物联网智能视频服务（消费版）

开发者手册

腾讯云

版权所有：腾讯云计算（北京）有限责任公司

【版权声明】

©2013-2025 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。
您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

开发者手册
P2P 接入指南
云存接入指南
物模型协议
消息队列接入指南
设备端与应用端信令交互说明
腾讯连连小程序交互信令说明

版权所有：腾讯云计算（北京）有限责任公司

开发者手册
P2P 接入指南

最近更新时间：2023-04-24 09:47:42

物联网智能视频服务（消费版）平台为用户提供了基于 X-P2P 协议的 P2P 接入能力。本文档为您介绍 P2P 接入相关内容。

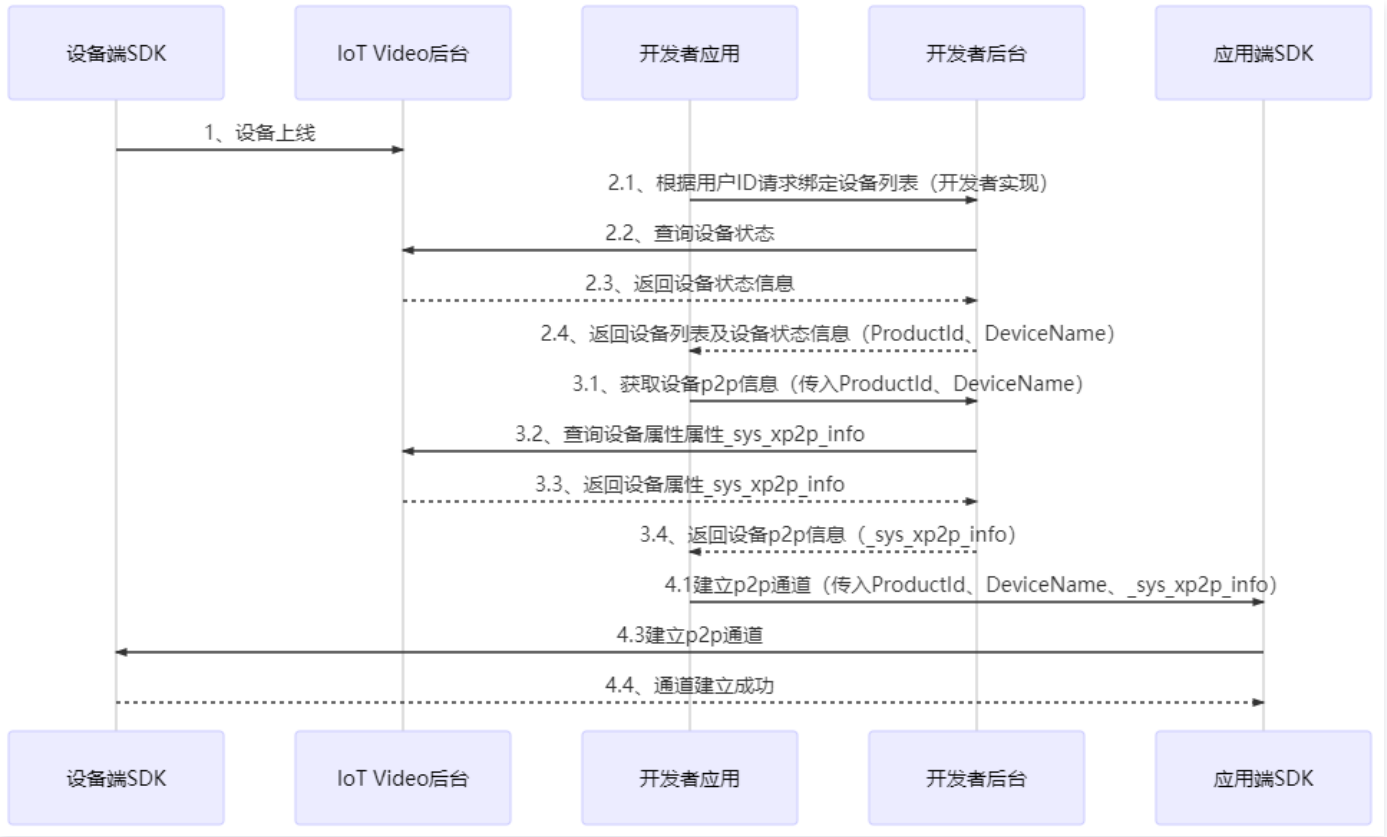
功能概述

- 目前一个设备最多支持建立4路 P2P 传输通道。
- P2P 通道除了支持音视频传输，还支持用户自定义数据传输。

操作步骤

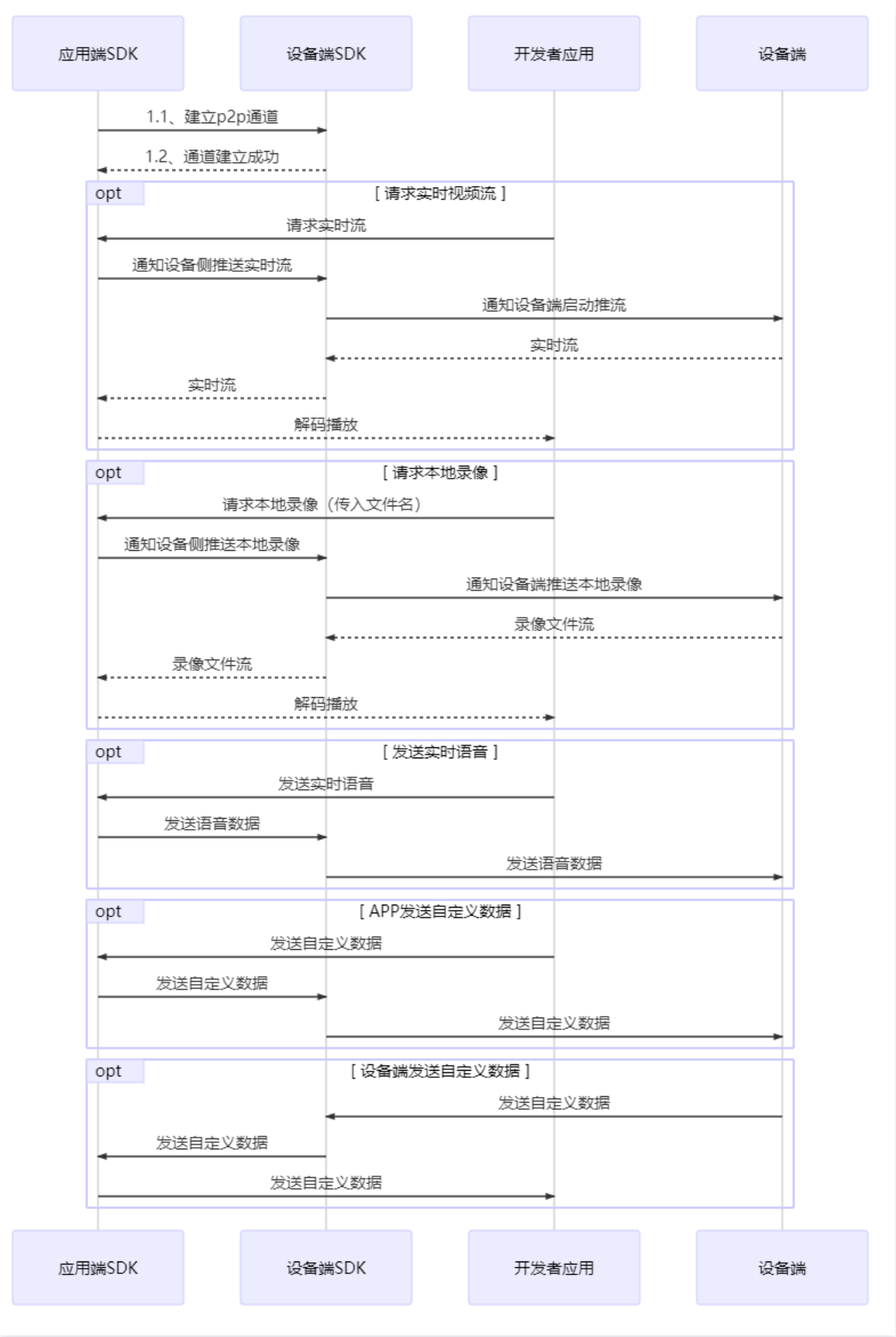
建立 P2P 连接

用户需要通过 X-P2P 进行应用端与已绑定的设备端之间的音视频传输，需要先建立 P2P 的连接，开发者后台通过 2.2查询设备状态 及 3.2查询设备属性 获取 _sys_xp2p_info。具体流程如下图：



P2P 音视频传输

应用端和设备端已建立 P2P 通道后，可进行音视频和自定义数据传输，流程如下图：



云存接入指南

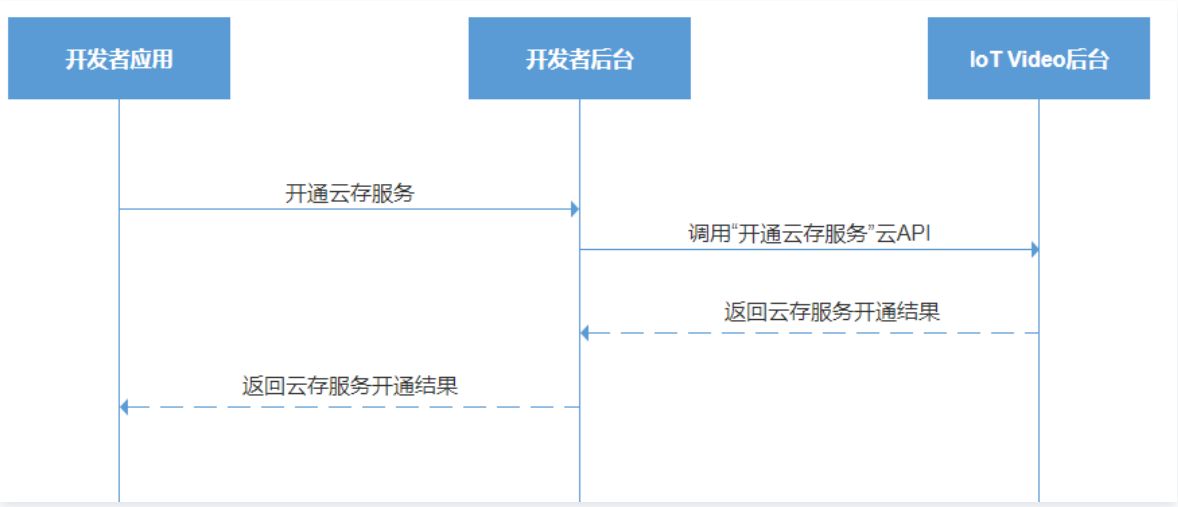
最近更新时间：2022-09-02 16:40:02

物联网智能视频服务（消费版）平台为用户提供了基于云存储功能，支持设备端音视频数据存储在云端，需要观看时，由应用端从云端拉取数据。

操作步骤

开通云存套餐

使用云存储功能前，需要先为设备 开通云存套餐，通过调用 IoT Video（Consumer Version）提供云 API 支持为设备购买云存套餐。云存套餐购买成功后，设备将通过物模型获取设备的云存套餐状态。



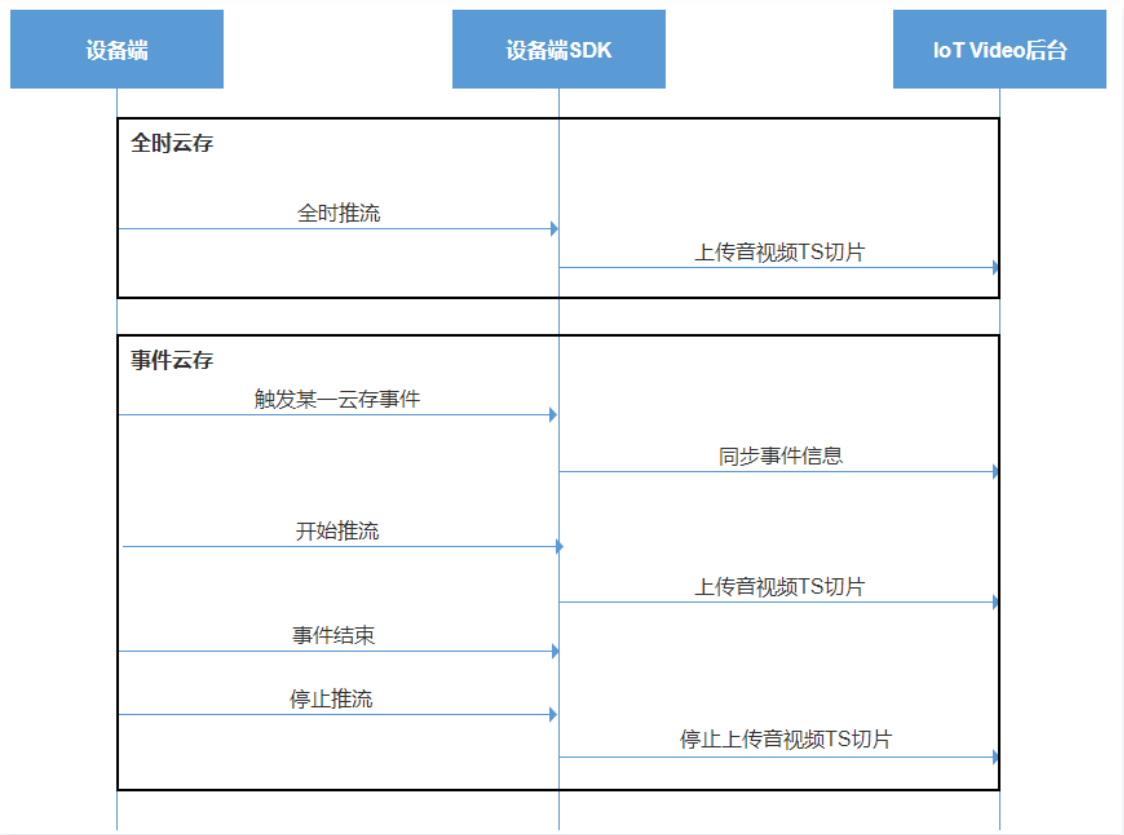
云存录像上传（视频流设备）

云存套餐分为全时云存套餐和事件云存套餐：

- **全时云存**：指设备在套餐生效时间内，可以把所有产生录像上传到云存进行存储。
- **事件云存**：是指设备在套餐生效时间内，当触发事件时，设备可以把事件发生时的录像上传到云端进行存储。

> **说明：**
> 一个设备同时只能存在一个云存套餐。

录像上传到云端存储流程如下：



查看云存录像（视频流设备）

IoT Video（Consumer Version）提供云 API 支持查询有录像的 日期 和 事件列表，并根据返回的日期和 VideoURL 拼接成 m3u8 播放地址，通过 m3u8 地址获取视频流。

m3u8 播放地址拼接

1. 通过调用 获取具有云存的日期 API接口获取云存日期
2. 调用 获取某一天云存时间轴 API获取指定日期的云存录像播放地址（VideoUrl）和每个录像片段的开始时间（StartTime）、结束时间（EndTime）。
3. 将返回的 VideoUrl、StartTime 和 EndTime 按 UNIX 时间戳以如下图格式进行拼接。

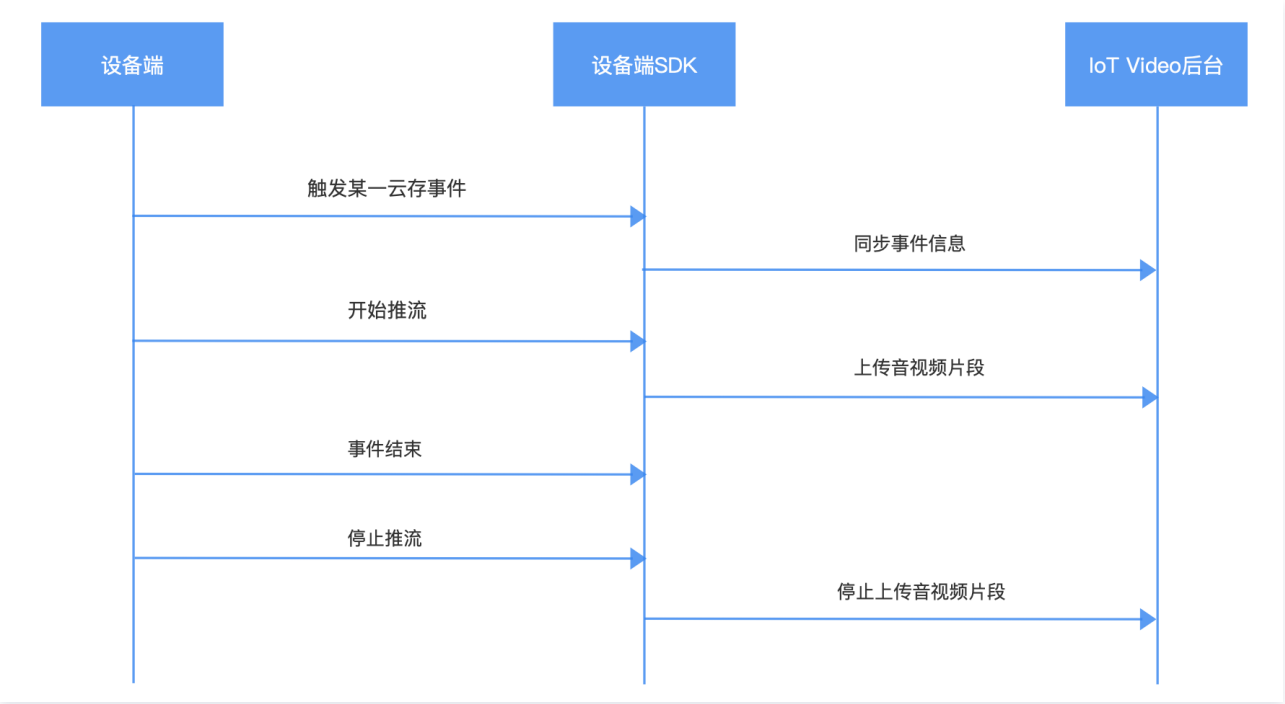
```
{VideoUrl}?starttime_epoch={StartTime}&endtime_epoch={EndTime}
```

4. 将拼接好的 VideoURL 作为入参调用 获取视频防盗链播放 URL API，即可获得该云存视频的防盗链的播放地址，进而调用小程序插件能力实现正常播放。

云存录像上传（图片流设备）

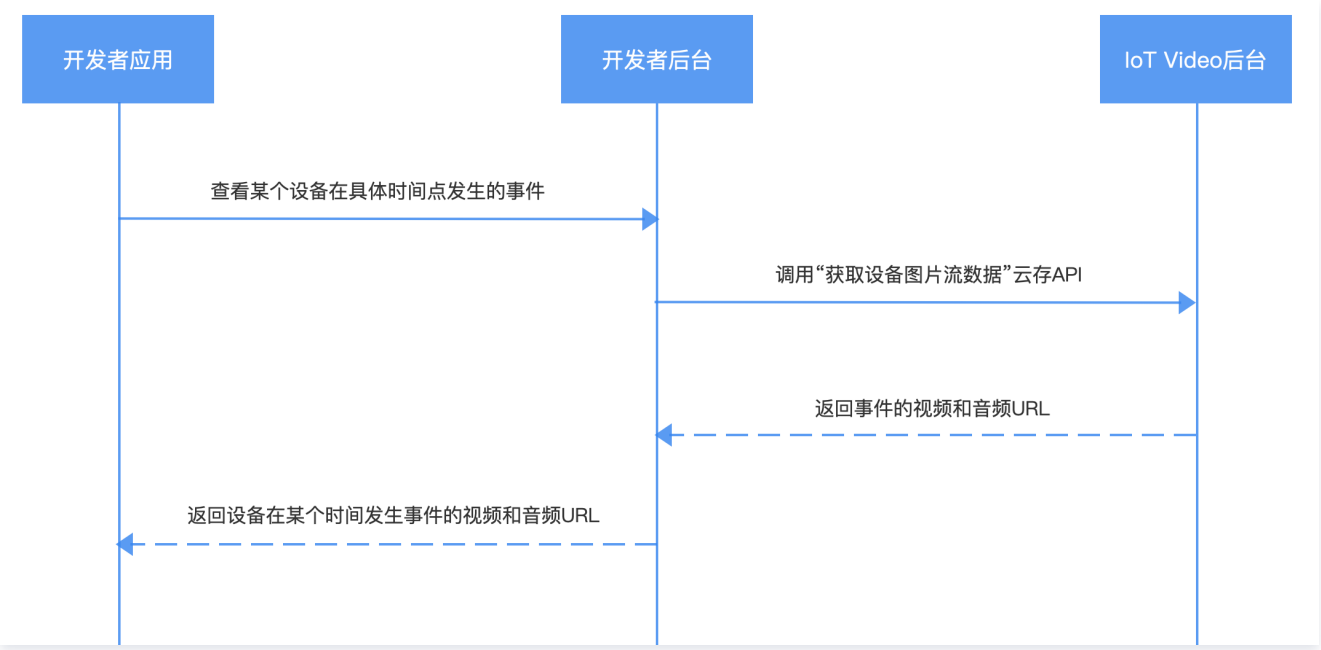
云存套餐仅有事件云存套餐。
事件云存：是指设备在套餐生效时间内，当触发事件时，设备可以把事件发生时的录像上传到云端进行存储。

> **说明：**
> 一个设备同时只能存在一个云存套餐。



查看云存录像（图片流设备）

IoT Video（Consumer Version）提供 云 API 支持获取图片流设备的事件云存录像(mjpg 格式的视频 + acc 格式的音频)。



获取图片流播放地址

1. 通过调用 获取具有云存的日期 API 接口获取事件云存的日期。
2. 通过调用 拉取云存事件列表 API 获取某个日期下的事件列表，包括每个事件的 StartTime（开始时间）、EndTime（结束时间）、Thumbnail（缩略图文件名）和 EventId（事件 ID）。
3. 调用 拉取云存事件缩略图 API 获取指定事件缩略图的访问地址。
4. 调用 获取设备图片流数据 API 获取事件的图片流视频、音频播放地址，进而调用小程序插件能力实现正常播放。

> **注意：**
> 每次事件记录的存储时间不超过10s。

物模型协议

最近更新时间：2024-08-23 17:43:51

简介

平台为用户提供了基于物模型的接入数据协议，用户创建完产品后即可定义物模型。本文档为您介绍物模型协议相关内容。

物模型设置指引

控制台设置

可通过控制台进行产品物模型设置，详情参见控制台指引指南 > 物模型 。

云API设置

可通过云 API 进行产品物模型设置，云 API 接口如下：

接口名称	备注
查询产品数据模板	获取当前产品的物模型的详细配置信息。
修改产品数据模板	修改产品物模型。
导入其他产品的数据模板	支持通过导入物模型进行物模型配置。

设备端使用指引

详情请参见设备端文档 物模型模块 。

查询物模型数据

控制台查询

可通过云日志进行查询设备物模型的变化，详情参见控制台指引 > 云日志 。

云 API 查询

接口名称	备注
获取设备属性数据	获取当前产品的属性物模型数据。
获取设备历史数据	获取指定时间上指定属性物模型的历史数据。
获取设备的历史事件	获取指定时间上指定事件类型的数据。
获取设备动作历史	获取指定时间上行为物模型的数据。

设备控制

云 API 控制

接口名称	备注
设备远程控制	用于服务器向设备下发属性物模型数据，实现设备远程控制。
设备透传指令控制	用于使用自定义透传协议进行设备远控或者传输自定义数据，设备端的数据收发参考 自定义信令模块，用户通过配置 消息队列 获取设备上报信息。向设备透传数据时，topic 配置格式为 `${ProductId}/${DeviceName}/data` 。
同步调用设备行为	用于服务器向设备同步下发行为物模型数据，实现设备远程控制。
异步调用设备行为	用于服务器向设备异步下发行为物模型数据，实现设备远程控制。

消息队列接入指南

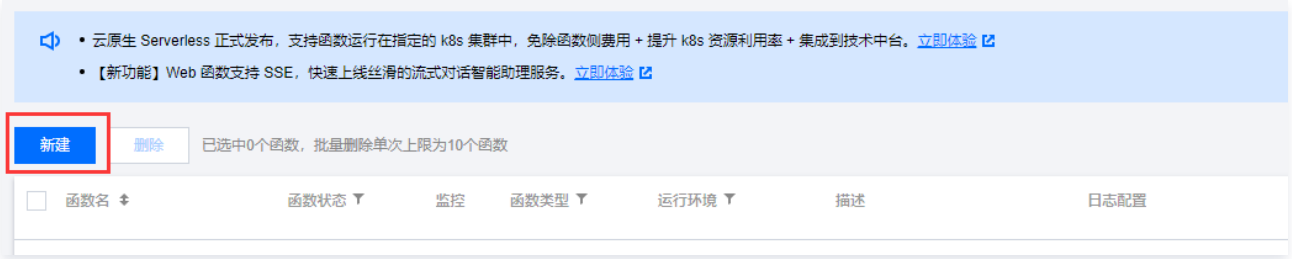
最近更新时间：2023-07-06 10:01:11

HTTP 转发入门实战

操作步骤

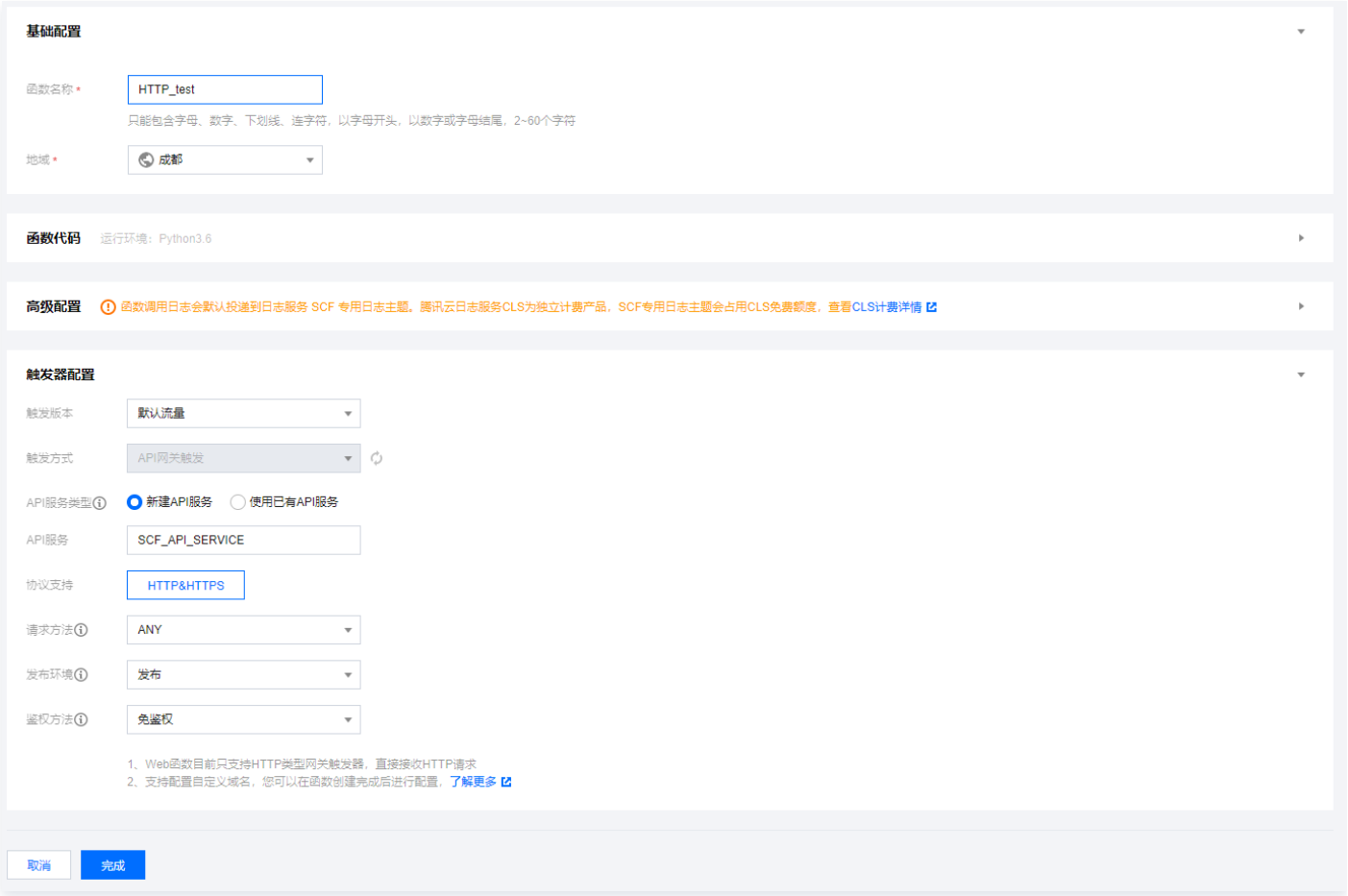
1. 登录 物联网智能视频服务（消费版），创建产品。
2. 进入产品列表页面，单击产品名称进入产品详情页。
3. 单击**消息队列**进入消息队列配置页面，单击 **HTTP 转发**进入 HTTP 转发设置页面。

4. 随后进入 云函数 > 函数服务，单击**新建**创建模板函数。

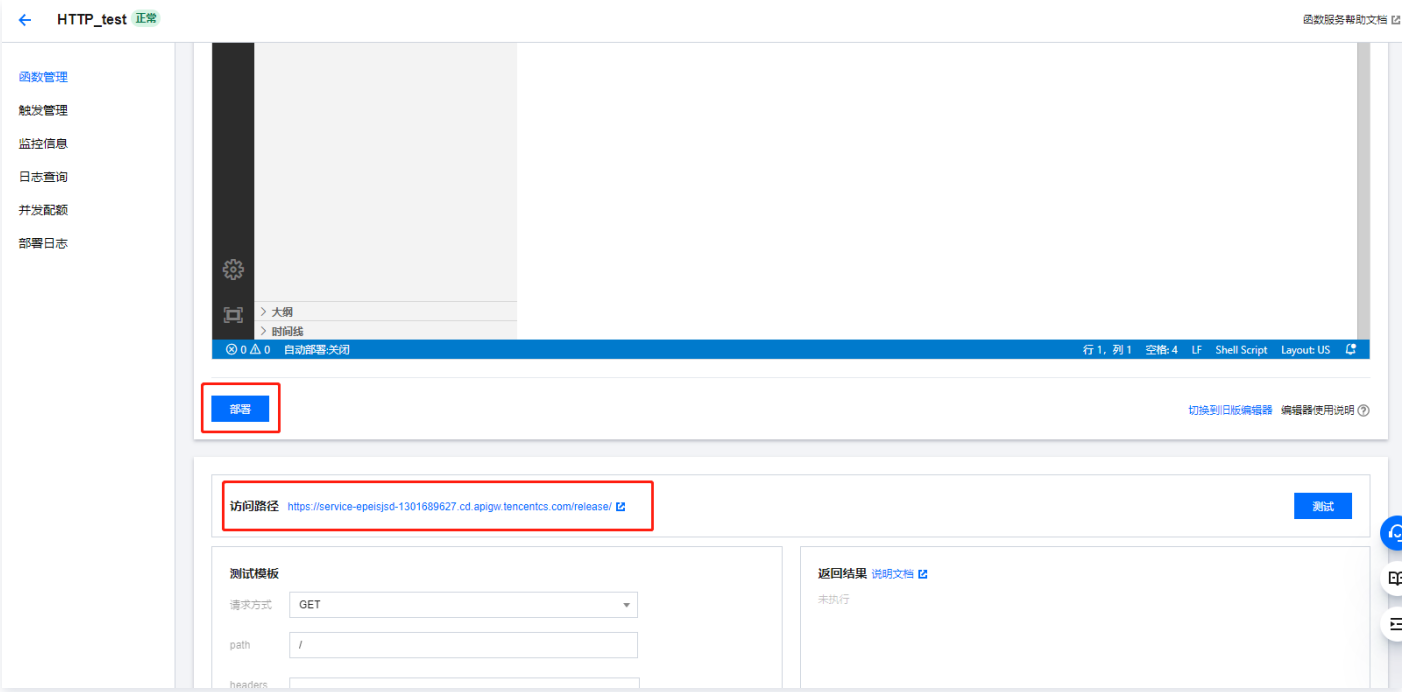


5. 创建方式选择**模板创建**，在下方“模糊搜索栏”搜索 `IoT Explorer` 。

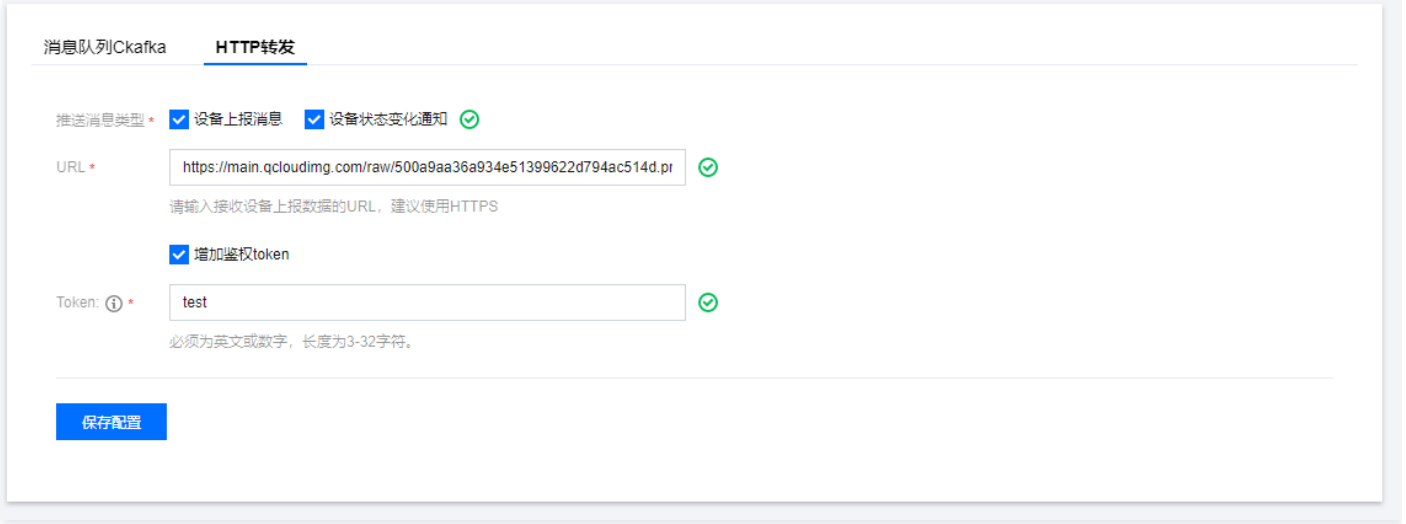
6. 选择需要语言的“IoT Explorer数据同步转发”模板后，单击**下一步**按钮。
7. 无需更改相关配置信息，单击**完成**按钮后进入函数管理页面。



8. 在“函数管理”页面选择**函数代码**功能栏，下滑单击**部署**按钮。

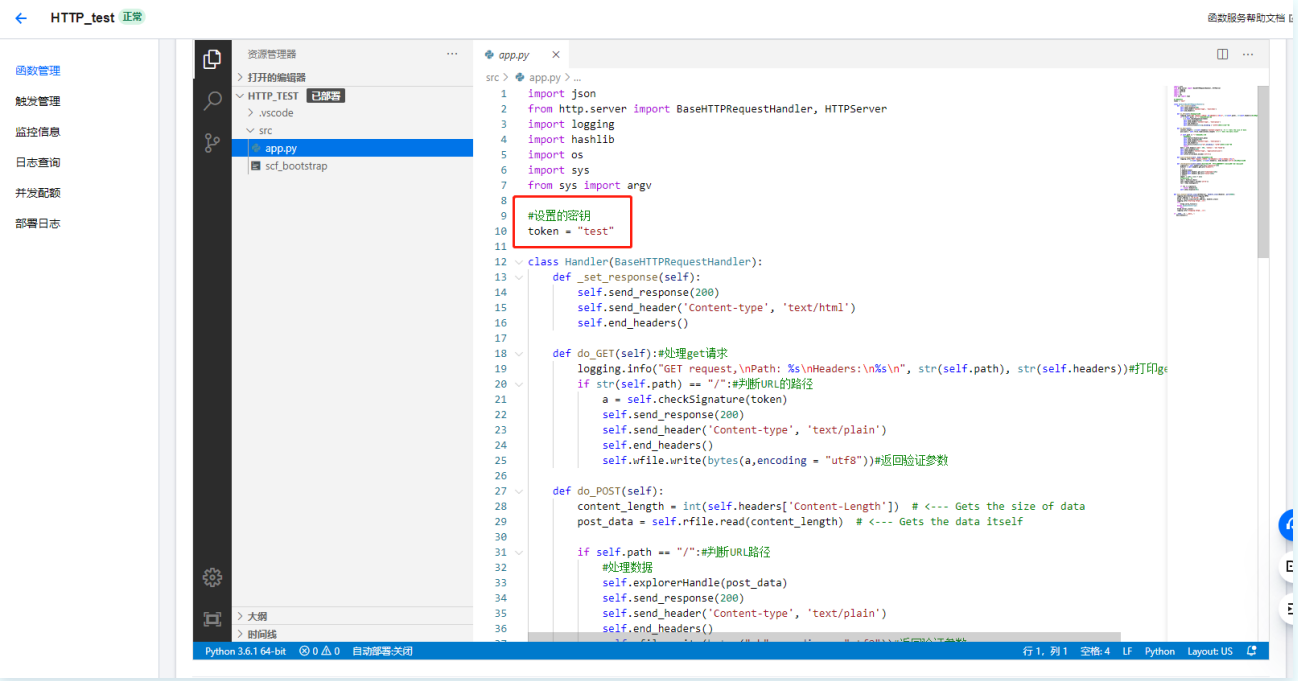


9. 部署成功后，复制“访问路径” URL（该访问路径即为数据同步转发功能内，接收设备消息数据的 URL）。
10. 回到 物联网智能视频服务（消费版），选择需要设置 HTTP 转发的产品，选择需要推送的消息类型，填写接收服务端地址，同时增加鉴权 token（默认为 test，若需自定义可查看下方的提示）。

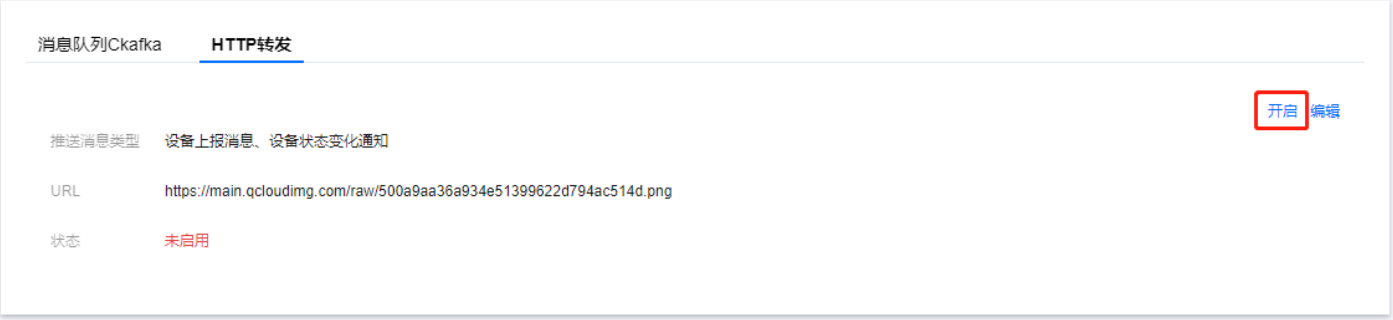


注意：

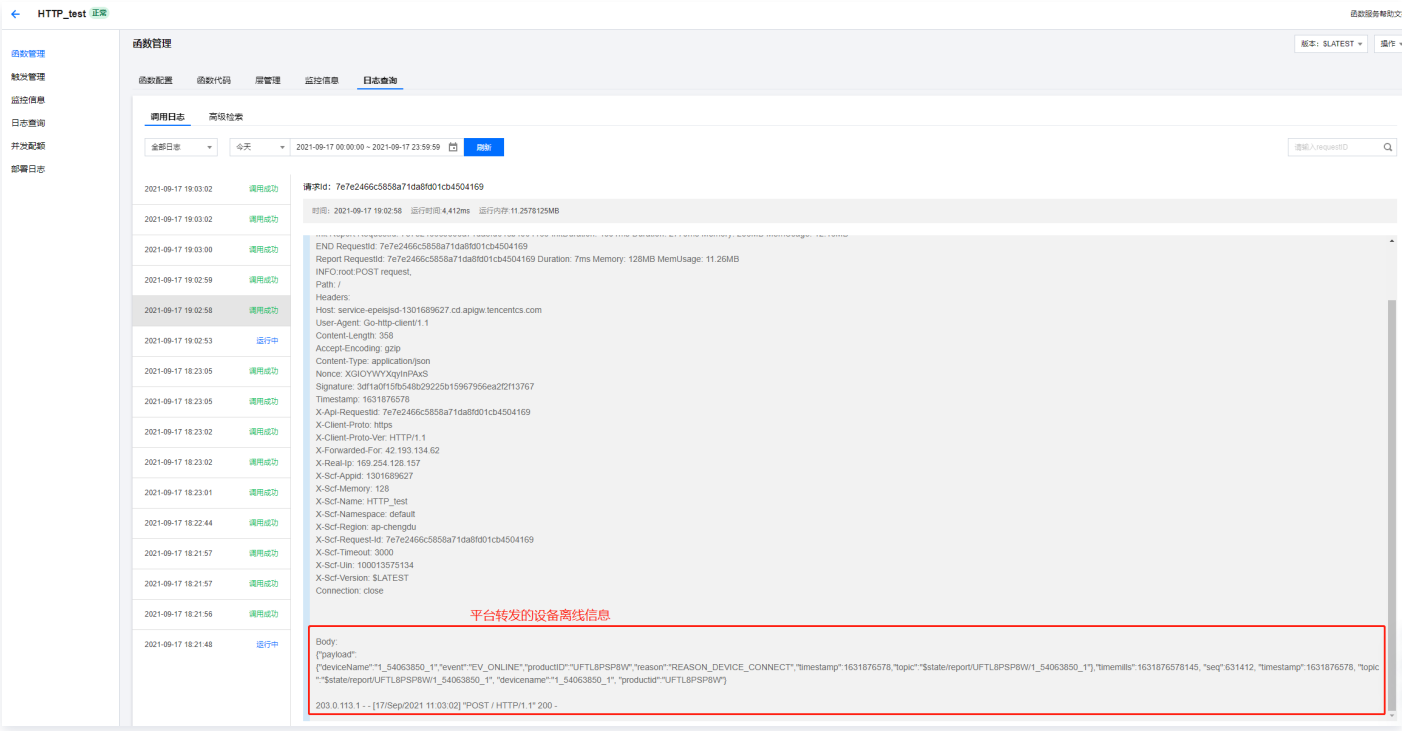
鉴权 token 如何自定义设置，可在函数部署阶段在代码中自定义配置。默认为 `test` 。

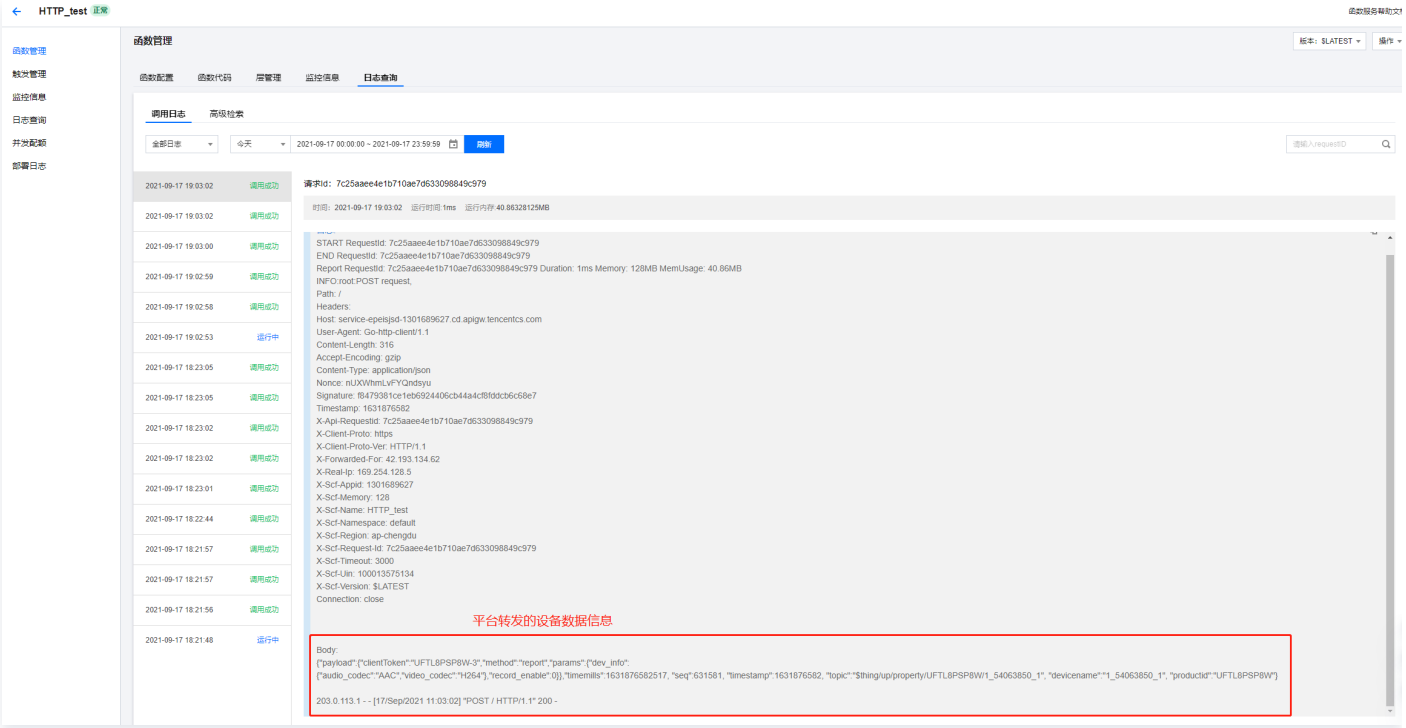


11. 配置完成后，单击**保存配置**按钮，跳转到详情页，可开启该产品的**生效状态**，完成该产品的消息队列配置。

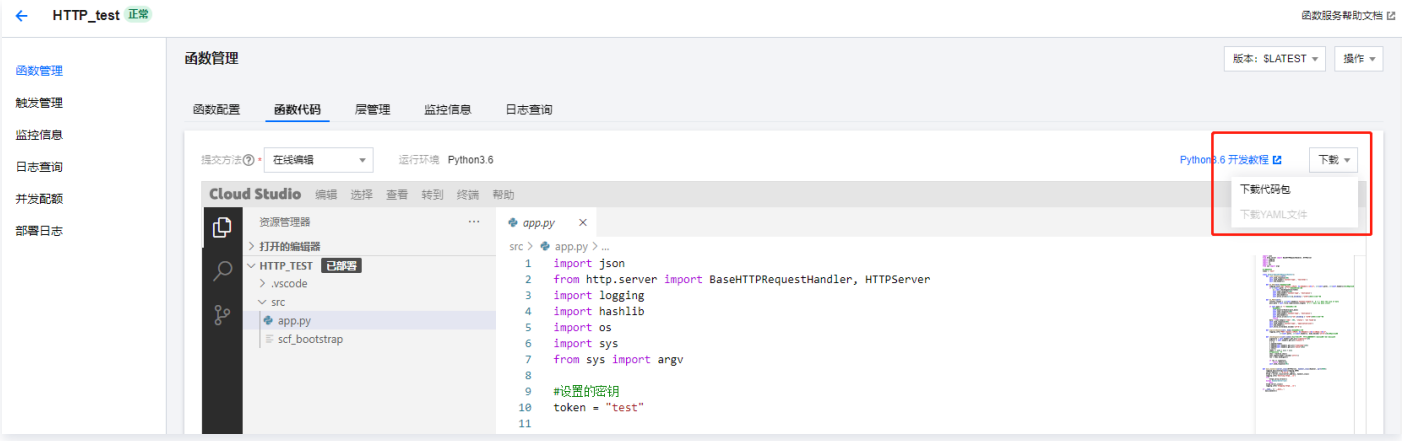


12.重新返回 云函数 界面，进入“日志查询”功能页面，可查看服务端接收到的由物联网开发平台同步的设备数据、上下线消息日志：





13.HTTP 接收示例代码可下载模板代码部署到用户自己的服务器上进行测试。



设备端与应用端信令交互说明

最近更新时间：2024-12-25 18:05:52

概述

底层应用端和设备端 P2P 传输通道建立后，将基于该通道实现 P2P 消息发送和音视频传输功能。下面将详细描述设备端与应用端的信令交互流程，分为音视频传输（设备推音视频流给应用端）、对讲音视频传输（应用端推音视频流给设备端，适用于语音对讲或音视频对讲）、接收信令（应用端发送消息给设备）、发送信令（设备发送信息给应用端）。

> **说明：**
> 提及参数为内部参数，请勿使用。

音视频传输

请求方法：get
请求路径： / ipc.flv

基本请求参数

_crypto

- 类型：可选。
- 可见性：外部参数。
- 描述：加密开关，值为 on 或 off，无此参数或不为 off 则默认开启加密。

示例：

```
_crypto=off
```

action

- 类型：必选。
- 可见性：外部参数。
- 描述：本次拉流请求的类型，有 live 直播，playback 回放，download 下载。

示例：

```
action=live
```

channel

- 类型：必选。
- 可见性：外部参数。
- 描述：本次拉流请求的通道号，单摄像头为0，多摄像头设备指的是第几个摄像头，nvr 设备指的是第几路数据，均从0开始计数。

示例：

```
channel=0
```

user_args

- 类型：可选。
- 可见性：外部参数。
- 描述：用户自定义参数，不宜过长，建议控制在1KB 以内，内容和格式不限，但不得出现非法 URL 字符。

示例：

```
user_args=this_is_user_args%E4%BD%A0%E5%A5%BD
```

requester

- 类型：可选（后续开发建议作为必选参数）。
- 可见性：外部参数。
- 描述：发起拉流请求的角色，不宜过长，建议控制在64字节以内。目前有这几种：app-android，app-ios，wxmp-android，wxmp-ios，server-voip，如有必要可扩展版本号等额外信息，例如：app-android-xiaomi，wxmp-ios-p2p-player-version123。

示例：

```
requester=app-android
```

直播请求

用于观看摄像头实时音视频内容，直播请求参数 action = live。

附加请求参数

quality

- 类型：必选。
- 可见性：外部参数。
- 描述：本次拉流请求的视频质量，有低中高三种，分别是：standard，high，super。

示例：

```
quality=standard
```

请求示例

```
http://ipc.p2p.com/ipc.flv?
action=live&channel=0&quality=high&_token=123456789&_peername=xxxxxxxx&requester=app-
android&req_id=xxxxxxxx&user_args=this_is_user_args%E4%BD%A0%E5%A5%BD
```

回放请求

用于回放摄像头SD 卡内录像，回放请求参数 action = playback。

附加请求参数

start_time

- 类型：必选。
- 可见性：外部参数。
- 描述：请求回放的开始时间，UNIX 秒级时间戳， 与 end_time 间隔建议在5秒以上，间隔过小可能导致索引错误。

示例：

```
start_time=123456789
```

end_time

- 类型：必选。
- 可见性：外部参数。
- 描述：请求回放的结束时间，UNIX 秒级时间戳， 与 start_time 间隔建议在5秒以上，间隔过小可能导致索引错误。

示例：

```
end_time=123456789
```

type

- 类型：可选。
- 可见性：外部参数。
- 描述：录像类型，可用于回放特定类型的录像，取值范围0到INT32_MAX，由数值含义由用户自定义，例如1为报警录像、2为人脸检测录像、3为H.264格式录像等，若未指定则默认为0。

示例：

```
type=1234
```

请求示例

```
http://ipc.p2p.com/ipc.flv?
action=playback&channel=0&start_time=1633060850&end_time=1633060880&type=1234&_token=123456789&_peername=x
xxxxxxx
```

下载请求

用于下载 SD 卡内文件，配合“查询文件列表”信令使用。
回放请求参数 action = download。

附加请求参数

file_name

- 类型：必选。
- 可见性：外部参数。
- 描述：下载的文件名，若包含文件路径请进行 URL 编码处理。

示例：

```
file_name=test_file.mp4
```

offset

- 类型：必选。
- 可见性：外部参数。
- 描述：下载文件的起始位置，用户可借助此参数开发断点续传功能。

示例：

```
offset=0
```

请求示例

```
http://ipc.p2p.com/ipc.flv?
action=download&channel=0&file_name=test_file.mp4&offset=0&_crypto=off&_peername=xxxxxxxx
```

对讲音视频传输

请求方法：post
请求路径： / voice

请求参数

基本请求参数

请参考“音视频传输 – 基本请求参数”中除 action，channel 外的其余参数。

附加请求参数

channel

- 类型：摄像头类设备可选，NVR 类设备必选。
- 可见性：外部参数。
- 描述：本次对讲请求的通道号，对于摄像头类设备此参数暂未使用，可省略；对于 NVR 类设备指的是第几路，从0开始计数。

示例：

```
channel=0
```

请求示例

```
http://ipc.p2p.com/voice?channel=0&_token=123456789&_peername=xxxxxxxx
```

接收信令

请求方法：post
请求路径： / command
默认开启加密，不可关闭。

请求参数

action

- 类型：必选。
- 可见性：外部参数。
- 描述： `inner_define` 为内部信令，请勿使用； `user_define` 为外部信令即用户自定义信令。

示例：

```
action=inner_define
action=user_define
```

channel

- 类型：摄像头类设备可选，NVR 类设备必选。
- 可见性：外部参数。
- 描述：本次对讲请求的通道号，对于摄像头类设备此参数暂未使用，可省略；对于 NVR 类设备指的是第几路，从0开始计数。

示例：

```
channel=0
```

cmd

- 类型：必选
- 可见性：外部参数
- 描述：消息类型

示例：

```
cmd=my_message
```

内部信令

本地录像回放相关信令

按起止时间查询录像

请求参数：

键	值	描述

cmd	get_record_index	信令类型。
start_time	UNIX 时间戳	查询录像的开始时间，单位为秒，与 end_time 间隔建议在5秒以上，间隔过小可能导致索引错误。
end_time	UNIX 时间戳	查询录像的结束时间，单位为秒，与 start_time 间隔建议在5秒以上，间隔过小可能导致索引错误。
type	枚举	表示录像的类型，是一个整数，由用户自己定义具体含义，目前暂不支持。

返回值：
json 格式的录像索引列表，基本格式如下：

```
{
    "video_list": [
        {
            "type": 0,
            "start_time": "<unix时间戳>",
            "end_time": "<unix时间戳>"
        },
        {
            "type": 0,
            "start_time": "<unix时间戳>",
            "end_time": "<unix时间戳>"
        }
    ]
}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=get_record_index&start_time=000&end_time=111
```

按月查询录像

请求参数：

键	值	描述
cmd	get_month_record	信令类型
time	时间	6位整数，前4位表示年，后两位表示月。

返回值：
json 格式，其中 `video_list` 的值转换为一个32位的整数，从低位到高位每一比特代表月份的第几天是否有录像；例如：8320（0010000010000000）表示8号和14号有录像。

```
{"video_list": "123456"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=get_month_record&time=202401
```

回放暂停

请求参数：

键	值	描述

cmd	playback_pause	信令类型

返回值：
json 格式， `<code>` **为信令执行结果返回值，0正常，其他异常。**

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_seek&progress=123456
```

回放继续

请求参数：

键	值	描述
cmd	playback_resume	信令类型

返回值：
json 格式， `<code>` **为信令执行结果返回值，0正常，其他异常。**

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_resume
```

设置播放位置

请求参数：

键	值	描述
cmd	playback_seek	信令类型
progress	整数	毫秒单位的时间戳，表示在当前文件内希望跳转播放的位置

返回值：
json 格式， `<code>` **为信令执行结果返回值，0正常，其他异常。**

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_seek&progress=123456
```

获取播放位置

请求参数：

键	值	描述
cmd	playback_progress	信令类型

返回值：

json 格式，`<code>`为信令执行结果返回值，0正常，其他异常；`<timestamp>`当前播放进度，单位毫秒。

```
{
    "status":"<code>",
    "progress":"<timestamp>"
}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_progress
```

设置快进

请求参数：

键	值	描述
cmd	playback_ff	信令类型
value	整数	• 0表示正常播放。 • 1表示只播放I帧。 • 2表示从相邻的两个I帧中取一个I帧播放。 • 3表示从相邻的三个I帧中取一个I帧播放。 以此类推

返回值：
json 格式，`<code>`为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_ff&value=1
```

设置快退

请求参数：

键	值	描述
cmd	playback_rewind	信令类型
start_time	时间戳	单位毫秒，表示倒放的结束时间
end_time	时间戳	单位毫秒，表示倒放的开始时间

返回值：
json 格式，`<code>`为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_rewind&start_time=100&end_time=200
```

设置倍速

请求参数：

键	值	描述
cmd	playback_speed	信令类型
speed	整数	单位毫秒，表示帧间隔，以视频为准进行同步，假设视频的正常帧率为25fps，即帧间隔是50ms，数值小于正常帧间隔则为快放。例如：设置为10则相当于5倍速快放；数值大于正常帧间隔则为慢放，例如设置为100则相当于0.5倍速慢放。

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=playback_speed&speed=50
```

TWeCall及语音对讲相关信令

挂断

> **说明：**
> 由 client 端发送给设备端，用于接听以后，表示主动挂断。

请求参数：

键	值	描述
cmd	call_hang_up	信令类型

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_hang_up
```

拒接

> **说明：**
> 由 client 端发送给设备端，用于接听以前，表示拒接。

请求参数：

键	值	描述
cmd	call_reject	信令类型

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_reject
```

取消

> **说明：**
> 由 client 端发送给设备端，用于接听以前，表示主动取消呼叫。

请求参数：

键	值	描述
cmd	call_cancel	信令类型

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_cancel
```

占线

> **说明：**
> 由 client 端发送给设备端，用于接听以后，表示有其他设备正在通话。

请求参数：

键	值	描述
cmd	call_busy	信令类型

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_busy
```

无应答

> **说明：**
> 由 client 端发送给设备端，用于接听以前，表示呼叫超时无人接听。

请求参数：

键	值	描述
cmd	call_timeout	信令类型

返回值：
json 格式，`<code>` 为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_timeout
```

对方关闭麦克风及摄像头

> **说明：**
> 由 client 端发送给设备端，表示对方关闭对方关闭麦克风及摄像头，设备端不会收到音视频数据。

请求参数：

键	值	描述
cmd	call_av_status_close	信令类型

返回值：
json格式，<code>为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_av_status_close
```

对方只开启麦克风

> **说明：**
> 由 client 端发送给设备端，表示对方只开启麦克风，设备端只能收到音频。

请求参数：

键	值	描述
cmd	call_av_status_audio_only	信令类型

返回值：
json格式，<code>为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_av_status_audio_only
```

对方只开启摄像头

> **说明：**
> 由 client 端发送给设备端，表示对方只开启摄像头，设备端只能收到视频。

请求参数：

键	值	描述
cmd	call_av_status_video_only	信令类型

返回值：
json格式，<code>为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_av_status_video_only
```

对方开启麦克风和摄像头

> **说明：**
> 由 client 端发送给设备端，表示对方开启麦克风和摄像头，设备端能收到音视频。

请求参数：

键	值	描述
cmd	call_av_status_audio_video	信令类型

返回值：
json格式，<code>为信令执行结果返回值，0正常，其他异常。

```
{"status":"<code>"}
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=call_av_status_audio_video
```

其他信令

查询设备状态

请求参数：

键	值	描述
cmd	get_device_st	信令类型
type	live / voice / playback	必选，询问设备是否接受直播、对讲、回放请求
quality	standard / high / super	type 为 live 时必选，询问设备是否支持低中高图像质量的直播请求

返回值：
json 格式，`status` 状态码，0接受连接请求，其他值拒绝连接请求；`appConnectNum` 当前已连接到设备的 APP 或小程序数；`maxConnectNum` 设备端支持的最大连接数。

```
[
    {
        "status":"code",
        "appConnectNum":"2",
        "maxConnectNum": "3"
    }
```

```
]
```

请求示例：

```
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=get_device_st&type=live&quality=standard
```

查询文件列表

请求参数：

键	值	描述
cmd	get_file_list	信令类型

返回值：
json 格式，`file_type` 文件类型；`file_name` 文件名；`file_size` 文件大小；`start_time` 开始时间；`end_time` 结束时间；`extra_info` 扩展信息，可用于携带哈希值或其他自定义文件信息。

```
{
    "file_list": [
        {
            "file_type": "video",
            "file_name": "1234.mp4",
            "file_size": "1024000",
            "start_time": "2024-01-23_12-34-00",
            "end_time": "2024-01-23_12-34-59",
            "extra_info": {
                "md5": "abvdefg"
            }
        },
        {
            "file_type": "picture",
            "file_name": "abcd.jpg",
            "file_size": "10240",
            "start_time": "2024-01-23_12-34-56",
            "end_time": "2024-01-23_12-34-56"
        }
    ]
}
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=get_file_list
```

查询NVR下设备名称

请求参数：

键	值	描述
cmd	get_nvr_list	信令类型

返回值：
json 格式，`DeviceName` 设备名；`Channel` 位于几通道；`Online` 是否在线。

```
[
    {
        "DeviceName": "name1",
```

```
        "Channel": "1",
        "Online": "0"
    },
    {
        "DeviceName": "name2",
        "Channel": "2",
        "Online": "1"
    }
]
```

请求示例：

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=inner_define&cmd=get_nvr_list
```

外部信令

即用户自定义信令
请求参数：

键	值	描述
cmd	信令内容	用户自定义信令，不宜过长，建议控制在1KB 以内，内容和格式不限，但不得出现非法 URL 字符

返回值：
格式不限，类型不限，不宜过长，建议控制在1KB 以内。
请求示例：

```
自定义信令进行云台控制示例：
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=user_define&cmd=ptz_left_speed_1
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxxx&action=user_define&cmd=ptz_right_speed_1
自定义信令传输少量二进制数据：
http://ipc.p2p.com/command?
_token=123456789&_peername=xxxxxxxx&action=user_define&cmd=36D7F336FCEC9E57318ECA0323BA94970296A776CE028D1
9F5CA5AC44E92952B4CE77642354CBAE7036F4DA954317462C61A1BC033DAC882FAAB9EDCBFEE47DA
```

发送信令

请求方法：post
请求路径： / feedback
默认开启加密，不可关闭。消息在 http 报文正文中发送，采用 json 格式。

请求参数

键	值	描述
product_id	设备三元组信息	供对方识别设备、消息管理等场景使用
device_name	设备三元组信息	供对方识别设备、消息管理等场景使用

请求示例

```
http://127.0.0.1:34567/ipc.p2p.com/forward/xxxxxxxxxxxxx/proxy.sample.server/feedback?
_token=123456789&product_id=xxxxxxxx&device_name=xxxxxxxx
```

用户自定义信令

用户自定义消息格式采用 json 格式，不得含有 `iv_private_cmd` 字段，其余内容不限，长度不宜过大，建议控制在16KB 以内。
示例：

```
发送
{
"my_message":"hello world"
}
接收
{
"status":"<code>"
}
```

`<code>` 为返回值，0正常，其他异常。

内部信令

内部消息使用以下 json 格式。

```
发送
{"iv_private_cmd":"<cmd>"}
接收
{"status":"<code>"}
```

挂断

由设备端发送给 client 端，用于接听以后，表示主动挂断。

```
{"iv_private_cmd":"call_hang_up"}
```

响应：

```
{"status":"<code>"}
```

`<code>` 为返回值，0正常，其他异常。

取消

由设备端发送给 client 端，用于接听以前，表示主动取消呼叫。

```
{"iv_private_cmd":"call_cancel"}
```

响应：

```
{"status":"<code>"}
```

`<code>` 为返回值，0正常，其他异常。

其他信令

设备端主动停止接收对端发送的音视频数据

由设备端发送给 client 端，用于双向对讲的情景。

```
{"iv_private_cmd":"recv_stop"}
```

响应：

```
{"status":"<code>"}
```

`<code>` 为返回值，0正常，其他异常。

腾讯连连小程序交互信令说明

最近更新时间：2024-12-02 15:17:33

设备端对接腾讯连连小程序 IoT Video 标准面板时使用的信令交互协议。

请求方法

```
/**
 * command 为字符串或对象
 */
function sendCommand(command) {
    message_id++
    if (typeof command !== 'string') command = 'action=user_define&channel=0&cmd=' +
JSON.stringify(command)
    return p2pExports.sendCommand(id, command)
}
```

内部信令

参考文档 设备端与应用端信令交互说明_内部信令 。

外部信令

操作云台

请求字段

字段	类型	说明
topic	string	ptz 操作类型。
message_id	number | string	请求 ID。
cmd	string	• ptz_release_pre：松手 。 • ptz_up_press：上 。 • ptz_right_press：右 。 • ptz_down_press：下 。 • ptz_left_press：左。

请求示例

```
{
    "topic": "ptz",
    "message_id": 0,
    "data": {
        "cmd": "ptz_release_pre"
    }
}
```

返回字段

字段	类型	说明
apex	string	是否到顶。 • yes：到顶。 • no：未到顶。
current_x	number	横向位置。

current_y	number	纵向位置。
max_x	number	横向位置最大值。
max_y	number	纵向位置最大值。
min_x	number	横向位置最小值。
min_y	number	纵向位置最小值。

返回示例

```
{
    "confirmation_topic": "ptz",
    "result": "0",
    "apex": "yes",
      "current_x": 123,
      "current_y": 123,
      "max_x": 123,
      "max_y": 123,
      "min_x": 123,
      "min_y": 123,
    "message_id": 0
}
```