +
    HashedRequestPayload
```

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- CanonicalURI：URI 参数，API 3.0 固定为正斜杠（/）；
- CanonicalQueryString：发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- SignedHeaders：参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。拼接规则：1）头部 key 统一转成小写；2）多个头部 key（小写）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
    Algorithm + \n +
```

```
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

- Algorithm：签名算法，目前固定为 TC3-HMAC-SHA256；
- RequestTimestamp：请求时间戳，即请求头部的 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- CredentialScope：凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

> 1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
> 2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败（返回签名过期错误）。

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

```
Authorization =
    Algorithm + ' ' +
    'Credential=' + SecretId + '/' + CredentialScope + ', ' +
    'SignedHeaders=' + SignedHeaders + ', '
    'Signature=' + Signature
```

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
        SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
        // 注意时区，否则容易出错
        sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
        String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

        // ************* 步骤 1：拼接规范请求串 *************
        String httpRequestMethod = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\n"
                + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
        System.out.println(canonicalRequest);

        // ************* 步骤 2：拼接待签名字符串 *************
        String credentialScope = date + "/" + service + "/" + "tc3_request";
        String hashedCanonicalRequest = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        System.out.println(signature);

        // ************* 步骤 4：拼接 Authorization *************
        String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
                + "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
        System.out.println(authorization);

        TreeMap<String, String> headers = new TreeMap<String, String>();
        headers.put("Authorization", authorization);
        headers.put("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

```
payload = ""
if http_request_method == "POST":
    canonical_querystring = ""
    ct = "json"
    payload = json.dumps(params)
canonical_headers = "content-type:application/%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
                     canonical_uri + "\n" +
                     canonical_querystring + "\n" +
                     canonical_headers + "\n" +
                     signed_headers + "\n" +
                     hashed_request_payload)
print(canonical_request)

# ************* 步骤 2：拼接待签名字符串 *************
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
                  str(timestamp) + "\n" +
                  credential_scope + "\n" +
                  hashed_canonical_request)
print(string_to_sign)

# ************* 步骤 3：计算签名 *************
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ************* 步骤 4：拼接 Authorization *************
authorization = (algorithm + " " +
                 "Credential=" + secret_id + "/" + credential_scope + ", " +
                 "SignedHeaders=" + signed_headers + ", " +
                 "Signature=" + signature)
print(authorization)

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

TCloudFinanceZone API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

- application/json（推荐），必须使用 TC3-HMAC-SHA256 签名方法。
- application/x-www-form-urlencoded，必须使用 HmacSHA1 或 HmacSHA256 签名方法。
- multipart/form-data（仅部分接口支持），必须使用 TC3-HMAC-SHA256 签名方法。

GET 请求的请求包大小不得超过 32 KB。POST 请求使用签名方法为 HmacSHA1、HmacSHA256 时不得超过 1 MB。POST 请求使用签名方法为 TC3-HMAC-SHA256 时支持 10 MB。

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
    "Response": {
        "TotalCount": 0,
        "InstanceStatusSet": [],
        "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
    }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0， InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
    "Response": {
        "Error": {
            "Code": "AuthFailure.SignatureFailure",
            "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
        },
        "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
    }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

公共参数

公共参数是用于标识用户和接口鉴权目的的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
X-TC-Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

集群相关接口
TDMQ询价接口

1. 接口描述

接口请求域名： tdmq.api3.finance.cloud.tencent.com。

TDMQ计费询价接口

默认接口请求频率限制：20次/秒。

接口更新时间：2023-04-18 16:43:30。

接口只验签名不鉴权。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见公共请求参数。

参数名称	必选	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetPrice
Version	是	否	String	公共参数，本接口取值：2021-06-21
Region	是	否	String	公共参数，地域信息可通过DescribeRegions接口查看产品支持的地域列表
GoodsNum	是	否	Uint64	产品数 示例值： Number of Products
GoodType	是	否	String	产品类型（sv_tdmq_pulsar_partition/ sv_tdmq_pulsar_message_count/ sv_tdmq_pulsar_storage） 示例值： Product Type

3. 输出参数

参数名称	类型	描述
Data	String	计费信息

参数名称	类型	描述
		示例值： data
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

该模块不涉及数据结构

错误码

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is cor
rect."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。 请在控制台检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。 签名计算错误，请对照调用方式中的接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码