

阿里云 音视频通信

API参考

文档版本：20190918

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 禁止： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明： 您也可以通过按Ctrl + A选中全部文件。
>	多级菜单递进。	设置 > 网络 > 设置网络类型
粗体	表示按键、菜单、页面名称等UI元素。	单击 确定 。
<code>courier</code> 字体	命令。	执行 <code>cd /d C:/windows</code> 命令，进入Windows系统文件夹。
##	表示参数、变量。	<code>bae log list --instanceid</code> <code>Instance_ID</code>
[]或者[a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ }或者{a b}	表示必选项，至多选择一个。	<code>swich {stand slave}</code>

目录

法律声明.....	I
通用约定.....	I
1 API概览.....	1
2 调用方式.....	2
2.1 概述.....	2
2.2 请求结构.....	2
2.3 公共参数.....	3
2.4 返回结果.....	4
2.5 签名机制.....	5
3 获取AccessKey.....	8
4 RAM资源授权.....	11
5 应用管理.....	12
5.1 DescribeApps.....	12
5.2 ModifyApp.....	14
6 频道管理.....	16
6.1 DeleteChannel.....	16
6.2 RemoveTerminals.....	17
7 数据服务.....	19
7.1 DescribeRtcDurationData.....	19
7.2 DescribeRtcUserCntData.....	21
7.3 DescribeRtcChannelCntData.....	23
7.4 DescribeRtcPeakUserCntData.....	25
7.5 DescribeRtcPeakChannelCntData.....	27
8 旁路直播.....	30
8.1 StartMPUTask.....	30
8.2 GetMPUTaskStatus.....	33
8.3 StopMPUTask.....	35
8.4 UpdateMPULayout.....	36
9 服务端错误码表.....	39

1 API概览

音视频通信提供以下相关API接口。

应用管理

API	描述
DescribeApps	调用DescribeApps查询应用列表。
ModifyApp	调用ModifyApp修改指定应用的名称。

频道管理

API	描述
DeleteChannel	调用DeleteChannel删除频道。
RemoveTerminals	调用RemoveTerminals将指定终端用户从频道踢出。

数据服务

API	描述
DescribeRtcDurationData	调用DescribeRtcDurationData获取应用在一段时间内的累计通信时长，区分音视频规格进行统计。
DescribeRtcUserCntData	调用DescribeRtcUserCntData查询应用在一段时间内的活跃用户数，即发生通信的用户终端数。
DescribeRtcChannelCntData	调用DescribeRtcChannelCntData查询应用在一段时间内的活跃频道数，即发生通信的频道数。
DescribeRtcPeakUserCntData	调用DescribeRtcPeakUserCntData查询应用在一段时间内的并发通信峰值，一组“发布-订阅”关系被标记为一次通信。
DescribeRtcPeakChannelCntData	调用DescribeRtcPeakChannelCntData查询应用在一段时间内的并发频道峰值数量。

旁路直播

API	描述
StartMPUTask	调用StartMPUTask开始任务。
GetMPUTaskStatus	调用GetMPUTaskStatus获取任务状态。
StopMPUTask	调用StopMPUTask停止任务。
UpdateMPULayout	调用UpdateMPULayout更新任务的布局。

2 调用方式

2.1 概述

音视频通信RTC API的调用是通过向服务端地址发送HTTP GET 请求，并按照接口说明在请求中加入请求参数来完成的。根据请求的处理情况，系统会返回处理结果。

音视频通信API的调用方式从以下方面为您说明：

- [#unique_19](#)
- [#unique_20](#)
- [#unique_21](#)
- [#unique_22](#)

2.2 请求结构

音视频通信RTC API的调用是通过向音视频通信服务端地址发送HTTP GET请求，并按照接口说明在请求中加入相应请求参数，调用后系统会返回处理结果。

音视频通信的API是RPC风格，您可以通过发送HTTP GET请求调用API，并支持通过HTTP或HTTPS通道进行请求通信，为了获得更高的安全性，推荐您使用HTTPS通道发送请求。请求及返回结果都使用UTF-8字符集进行编码，请求结构如下所示。

```
https://Endpoint/?Action=xx&Version=xx&Parameters
```

- Endpoint：音视频通信API的服务接入地址为rtc.aliyuncs.com。
- Action：要执行的操作，例如调用CreateChannel创建频道。
- Version：要使用的API版本，目前音视频通信的API版本是2018-01-11。
- Parameters：请求参数，每个参数之间用&分隔。请求参数由公共请求参数和API自定义参数组成。公共参数中包含API版本号、身份验证等信息，详情请参见[#unique_20](#)。

下面是一个调用CreateChannel创建频道的示例：



说明：

为了便于您的查看，本文档中的示例都做了格式化处理。

```
https://rtc.aliyuncs.com/?Action=CreateChannel  
&AppId=xxx  
&ChannelId=xxx&  
&Format=XML
```

```
&AccessKeyId=xxx
&SignatureMethod=HMAC-SHA1
&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
&Version=2018-01-11
&SignatureVersion=1.0
&Signature=xxxx
&Timestamp=2017-03-29T09%3A22%3A32Z
...
```

2.3 公共参数

本文档为您介绍了调用RTC API使用的公共参数，公共参数包含请求参数和返回参数。

公共请求参数

参数	类型	是否必选	描述
Format	String	否	返回值的类型，支持JSON与XML，默认为XML。
Version	String	是	API版本号，为日期形式：YYYY-MM-DD，本版本对应为2018-01-11。
AccessKeyId	String	是	阿里云颁发给用户的访问服务所用的密钥ID。
Signature	String	是	签名结果串。关于签名的计算方法，请参见 #unique_25 。
SignatureMethod	String	是	签名方式，目前支持HMAC-SHA1。
Timestamp	String	是	请求的时间戳。日期格式按照ISO8601标准表示，使用UTC时间，格式为：YYYY-MM-DDThh:mm:ssZ。 例如，2017-3-29T12:00:00Z为北京时间2017年3月29日的20点0分0秒。
SignatureVersion	String	是	签名算法版本，目前版本是1.0。
SignatureNonce	String	是	唯一随机数。在不同请求间使用不同的随机数值，防止网络重放攻击。 代码示例参见 SignatureNonce 。
SecurityToken	String	否	STS临时授权的Token，默认为空。 相关概念详情请参见 账号权限 、 STS临时授权访问 。

示例

```
http://rtc.aliyuncs.com/  
?Format=json  
&Version=2018-01-11  
&Signature=vpEEL0zFHfxXYZSFV0n7%2FZiFL9o%3D  
&SignatureMethod=Hmac-SHA1  
&SignatureNonce=9166ab59-f445-4005-911d-664c1570df0f  
&SignatureVersion=1.0  
&Action=ModifyApp  
&AccessKeyId=XXXXXXX  
&Timestamp=2017-03-29T09%3A22%3A32Z  
...
```

公共返回参数

API返回结果采用统一格式，返回2xx HTTP状态码代表调用成功；返回4xx或5xx HTTP状态码代表调用失败。调用成功返回的数据格式有XML和JSON两种，可以在发送请求时指定返回的数据格式，默认为XML格式。

每次接口调用，无论成功与否，系统都会返回一个唯一识别码RequestId。

· XML格式

```
<?xml version="1.0" encoding="UTF-8"?>  
  <!--结果的根结点-->  
  <接口名称+Response>  
    <!--返回请求标签-->  
    <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>  
    <!--返回结果数据-->  
  </接口名称+Response>
```

· JSON格式

```
{  
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216", /* 返回结果数  
据 */  
}
```

2.4 返回结果

本文档为您介绍调用阿里云RTC API的返回结果。返回结果主要有XML和JSON两种格式，默认为XML格式，您也可以指定请求参数Format变更返回结果的格式。



说明:

为了便于您查看，本文档返回示例进行了换行或缩进等格式处理。

正常返回示例

调用API返回2xx HTTP状态码代表调用成功。

- XML示例如下所示。

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--结果的根结点-->
<接口名称+ Response>
  <!--返回请求标签-->
  <RequestId>25818875-5F78-4A13-BEF6-D7393642CA58</RequestId>
  <!--返回结果数据-->
</接口名称+Response>
```

- JSON示例如下所示。

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216", /* 返回结果数
据 */
}
```

异常返回示例

调用API返回4xx或者5xx HTTP状态码代表调用失败，并返回RequestId、HostId、错误代码和错误信息。当您无法排查错误时，可以[提交工单](#)联系我们，并在工单中注明服务节点HostId和RequestId。

- XML示例如下所示。

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>rtc.aliyuncs.com</HostId>
  <Code>UnsupportedOperation</Code>
  <Message>The specified action is not supported.</Message>
</Error>
```

- JSON示例如下所示。

```
{
  "RequestId": "8906582E-6722-409A-A6C4-0E7863B733A5",
  "HostId": "rtc.aliyuncs.com",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

2.5 签名机制

为保证API的安全调用，在调用API时，阿里云RTC会对每个API请求通过签名（Signature）进行身份验证。无论您使用HTTP还是HTTPS协议提交请求，都需要在请求中包含签名信息。



说明：

阿里云RTC提供了多种编程语言的SDK及第三方SDK，可以省略计算签名步骤，SDK下载请单击[阿里云SDK中心](#)。

RAM授权

为了确保您的账号安全，建议您使用子账号的身份凭证调用API。如果您使用RAM账号调用API，您需要为该RAM账号创建、附加相应的授权策略。RTC中可授权的资源 and 接口列表，请参见[#unique_28](#)，获取AccessKey请参见[#unique_29](#)。

API签名

RTC服务会对每个API请求进行身份验证，签名计算过程请参见[#unique_25](#)。

RTC通过使用AccessKey ID和AccessKey Secret进行对称加密的方法来验证请求的发送者身份。AccessKey是为阿里云账号和RAM用户发布的一种身份凭证（类似于用户的登录密码），其中AccessKey ID用于标识访问者的身份，AccessKey Secret是用于加密签名字符串和服务器端验证签名字符串的密钥，必须严格保密。

以CreateChannelAPI为例，假设使用的AccessKeyId为testid，AccessKeySecret为testsecret。签名前的请求URL如下：

```
https://rtc.aliyuncs.com/?Action=CreateChannel
&AppId=xxx
&ChannelId=xxx
&Format=XML
&AccessKeyId=testid
&SignatureMethod=HMAC-SHA1
&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
&Version=2018-01-11
&SignatureVersion=1.0
```

使用testsecret&，计算得到的签名值是：

```
Y8c1u47r2gHn6scXqz92wklKws1=
```

将签名作为Signature参数加入到URL请求中，最后得到的URL为：

```
https://rtc.aliyuncs.com/?Action=CreateChannel
&AppId=xxx
&ChannelId=xxx
&Format=XML
&AccessKeyId=testid
&SignatureMethod=HMAC-SHA1
&SignatureNonce=3ee8c1b8-83d3-44af-a94f-4e0ad82fd6cf
&Version=2018-01-11
&SignatureVersion=1.0&Signature=Y8c1u47r2gHn6scXqz92wklKws1=
```

阿里云为您提供多种编程语言SDK签名示例代码。

- [PHP 签名示例](#)
- [Python签名示例](#)
- [.NET 签名示例](#)
- [Go 签名示例](#)

- [Node.js 签名示例](#)
- [C/C++ 签名示例](#)

3 获取AccessKey

您可以为阿里云主账号和子账号创建一个访问密钥（AccessKey）。在调用阿里云API时您需要使用AccessKey完成身份验证。

背景信息

AccessKey包括AccessKeyId和AccessKeySecret。

- AccessKeyId：用于标识用户。
- AccessKeySecret：用于验证用户的密钥。AccessKeySecret必须保密。



警告：

主账号Accesskey泄露会威胁您所有资源的安全。建议使用子账号（RAM用户）Accesskey进行操作，可有效降低Accesskey泄露的风险。

操作步骤

1. 以主账号登录[阿里云管理控制台](#)。
2. 将鼠标置于页面右上方的账号图标，单击accesskeys。
3. 在安全提示页面，选择获取主账号还是子账号的Accesskey。

安全提示



提示信息云账号AccessKey是您访问阿里云API的密钥，具有该账户完全的权限，请您务必妥善保管！不要通过任何方式(eg, Github)将AccessKey公开到外部渠道，以避免被他人利用而造成 [安全威胁](#)。强烈建议您遵循 [阿里云安全最佳实践](#)，使用RAM子用户AccessKey来进行API调用。

继续使用AccessKey

开始使用子用户AccessKey

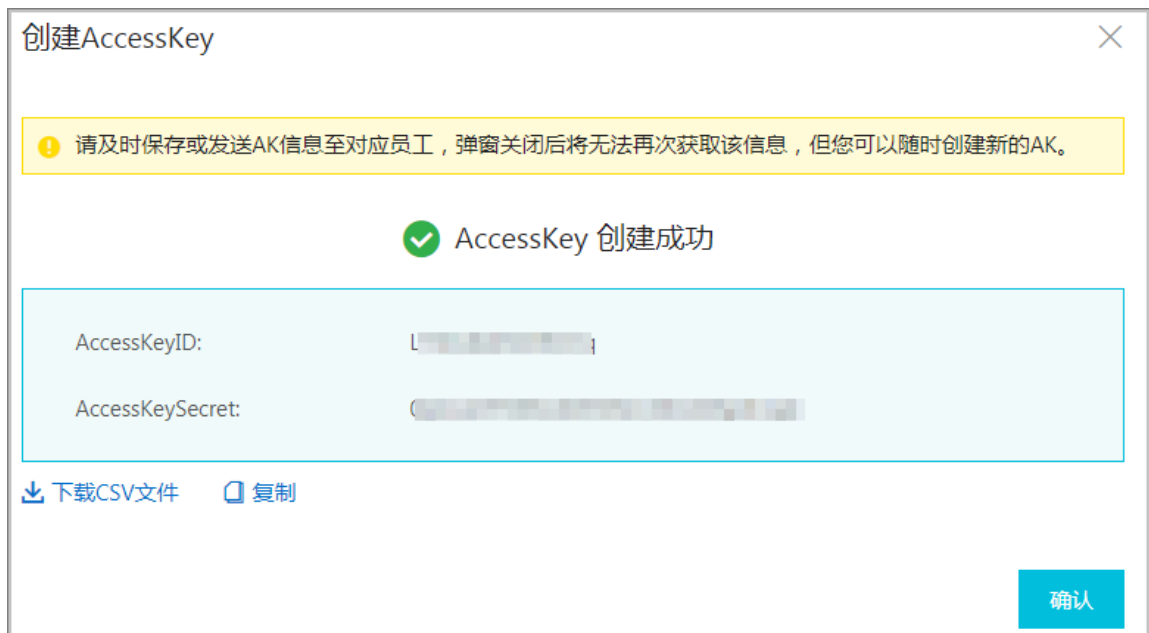
4. 获取主账号Accesskey。

- a) 单击继续使用AccessKey。
- b) 在安全信息管理页面，单击创建AccessKey。
- c) 在手机验证页面，获取验证码，完成手机验证，单击确定。
- d) 在新建用户AccessKey页面，展开AccessKey详情，查看AccessKeyId和AccessKeySecret。可以单击保存AK信息，下载AccessKey信息。



5. 您也可以获取子账号AccessKey。

- a) 单击开始使用子用户AccessKey。
- b) 如果您未创建RAM用户，在系统跳转的[RAM访问控制台](#)新建用户页面，创建RAM用户。如果是获取已有RAM用户的Accesskey，则跳过此步骤。
- c) 在左侧导航栏，选择人员管理 > 用户，搜索需要获取AccessKey的用户。
- d) 单击用户登录名称，在用户详情页认证管理页签下的用户AccessKey区域，单击创建新的AccessKey。
- e) 在手机验证页面，获取验证码，完成手机验证，单击确定。
- f) 在创建AccessKey页面，查看AccessKeyId和AccessKeySecret。您可以单击下载CSV文件，下载AccessKey信息，或者单击复制，复制AccessKey信息。



4 RAM资源授权

在使用RAM账号调用RTC API之前，需要主账号通过创建授权策略对RAM账号进行授权。在授权策略中，使用资源描述符（Alibaba Cloud Resource Name, ARN）指定授权资源。

可授权的音视频通信资源类型

在进行RAM子账号授权时，音视频通信的资源类型及描述方式如下表所示。

资源类型	授权策略中的资源描述方式	说明
service	acs:rtc::\$accountid:	授权子账户管理RT 服务。例如：变配、查询账户信息等。
appid	acs:rtc::\$accountid:app/	授权子账户管理自己的应用。例如：停用、启用、配置、查询等。

可授权的音视频通信接口

当子账号调用RTC API对主账号的资源进行访问时，RTC 后台会进行权限检查。

下表列举了RTC中可授权的API及其鉴权规则。

API	鉴权规则
DescribeApps	acs:rtc:*\$accountid:app/
StartApp	acs:rtc::\$accountid:app/\$appid
StopApp	acs:rtc::\$accountid:app/\$appid
CreateChannel	acs:rtc::\$accountid:app/\$appid
DeleteChannel	acs:rtc::\$accountid:app/\$appid
UpdateChannel	acs:rtc::\$accountid:app/\$appid
DescribeRealTimeRecordDetail	acs:rtc::\$accountid:app/\$appid
DescribeRealTimeRecordList	acs:rtc::\$accountid:app/\$appid
DescribeRecordDetail	acs:rtc::\$accountid:app/\$appid
DescribeRecordList	acs:rtc::\$accountid:app/\$appid
DescribeStatis	acs:rtc::\$accountid:app/\$appid

5 应用管理

5.1 DescribeApps

调用DescribeApps查询应用列表。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	否	DescribeApps	操作接口名，系统规定参数，取值：DescribeApps。
AppId	String	否	yourAppId	应用ID，通过控制台开通创建，长度最大8个字节，不填则返回该用户下所有应用信息。
Order	String	否	asc	排序，不填默认desc取最近记录。 取值： · asc：递增。 · desc：递减。
PageNum	Integer	否	1	不填则默认1页，取值必须大于0。
PageSize	Integer	否	2	不填则默认为10，取值必须大于0。
Status	String	否	1	状态： · 1：可用。 · 2：停用。 · 3：欠费停用。

返回数据

名称	类型	示例值	描述
AppList			信息列表。
AppId	String	yourAppId	应用ID。
AppName	String	abc	应用名称。
AppType	String	conference	应用类型。 · conference: 会议。 · universal: 通用型。
BillType	String	payByDuration	按时长计费: payByDuration。
CreateTime	String	2017-08-30 10:47:25.0	创建时间。
ServiceAreas		["CN"]	服务地区。
Status	Integer	1	状态: · 1: 可用。 · 2: 停用。 · 3: 欠费停用。
RequestId	String	6159ba01-6687-4fb2-a831-f0cd8d188648	请求ID。
TotalNum	Integer	2	返回结果数。
TotalPage	Integer	1	返回分页数。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeApps
&PageNum=1
&PageSize=2
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeAppsResponse>
  <RequestId>6159ba01-6687-4fb2-a831-f0cd8d188648</RequestId>
  <TotalNum>2</TotalNum>
  <TotalPage>1</TotalPage>
  <AppList>
    <Status>1</Status>
    <AppName>coco-conf</AppName>
    <CreateTime>2017-08-30 10:47:25.0</CreateTime>
    <AppId>af2ace82-dc55-40b1-9fa3-688787214b1d</AppId>
    <ServiceAreas>["CN"]</ServiceAreas>
    <AppType>universal</AppType>
    <BillType>payByDuration</BillType>
  </AppList>
</DescribeAppsResponse>
```

JSON 格式

```
{
  "TotalPage":1,
  "AppList":[
    {
      "ServiceAreas":["CN"],
      "Status":1,
      "CreateTime":"2017-08-30 10:47:25.0",
      "AppId":"af2ace82-dc55-40b1-9fa3-688787214b1d",
      "AppName":"coco-conf",
      "BillType":"payByDuration",
      "AppType":"universal"
    }
  ],
  "TotalNum":2,
  "RequestId":"6159ba01-6687-4fb2-a831-f0cd8d188648"
}
```

错误码

访问[错误中心](#)查看更多错误码。

5.2 ModifyApp

调用ModifyApp修改指定应用的名称。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
AppId	String	是	yourAppId	应用ID，通过控制台开通创建。

名称	类型	是否必选	示例值	描述
AppName	String	是	yourAppName	应用名称。
Action	String	否	ModifyApp	操作接口名，系统规定参数，取值： ModifyApp 。

返回数据

名称	类型	示例值	描述
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=ModifyApp
&AppName=coco-conf
&AppId=xxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<ModifyAppResesponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
</ModifyAppResesponse>
```

JSON 格式

```
{
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6 频道管理

6.1 DeleteChannel

调用DeleteChannel删除频道。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DeleteChannel	操作接口名，系统规定参数，取值：DeleteChannel。
AppId	String	是	yourAppId	应用ID，通过控制台开通创建。
ChannelId	String	是	yourChannelId	频道ID，加入的频道。

返回数据

名称	类型	示例值	描述
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CF8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com/?Action=DeleteChannel
&AppId=xxx
&ChannelId=xxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DeleteChannelResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CF8</RequestId>
```

```
</DeleteChannelResesponse>
```

JSON 格式

```
{
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CF8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

6.2 RemoveTerminals

调用RemoveTerminals将指定终端用户从频道踢出。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RemoveTerminals	操作接口名，系统规定参数，取值：RemoveTerminals。
AppId	String	是	yourAppId	应用ID，通过控制台开通创建。
ChannelId	String	是	yourChannelId	频道ID，加入的频道。
TerminalIds.N	RepeatList	是	yourTerminalIds	终端ID列表。

返回数据

名称	类型	示例值	描述
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。
Terminals			终端信息。
Code	Integer	0	状态码。

名称	类型	示例值	描述
Id	String	181131	终端ID。
Message	String	Success	删除终端操作结果。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=RemoveTerminals
&ChannelId=xxxx
&Terminals.N=xxx
&AppId=xxxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<RemoveTerminalsResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <Terminals>
    <Id>181131</Id>
    <Code>0</Code>
    <Message>Success</Message>
  </Terminals>
  <Terminals>
    <Id>181131</Id>
    <Code>0</Code>
    <Message>Success</Message>
  </Terminals>
</RemoveTerminalsResponse>
```

JSON 格式

```
{
  "Terminals": [
    {
      "Message": "Success",
      "Id": "181131",
      "Code": 0
    },
    {
      "Message": "Success",
      "Id": "181131",
      "Code": 0
    }
  ],
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

7 数据服务

7.1 DescribeRtcDurationData

调用DescribeRtcDurationData获取应用在一段时间内的累计通信时长，区分音视频规格进行统计。



说明：

数据查询的起止时间跨度最大为30天。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRtcDurationData	操作接口名，系统规定参数，取值：DescribeRtcDurationData。
AppId	String	否	AppId	应用ID，不填则返回所有应用汇总数据。
EndTime	String	否	2018-01-29T00:00:00Z	结束时间，UTC格式。
Interval	String	否	3600	时间粒度参数，单位：秒。 取值为3600（小时粒度）或86400（天粒度），默认为3600。
ServiceArea	String	否	cn	服务区域。
StartTime	String	否	2018-01-29T00:00:00Z	起始时间，UTC格式。

返回数据

名称	类型	示例值	描述
DurationDataPerInterval			通信时长统计数据结构。
AudioDuration	Long	200	纯音频通信时长，音频为基础规格。
ContentDuration	Long	200	屏幕共享时长，音频为基础规格。
TimeStamp	String	2018-01-29T00:00:00Z	时间戳，UTC格式。
TotalDuration	Long	1000	通信总时长，单位：分钟。
V1080Duration	Long	300	全高清通信时长，视频为1920X1080及以下，音频为基础规格。
V360Duration	Long	300	标清通信时长，视频为640X480及以下，音频为基础规格。
V720Duration	Long	200	高清通信时长，视频为1280X720及以下，音频为基础规格。
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeRtcDurationData
&AppId=xxx
&StartTime=2018-01-29T00:00:00Z
&EndTime=2018-01-30T00:00:00Z
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeRtcDurationDataResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <DurationDataPerInterval>
    <TimeStamp>2018-01-29T00:00:00Z</TimeStamp>
    <TotalDuration>1000</TotalDuration>
```



```
<AudioDuration>200</AudioDuration>
<V360Duration>300</V360Duration>
<V720Duration>300</V720Duration>
<V1080Duration>300</V1080Duration>
<ContentDuration>200</ContentDuration>
</DurationDataPerInterval>
</DescribeRtcDurationDataResponse>
```

JSON 格式

```
{
  "DurationDataPerInterval": [
    {
      "TimeStamp": "2018-01-29T00:00:00Z",
      "ContentDuration": 200,
      "AudioDuration": 200,
      "TotalDuration": 1000,
      "V1080Duration": 300,
      "V360Duration": 300,
      "V720Duration": 300
    }
  ],
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

7.2 DescribeRtcUserCntData

调用DescribeRtcUserCntData查询应用在一段时间内的活跃用户数，即发生通信的用户终端数。



说明：

数据查询的起止时间跨度最大为30天。

调试

您可以在[OpenAPI Explorer](#)中直接运行该接口，免去您计算签名的困扰。运行成功后，[OpenAPI Explorer](#)可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRtcUserCntData	操作接口名，系统规定参数，取值：DescribeRtcUserCntData。

名称	类型	是否必选	示例值	描述
AppId	String	否	AppId	应用ID，不填则返回所有应用的汇总数据。
EndTime	String	否	2018-01-29T00:00:00Z	结束时间，UTC格式。
Interval	String	否	3600	时间粒度参数，单位：秒。 取值为3600（小时粒度）或86400（天粒度），默认为3600。
ServiceArea	String	否	cn	服务区域。
StartTime	String	否	2018-01-29T00:00:00Z	起始时间，UTC格式。

返回数据

名称	类型	示例值	描述
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。
UserCntDataPerInterval			活跃用户统计数据结构。
ActiveUserCnt	Long	10	当前活跃用户数（基于发生通信的用户终端统计）。
TimeStamp	String	2018-01-29T00:00:00Z	时间戳，UTC格式。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeRtcUserCntData
&AppId=xxx
&StartTime=2018-01-29T00:00:00Z
&EndTime=2018-01-30T00:00:00Z
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeRtcUserCntDataResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <UserCntDataPerInterval>
    <TimeStamp>2018-01-29T00:00:00Z</TimeStamp>
    <ActiveUserCnt>10</ActiveUserCnt>
  </UserCntDataPerInterval>
</DescribeRtcUserCntDataResponse>
```

JSON 格式

```
{
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8",
  "UserCntDataPerInterval": [
    {
      "ActiveUserCnt": 10,
      "TimeStamp": "2018-01-29T00:00:00Z"
    }
  ]
}
```

错误码

访问[错误中心](#)查看更多错误码。

7.3 DescribeRtcChannelCntData

调用DescribeRtcChannelCntData查询应用在一段时间内的活跃频道数，即发生通信的频道数。



说明：

数据查询的起止时间跨度最大为30天。

调试

您可以在[OpenAPI Explorer](#)中直接运行该接口，免去您计算签名的困扰。运行成功后，[OpenAPI Explorer](#)可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRtcChannelCntData	操作接口名，系统规定参数，取值：DescribeRtcChannelCntData。
AppId	String	否	AppId	应用ID，不填则返回所有应用的汇总数据。

名称	类型	是否必选	示例值	描述
EndTime	String	否	2018-01-29T00:00:00Z	结束时间，UTC格式。
Interval	String	否	3600	时间粒度参数，单位：秒。 取值为3600（小时粒度）或86400（天粒度），默认为3600。
ServiceArea	String	否	cn	服务区域。
StartTime	String	否	2018-01-29T00:00:00Z	起始时间，UTC格式。

返回数据

名称	类型	示例值	描述
ChannelCntDataPerInterval			活跃频道统计数据结构。
ActiveChannelCnt	Long	10	活跃频道数量统计（时间单元累计）。
TimeStamp	String	2018-01-29T00:00:00Z	时间戳，UTC格式。
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeRtcChannelCntData
&AppId=xxx
&StartTime=2018-01-29T00:00:00Z
&EndTime=2018-01-30T00:00:00Z
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeRtcChannelCntDataResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <ChannelCntDataPerInterval>
    <TimeStamp>2018-01-29T00:00:00Z</TimeStamp>
```

```
<ActiveChannelCnt>10</ActiveChannelCnt>
</ChannelCntDataPerInterval>
</DescribeRtcChannelCntDataResesponse>
```

JSON 格式

```
{
  "ChannelCntDataPerInterval": [
    {
      "TimeStamp": "2018-01-29T00:00:00Z",
      "ActiveChannelCnt": 10
    }
  ],
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

7.4 DescribeRtcPeakUserCntData

调用DescribeRtcPeakUserCntData查询应用在一段时间内的并发通信峰值，一组“发布-订阅”关系被标记为一次通信。



说明：

数据查询的起止时间跨度最大为30天。

调试

您可以在[OpenAPI Explorer](#)中直接运行该接口，免去您计算签名的困扰。运行成功后，[OpenAPI Explorer](#)可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRtcPeakUserCntData	操作接口名。系统规定参数，取值：DescribeRtcPeakUserCntData。
AppId	String	否	yourAppId	应用ID，不填则返回所有应用的汇总数据。
EndTime	String	否	2018-01-29T00:00:00Z	结束时间，UTC格式。

名称	类型	是否必选	示例值	描述
Interval	String	否	3600	时间粒度参数，单位：秒。 取值为3600（小时粒度）或86400（天粒度），默认值为3600。
ServiceArea	String	否	cn	服务区域。
StartTime	String	否	2018-01-29T00:00:00Z	起始时间，UTC格式。

返回数据

名称	类型	示例值	描述
PeakUserCntDataPerInterval			峰值并发通信统计数据结构。
ActiveUserPeak	Long	10	峰值时刻的并发通信数。
ActiveUserPeakTime	String	2018-01-29T00:01:00Z	并发通信峰值时刻，UTC格式。
TimeStamp	String	2018-01-29T00:00:00Z	时间戳，UTC格式。
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeRtcPeakUserCntData
&AppId=xxx
&StartTime=2018-01-29T00:00:00Z
&EndTime=2018-01-30T00:00:00Z
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeRtcPeakUserCntDataResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <PeakUserCntDataPerInterval>
    <TimeStamp>2018-01-29T00:00:00Z</TimeStamp>
```

```
<ActiveUserPeak>10</ActiveUserPeak>
<ActiveUserPeakTime>2018-01-29T00:01:00Z</ActiveUserPeakTime>
</PeakUserCntDataPerInterval>
</DescribeRtcPeakUserCntDataResponse>
```

JSON 格式

```
{
  "PeakUserCntDataPerInterval": [
    {
      "TimeStamp": "2018-01-29T00:00:00Z",
      "ActiveUserPeakTime": "2018-01-29T00:01:00Z",
      "ActiveUserPeak": 10
    }
  ],
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

7.5 DescribeRtcPeakChannelCntData

调用DescribeRtcPeakChannelCntData查询应用在一段时间内的并发频道峰值数量。



说明:

数据查询的起止时间跨度最大为30天。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DescribeRtcPeakChannelCntData	操作接口名，系统规定参数，取值：DescribeRtcPeakChannelCntData。
AppId	String	否	yourAppId	应用ID，不填返回所有应用的汇总数据。
EndTime	String	否	2018-01-29T00:00:00Z	结束时间，UTC格式。

名称	类型	是否必选	示例值	描述
Interval	String	否	3600	时间粒度参数，单位：秒。 取值为3600（小时粒度）或86400（天粒度），默认为3600。
ServiceArea	String	否	cn	服务区域。
StartTime	String	否	2018-01-29T00:00:00Z	起始时间，UTC格式。

返回数据

名称	类型	示例值	描述
PeakChannelCntDataPerInterval			并发频道峰值数据结构。
ActiveChannelPeak	Long	10	并发频道峰值数量。
ActiveChannelPeakTime	String	2018-01-29T00:01:00Z	峰值时刻，UTC格式。
TimeStamp	String	2018-01-29T00:00:00Z	时间戳，UTC格式。
RequestId	String	16A96B9A-F203-4EC5-8E43-CB92E68F4CD8	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=DescribeRtcPeakChannelCntData
&AppId=xxx
&StartTime=2018-01-29T00:00:00Z
&EndTime=2018-01-30T00:00:00Z
&<公共请求参数>
```

正常返回示例

XML 格式

```
<DescribeRtcPeakChannelCntDataResponse>
  <RequestId>16A96B9A-F203-4EC5-8E43-CB92E68F4CD8</RequestId>
  <PeakChannelCntDataPerInterval>
    <TimeStamp>2018-01-29T00:00:00Z</TimeStamp>
```



```
<ActiveChannelPeak>10</ActiveChannelPeak>
<ActiveChannelPeakTime>2018-01-29T00:01:00Z</ActiveChan
nelPeakTime>
</PeakChannelCntDataPerInterval>
</DescribeRtcPeakChannelCntDataResesponse>
```

JSON 格式

```
{
  "PeakChannelCntDataPerInterval": [
    {
      "TimeStamp": "2018-01-29T00:00:00Z",
      "ActiveChannelPeak": 10,
      "ActiveChannelPeakTime": "2018-01-29T00:01:00Z"
    }
  ],
  "RequestId": "16A96B9A-F203-4EC5-8E43-CB92E68F4CD8"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8 旁路直播

8.1 StartMPUTask

调用StartMPUTask开始任务。

调用该接口前，请您注意：

- 禁止同一个StreamURL（直播推流地址）在不同Task（任务）中同时使用。
- Task（任务）停止10S之内，禁止使用同一个StreamURL（直播推流地址）。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	StartMPUTask	操作接口名，系统规定参数，取值：StartMPUTask。
AppId	String	是	yourAppId	应用ID，创建应用后生成。
ChannelId	String	是	yourChannelId	频道ID。
LayoutIds.N	RepeatList	是	1	布局ID数据，用户可在一次任务中指定多个布局，系统会根据当时channel中的人数进行切换。详情请参见 布局 。 N的取值范围1~15。
MediaEncode	Integer	是	2	编码选项，具体请参见本文中的MediaEncode枚举值。

名称	类型	是否必选	示例值	描述
StreamURL	String	是	yourStreamURL	直播推流地址，生成规则请参见 推流地址与播流地址（原画） 。 对已开防盗链鉴权的域名，需要在推流地址中包含鉴权串。
TaskId	String	是	yourTaskId	任务ID，此ID为旁路直播的标识，需保证唯一。 字符只允许A-Za-z0-9_-，长度限制64字节。
BackgroundColor	Integer	否	0	背景色RGB，默认是0（黑色）。
CropMode	Integer	否	0	视频的裁剪方式，默认值为2。 0：不保持比例铺满。 1：保持比例裁剪。 2：保持比例留边。
TaskProfile	String	否	4IN_720P	任务计费配置，根据您的不同设置，进行收费。TaskProfile设置： 4IN_720P。 2IN_720P。 1IN_720P。 4IN_360P。 2IN_360P。 1IN_360P。 - Mixed_Audio。 1IN_Audio。
UserPanes.N.PaneId	Integer	否	2	布局ID，从左上到右下排序，从0开始。N的取值范围1~15。
UserPanes.N.SourceType	String	否	camera	对应布局的用户视频输入，camera和shareScreen两种。N的取值范围1~15。

名称	类型	是否必选	示例值	描述
UserPanes.N. UserId	String	否	UserId	对应布局框格的用户ID。N的取值范围1~15。

MediaEncode枚举值如下所示。

ID	分辨率	码流	帧率
0	audio only	—	—
1	360p (640 x 360)	500Kbps	15fps
10	540p (960 x 540)	700Kbps	24fps
20	720p (1280 x 720)	1Mbps	25fps
22	720p (720 x 1280 , 竖屏)	1Mbps	30fps
30	1080p (1920 x 1080)	2Mbps	30fps

返回数据

名称	类型	示例值	描述
RequestId	String	760bad5327 6431c499e3 0dc36f6b26be	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=StartMPUTask
&AppId=xxxx
&ChannelId=xxx
&TaskId=xxx
&MediaEncode=2
&BackgroundColor=0
&LayoutIds.1=1
&LayoutIds.2=2
&UserPanes.0.PaneId=xxx
&UserPanes.0.UserId=xxx
&UserPanes.0.SourceType=xxx
&UserPanes.1.PaneId=xxx
```

```
&UserPanes.1.UserId=xxx
&UserPanes.1.SourceType=xxx
&StreamURL=xxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<StartMPUTaskResponse>
  <RequestId>760bad53276431c499e30dc36f6b26be</RequestId>
</StartMPUTaskResponse>
```

JSON 格式

```
{
  "RequestId": "760bad53276431c499e30dc36f6b26be"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.2 GetMPUTaskStatus

调用GetMPUTaskStatus获取任务状态。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetMPUTaskStatus	操作接口名，系统规定参数，取值：GetMPUTaskStatus。
AppId	String	是	yourAppId	应用ID，创建应用后生成。
TaskId	String	是	yourTaskId	任务ID。

返回数据

名称	类型	示例值	描述
RequestId	String	760bad53276431c499e30dc36f6b26be	请求ID。
Status	Integer	0	任务的状态ID： · 0：等待channel开始。 · 1：任务运行中。 · 2：任务已停止。 · 3：用户停止任务。 · 4：Channel已停止。 · 5：CDN网络问题，直播停止。 · 6：直播URL问题，直播停止。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=GetMPUTaskStatus
&AppId=xxxx
&ChannelId=xxx
&TaskId=xxx<公共请求参数>
```

正常返回示例

XML 格式

```
<GetMPUTaskStatusResponse>
  <RequestId>760bad53276431c499e30dc36f6b26be</RequestId>
  <Status>0</Status>
</GetMPUTaskStatusResponse>
```

JSON 格式

```
{
  "Status":0,
  "RequestId":"760bad53276431c499e30dc36f6b26be"
}
```

错误码

访问[错误中心](#)查看更多错误码。

8.3 StopMPUTask

调用StopMPUTask停止任务。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	StopMPUTask	操作接口名，系统规定参数，取值：StopMPUTask。
AppId	String	是	yourAppId	应用ID，创建应用后生成。
TaskId	String	是	yourTaskId	任务ID。

返回数据

名称	类型	示例值	描述
RequestId	String	760bad5327 6431c499e3 0dc36f6b26be	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=StopMPUTask
&AppId=xxxx
&TaskId=xxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<StopMPUTaskResponse>
  <RequestId>760bad53276431c499e30dc36f6b26be</RequestId>
</StopMPUTaskResponse>
```

JSON 格式

```
{
  "RequestId": "760bad53276431c499e30dc36f6b26be"
```

```
}

```

错误码

访问[错误中心](#)查看更多错误码。

8.4 UpdateMPULayout

调用UpdateMPULayout更新任务的布局。

调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	UpdateMPULayout	操作接口名，系统规定参数，取值：UpdateMPULayout。
AppId	String	是	yourAppId	应用ID，控制台查询。
LayoutIds.N	RepeatList	是	1	布局ID数据，用户可在一次任务中指定多个布局，系统会根据当时channel（频道）中的人数进行切换。详情请参见 布局 。 N的取值范围1~15。
TaskId	String	是	abcde	任务ID，此ID为旁路直播的标识，需保证唯一。 字符只允许A-Za-z0-9_-，长度限制64字节。
Background Color	Integer	否	0	背景色RGB，默认是0（黑色）。
CropMode	Integer	否	2	视频的裁剪模式： · 0：不保持比例铺满。 · 1：保持比例裁剪。 · 2（默认）：保持比例留边。

名称	类型	是否必选	示例值	描述
UserPanes.N. PaneId	Integer	否	2	布局ID，从左上到右下排序，从0开始。 N的取值范围1~15。
UserPanes.N. SourceType	String	否	camera	对应布局的用户视频输入，camera和shareScreen两种。
UserPanes.N. UserId	String	否	UserId	对应布局的用户视频输入，camera和shareScreen两种。 N的取值范围1~15。

返回数据

名称	类型	示例值	描述
RequestId	String	760bad5327 6431c499e3 0dc36f6b26be	请求ID。

示例

请求示例

```
https://rtc.aliyuncs.com?Action=UpdateMPULayout
&AppId=xxxx
&TaskId=xxx
&BackgroundColor=0
&LayoutIds.1=1
&LayoutIds.1=2
&UserPanes.0.PaneId=xxx
&UserPanes.0.UserId=xxx
&UserPanes.0.SourceType=xxx
&UserPanes.1.PaneId=xxx
&UserPanes.1.UserId=xxx
&UserPanes.1.SourceType=xxx
&<公共请求参数>
```

正常返回示例

XML 格式

```
<UpdateMPULayoutResponse>
  <RequestId>760bad53276431c499e30dc36f6b26be</RequestId>
```

```
</UpdateMPULayoutResponse>
```

JSON 格式

```
{  
  "RequestId":"760bad53276431c499e30dc36f6b26be"  
}
```

错误码

访问[错误中心](#)查看更多错误码。

9 服务端错误码表

本文为您汇总了服务端API返回的错误码，方便您快速定位错误信息，更好地调用音视频通信相关API。

十六进制	十进制	HTTP错误码	错误名	描述
0x02070101	34013441	400	Request. NoTimestamp	URL中没有Timestamp
0x02070102	34013442	400	Request. NoSignature	URL中没有Signature
0x02070103	34013443	400	Request. NoRequestID	URL中没有RequestID
0x02070104	34013444	400	Request. NoBizID	URL中没有BizID
0x02070105	34013445	400	Request. NoAliUid	URL中没有AliUid
0x02070106	34013446	400	Request. NoAppID	请求中没有AppID
0x02070107	34013447	400	Request. NoDescription	请求中没有AppDescription
0x02070108	34013448	400	Request. NoAppType	请求中没有AppType
0x02070109	34013449	400	Request. NoServiceAreas	请求中没有ServiceAreas
0x0207010A	34013450	400	Request. NoBillType	请求中没有BillType
0x0207010B	34013451	400	Request. NoChannelID	请求中没有ChannelID
0x0207010C	34013452	400	Request. NoUserGroups	请求中没有UserGroups
0x0207010D	34013453	400	Request. NoUserID	请求中没有UserID
0x0207010E	34013454	400	Request. NoSessionID	请求中没有SessionID

十六进制	十进制	HTTP错误码	错误名	描述
0x0207010F	34013455	400	Request.NoNonce	请求中没有Nonce
0x02070110	34013456	400	Request.NoTerminalIDs	请求中没有TerminalIDs
0x02070111	34013457	400	Request.MismatchBizID	请求中BizID不匹配
0x02070112	34013458	400	Request.InvalidAppID	请求中AppID非法
0x02070113	34013459	400	Request.InvalidAppKey	请求中AppKey非法
0x02070114	34013460	400	Request.InvalidAppCallback	请求中AppCallback非法
0x02070115	34013461	400	Request.InvalidChannelID	请求中ChannelID非法
0x02070116	34013462	400	Request.InvalidChannelKey	请求中ChannelKey非法
0x02070117	34013463	400	Request.InvalidNonce	请求中Nonce非法
0x02070118	34013464	400	Request.InvalidTimestamp	请求中Timestamp非法
0x02070119	34013465	400	Request.IllegalTerminal	请求中Terminal非法
0x0207011A	34013466	400	Request.ChannelQueryExpire	查询的Channel已过期
0x0207011B	34013467	400	Request.ExistApp	请求的App已存在
0x0207011C	34013468	400	Request.ExistAppCallback	请求的AppCallback已存在
0x0207011D	34013469	400	Request.ExistChannel	请求的Channel已存在

十六进制	十进制	HTTP错误码	错误名	描述
0x02070401	34014209	401	Request. IllegalTim estamp	Timestamp字段 非法
0x02070402	34014210	401	Request. IllegalSig nature	Signature验证失 败
0x02070403	34014211	401	Request. IllegalAuth	Auth字段校验失 败
0x02070404	34014212	401	Request. IllegalBizID	BizID字段校验失 败
0x02070405	34014213	401	Request. IllegalAliUID	AliUID字段校验 失败
0x02070406	34014214	401	Request. IllegalBro adcast	Broadcast字段 校验失败
0x02070407	34014215	401	Request. IllegalUse rGroups	UserGroups字段 校验失败
0x02070801	34015233	404	Service. NoSignaling	未获取到信令服务
0x02070802	34015234	404	Service.NoApp	App不存在
0x02070803	34015235	404	Service. NoAppCallback	AppCallback不 存在
0x02070804	34015236	404	Service. NoChannel	Channel不存在
0x02070201	34013697	500	Service. BuildSignature	生成signature失 败
0x02070202	34013698	500	Service. ParseHTTPB ody	解析http body失 败
0x02070203	34013699	500	Service. AppRDSOper ate	App RDS访问失 败
0x02070204	34013700	500	Service. AppCallbac kRDSOperate	App callback RDS访问失败

十六进制	十进制	HTTP错误码	错误名	描述
0x02070205	34013701	500	Service. ChannelRDS Operate	channel RDS访问失败
0x02070206	34013702	500	Service. GenerateChannelKey	生成channel key失败
0x02070207	34013703	500	Service. GenerateChannelToken	生成channel token失败
0x02070208	34013704	500	Service. SignalingSendMessage	向signaling发送消息失败
0x02071001	34017281	200	Service. Inactive	服务不可用
0x02071002	34017282	200	Channel. NotExist	频道不存在
0x02071003	34017283	200	Service.NoExist	服务不存在
0x02071004	34017284	200	App.Stop	应用停用，请在控制台重新恢复该AppID到可用状态
0x02071005	34017285	200	App. ChargeOverDue	AppID已欠费，请续费
0x02071006	34017286	200	Channel. Exceeds	超过最大可用频道数，请删除不再使用的频道
0x02071007	34017287	200	Request. ChannelQueryExpire	查询的Channel已过期
0x02071008	34017288	200	Request. ExistApp	请求的App已存在
0x02071009	34017289	200	Request. ExistAppCallback	请求的AppCallback已存在
0x0207100A	34017290	200	Request. ExistChannel	请求的Channel已存在