

物联网智能视频服务（消费版）

开发者手册



腾讯云

【 版权声明 】

©2013–2024 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的内容。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 】

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

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。

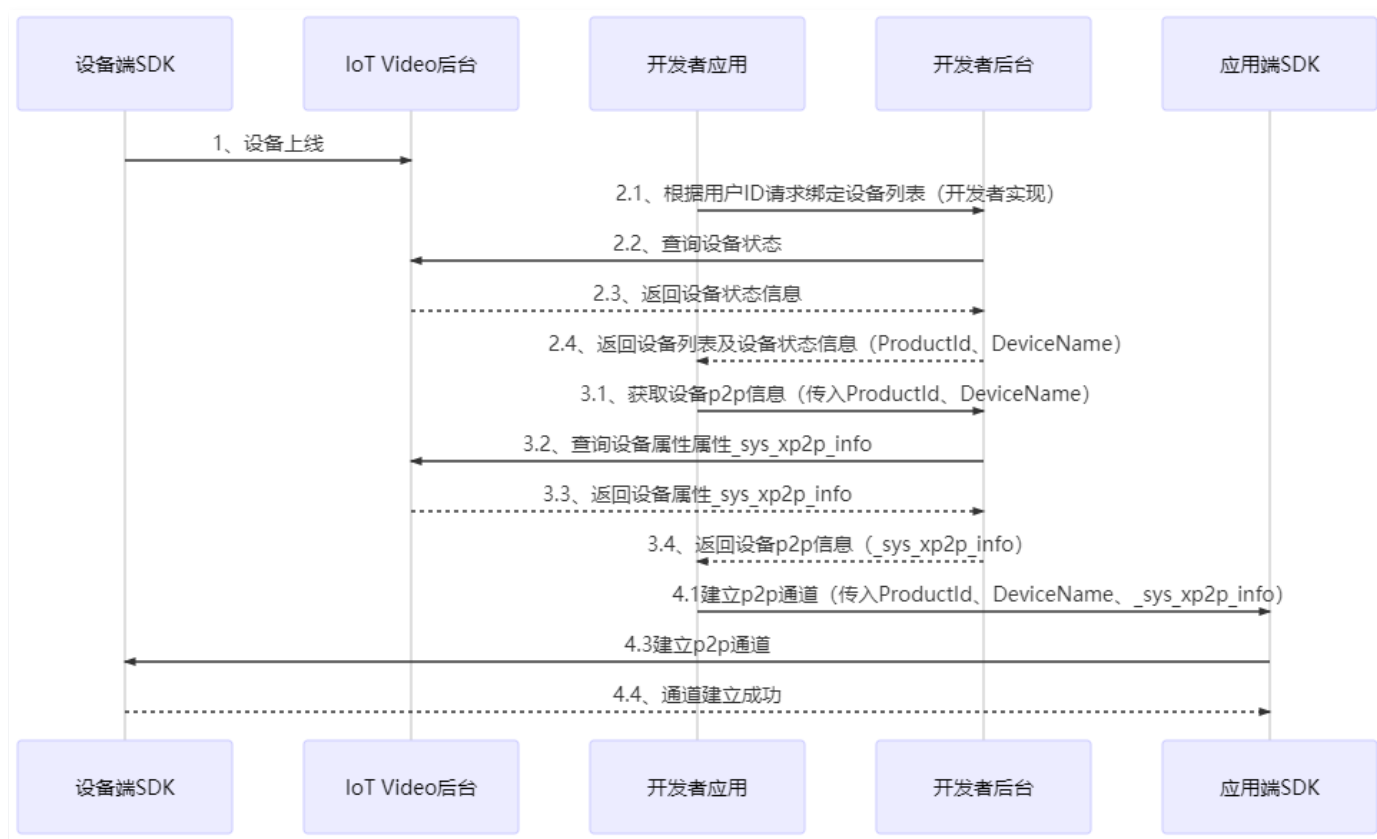
功能概述

- 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

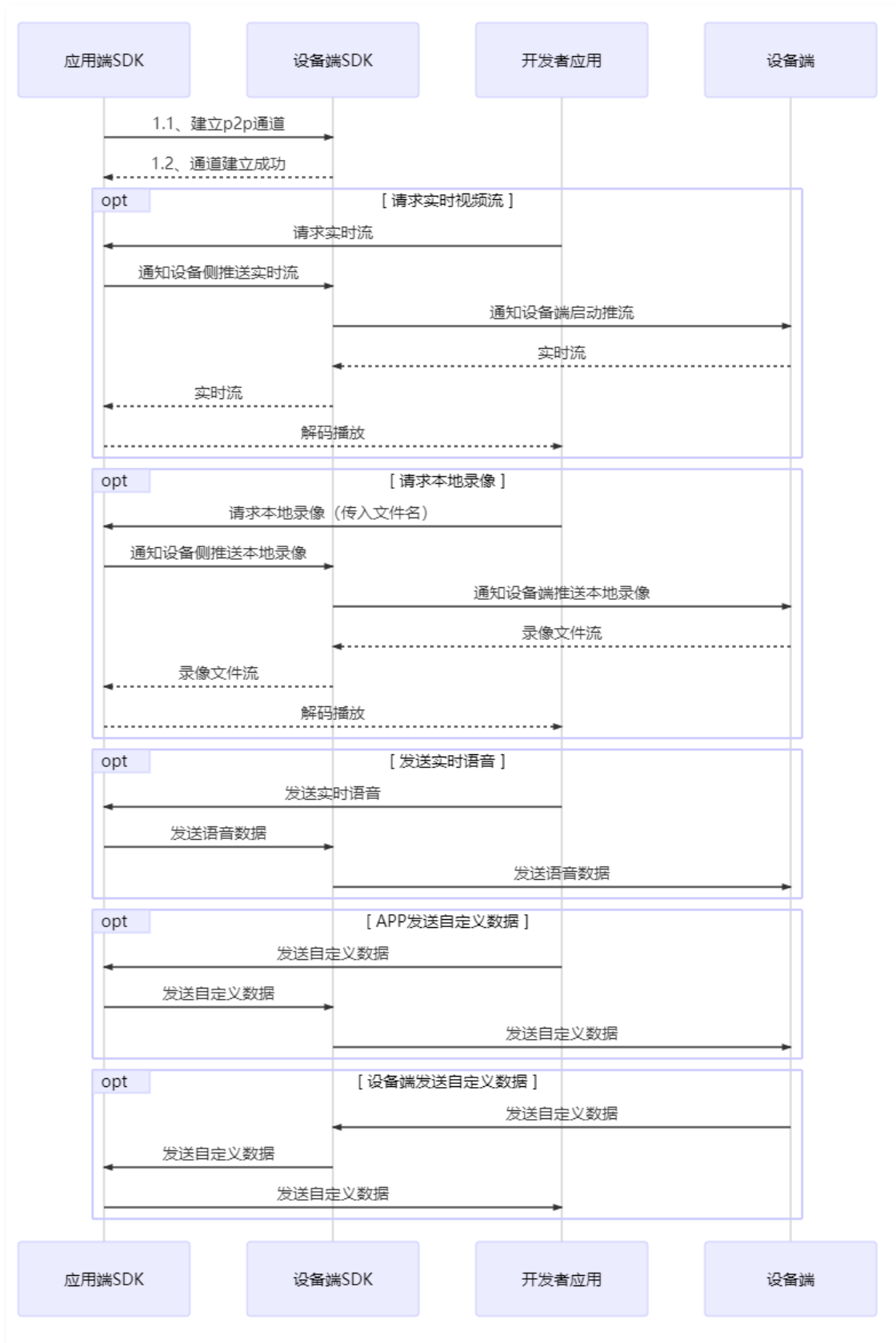
建立 P2P 连接

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

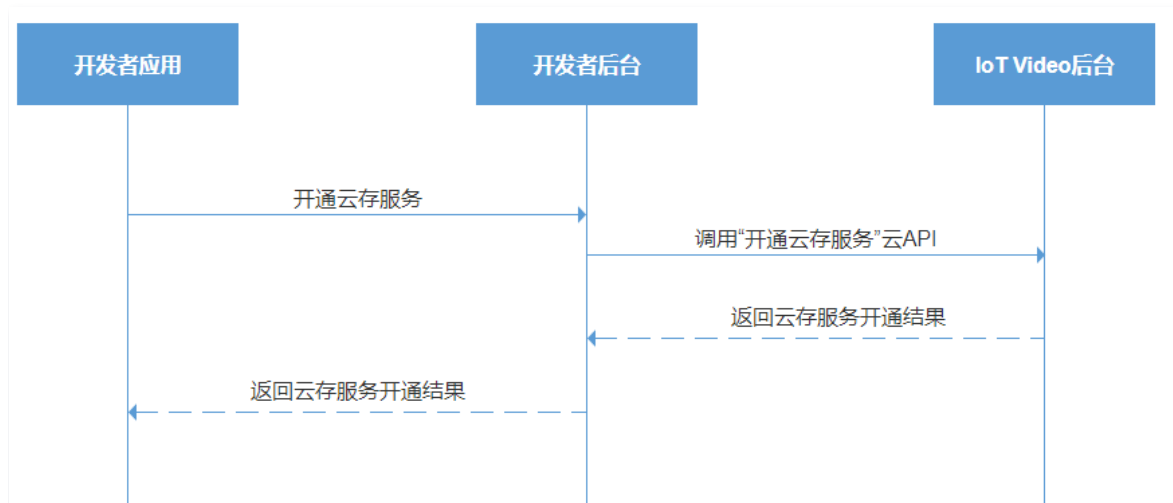
最近更新时间：2022-09-02 16:40:02

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）

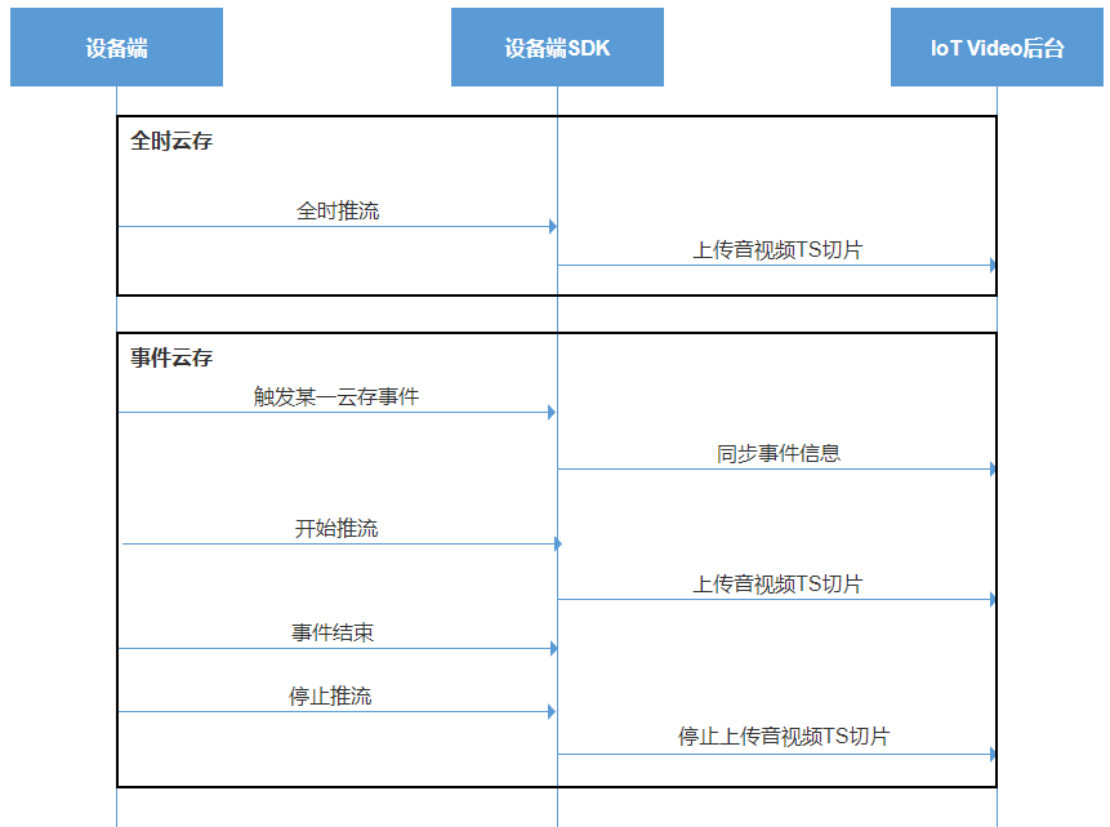
云存套餐分为全时云存套餐和事件云存套餐：

- **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。

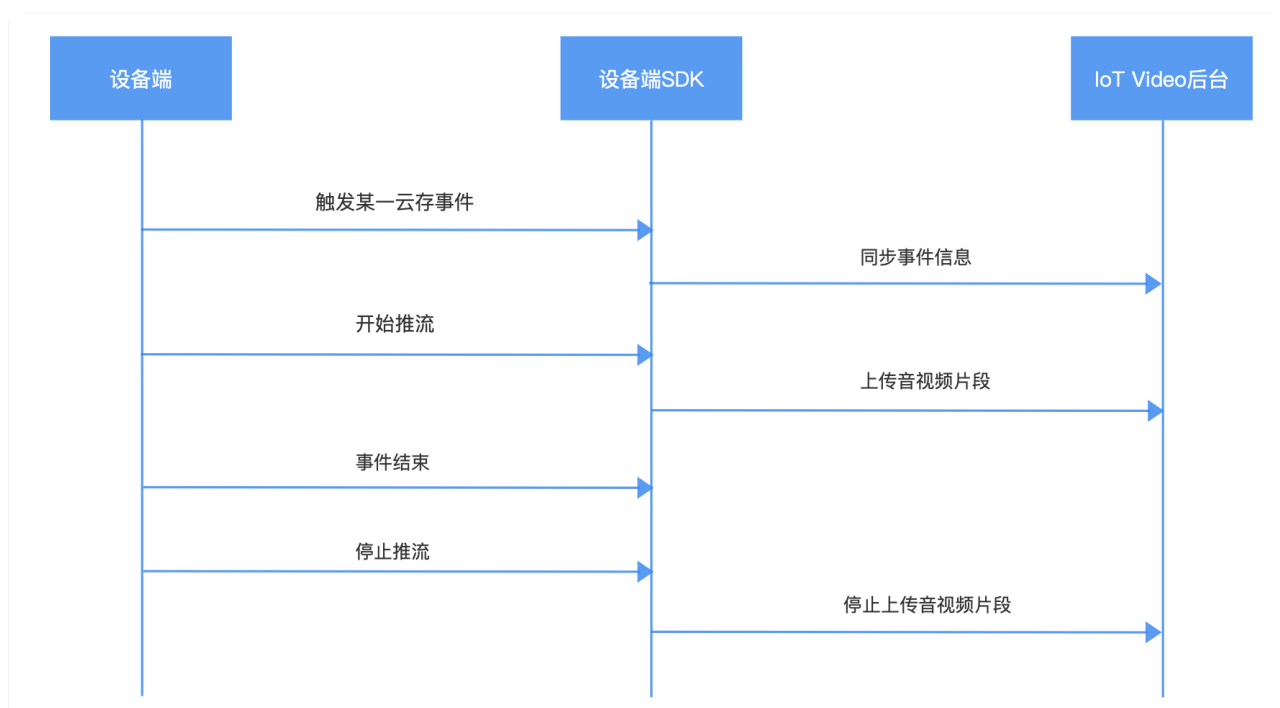
云存录像上传（图片流设备）

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。

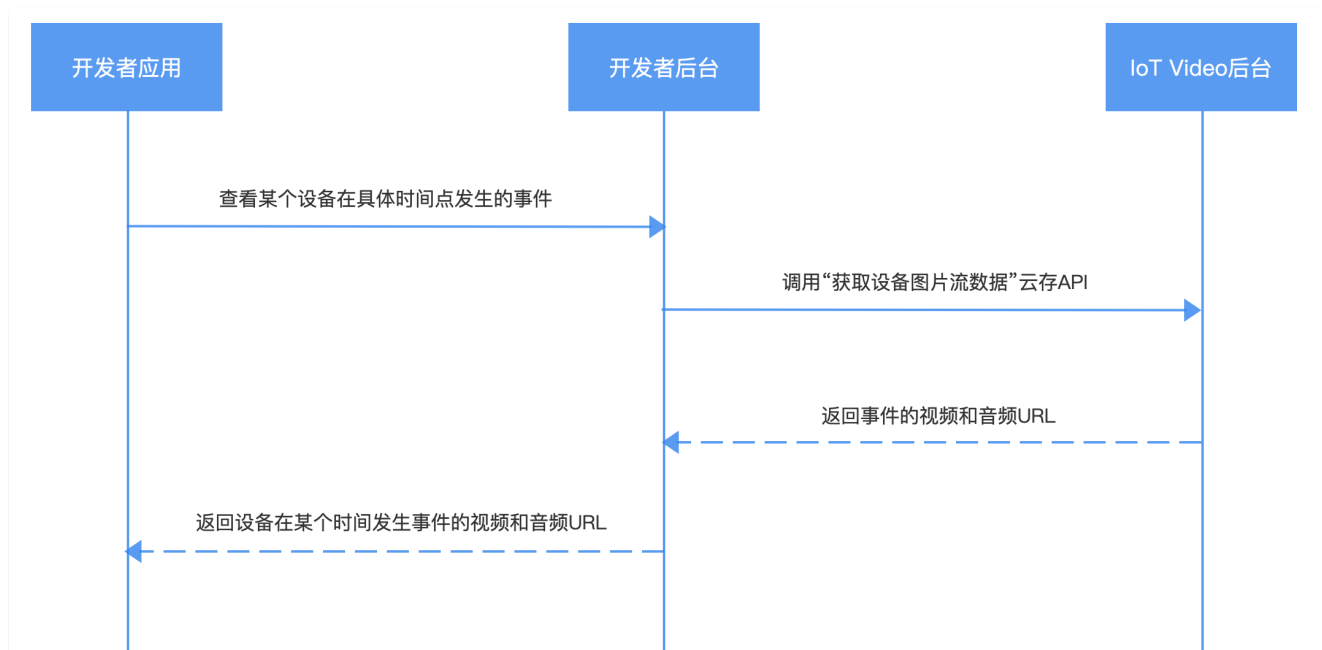
说明：

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：2022-11-04 11:00:44

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，详情参加控制台指引 > [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 <code>\${ProductId}/\${DeviceName}/data</code> 。
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。

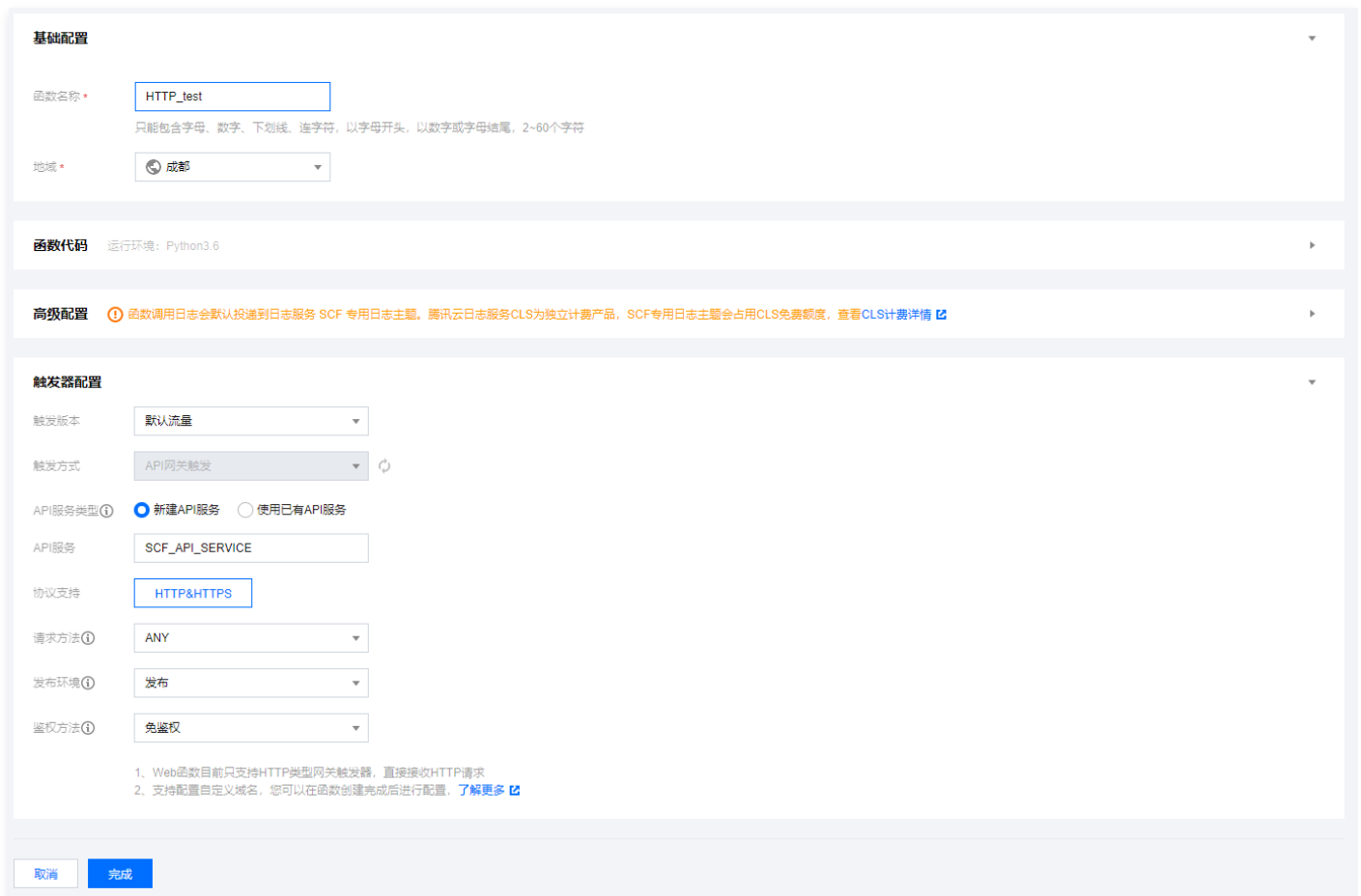


5. 创建方式选择**模板创建**，在下方“模糊搜索栏”搜索 `IoT Explorer`。

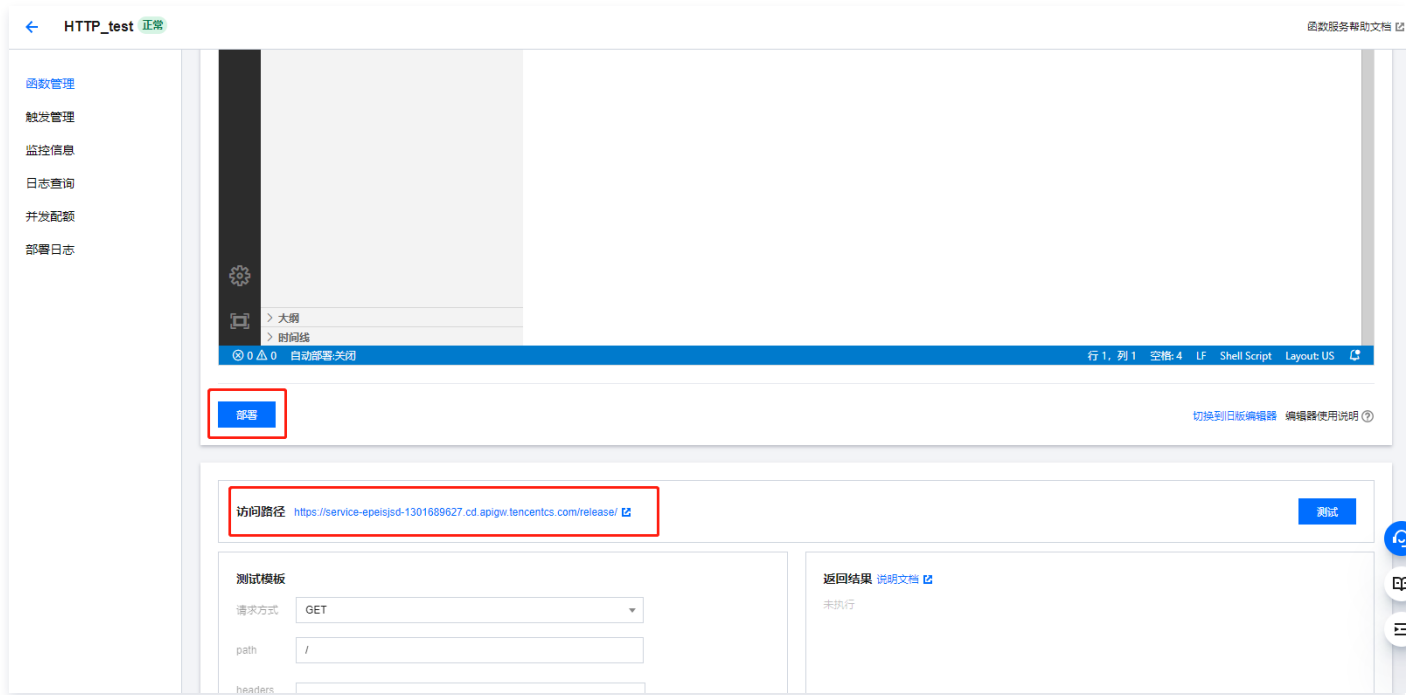


6. 选择需要语言的“IoT Explorer数据同步转发”模板后，单击下一步按钮。

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。

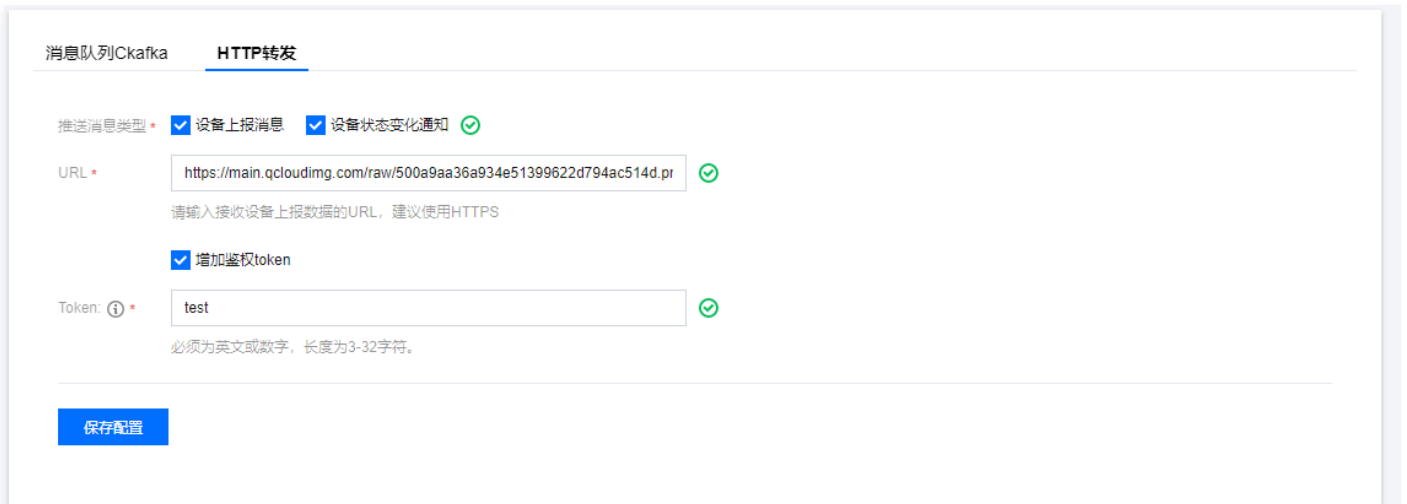


8. 在“函数管理”页面选择函数代码功能栏，下滑单击部署按钮。



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。

```

1 import json
2 from http.server import BaseHTTPRequestHandler, HTTPServer
3 import logging
4 import hashlib
5 import os
6 import sys
7 from sys import argv
8
9 #设置的密钥
10 token = "test"
11
12 class Handler(BaseHTTPRequestHandler):
13     def __init__(self):
14         self.send_response(200)
15         self.send_header('Content-type', 'text/html')
16         self.end_headers()
17
18     def do_GET(self):#处理get请求
19         logging.info("GET request,\nPath: %s\nHeaders:\n%s\n", str(self.path), str(self.headers))#打印
20         if str(self.path) == "/":#判断URL的路径
21             a = self.checkSignature(token)
22             self.send_response(200)
23             self.send_header('Content-type', 'text/plain')
24             self.end_headers()
25             self.wfile.write(bytes(a,encoding = "utf8"))#返回验证参数
26
27     def do_POST(self):
28         content_length = int(self.headers['Content-Length']) # <--- Gets the size of data
29         post_data = self.rfile.read(content_length) # <--- Gets the data itself
30
31         if self.path == "/":#判断URL路径
32             #处理数据
33             self.explorerHandle(post_data)
34             self.send_response(200)
35             self.send_header('Content-type', 'text/plain')
36             self.end_headers()

```

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。

消息队列Ckafka		HTTP转发	开启 编辑
推送消息类型	设备上报消息、设备状态变化通知		
URL	https://main.qcloudimg.com/raw/500a9aa36a934e51399622d794ac514d.png		
状态	未启用		

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：

函数管理

函数配置 函数代码 部署管理 监控信息 日志查询

调用日志 高级检索

全部日志 今天 2021-09-17 00:00:00 - 2021-09-17 23:59:59 刷新

请求ID: 7e7e2466c5858a71da8f001cb4504169

时间: 2021-09-17 19:02:58 运行时间: 4.412ms 运行内存: 11.2578125MB

END RequestId: 7e7e2466c5858a71da8f001cb4504169

Report RequestId: 7e7e2466c5858a71da8f001cb4504169 Duration: 7ms Memory: 128MB MemUsage: 11.26MB

INFO:root:POST request,

Path: /

Headers:

Host: service-epejsd-1301689627.cd.apigw.tencentcs.com

User-Agent: Go-http-client/1.1

Content-Length: 368

Accept-Encoding: gzip

Content-Type: application/json

Nonce: XGCGVWYXqy9P9uS

Signature: 3d1a0f15b548b2525b15967956e22f213767

Timestamp: 1631876578

X-Api-RequestId: 7e7e2466c5858a71da8f001cb4504169

X-Client-Proto: https

X-Client-Proto-Version: HTTP/1.1

X-Forwarded-For: 42.193.134.62

X-Real-IP: 169.254.128.157

X-Sc-AppId: 1301689627

X-Sc-Memory: 128

X-Sc-Name: HTTP_test

X-Sc-NameSpace: default

X-Sc-Region: ap-chengdu

X-Sc-Request-Id: 7e7e2466c5858a71da8f001cb4504169

X-Sc-Timeout: 3000

X-Sc-Uin: 100013575134

X-Sc-Version: SLATEST

Connection: close

平台转发的设备离线信息

Body:

```
{
  "payload": {
    ["deviceName": "1_54063850_1", "event": "EV_ONLINE", "productId": "UFTL8PSPBW", "reason": "REASON_DEVICE_CONNECT", "timestamp": 1631876578, "topic": "State/report/UFTL8PSPBW_1_54063850_1", "timeMillis": 1631876578145, "seq": 631412, "timestamp": 1631876578, "topic": "State/report/UFTL8PSPBW_1_54063850_1", "deviceName": "1_54063850_1", "productId": "UFTL8PSPBW"}
  }
}
```

203.0.113.1 - [17/Sep/2021 11:03:02] "POST / HTTP/1.1" 200 -

函数管理

函数配置 函数代码 部署管理 监控信息 日志查询

调用日志 高级检索

全部日志 今天 2021-09-17 00:00:00 - 2021-09-17 23:59:59 刷新

请求ID: 7c25aaee4e1b710ae7d533098849c979

时间: 2021-09-17 19:03:02 运行时间: 1ms 运行内存: 40.88328125MB

START RequestId: 7c25aaee4e1b710ae7d533098849c979

END RequestId: 7c25aaee4e1b710ae7d533098849c979

Report RequestId: 7c25aaee4e1b710ae7d533098849c979 Duration: 1ms Memory: 128MB MemUsage: 40.86MB

INFO:root:POST request,

Path: /

Headers:

Host: service-epejsd-1301689627.cd.apigw.tencentcs.com

User-Agent: Go-http-client/1.1

Content-Length: 316

Accept-Encoding: gzip

Content-Type: application/json

Nonce: nUXWhmLVFYQndsyu

Signature: 9f47938fce1eb6924406cb444c8f89dc6c6b6e7

Timestamp: 1631876582

X-Api-RequestId: 7c25aaee4e1b710ae7d533098849c979

X-Client-Proto: https

X-Client-Proto-Version: HTTP/1.1

X-Forwarded-For: 42.193.134.62

X-Real-IP: 169.254.128.5

X-Sc-AppId: 1301689627

X-Sc-Memory: 128

X-Sc-Name: HTTP_test

X-Sc-NameSpace: default

X-Sc-Region: ap-chengdu

X-Sc-Request-Id: 7c25aaee4e1b710ae7d533098849c979

X-Sc-Timeout: 3000

X-Sc-Uin: 100013575134

X-Sc-Version: SLATEST

Connection: close

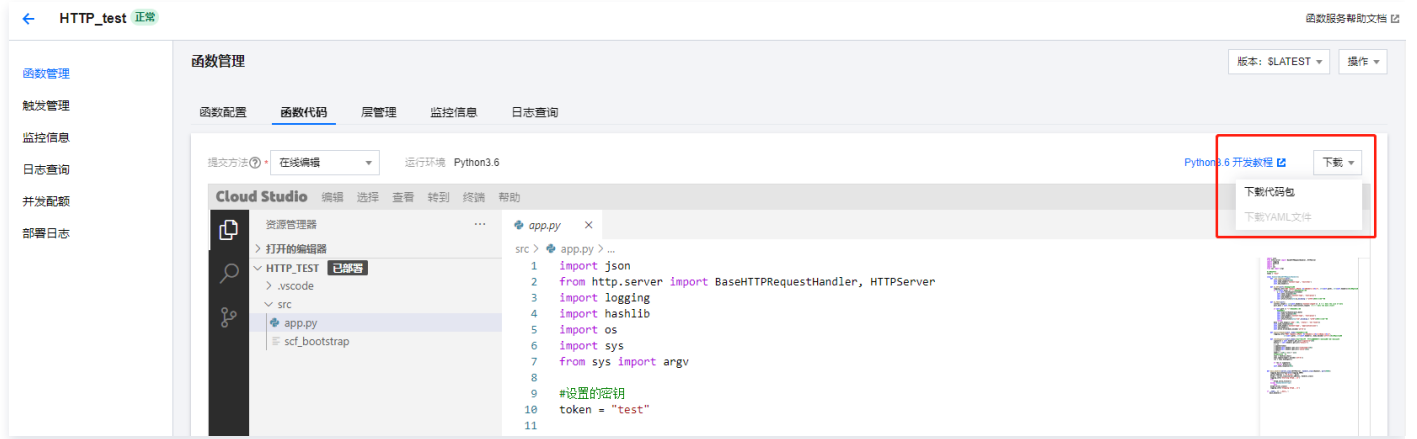
平台转发的设备数据信息

Body:

```
{
  "payload": {
    ["clientToken": "UFTL8PSPBW-3", "method": "report", "params": {
      ["dev_info": {
        ["audio_codec": "AAC", "video_codec": "H264", "record_enable": 0],
        "timeMillis": 1631876582517, "seq": 631581, "timestamp": 1631876582, "topic": "Shing/up/property/UFTL8PSPBW_1_54063850_1", "deviceName": "1_54063850_1", "productId": "UFTL8PSPBW"}
      }
    }
  }
}
```

203.0.113.1 - [17/Sep/2021 11:03:02] "POST / HTTP/1.1" 200 -

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：2024-01-02 10:33:21

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：未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。目前有以下几种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，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=xxxxxxx&requester=app-android&req_id=xxxxxxx&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
```

请求示例

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

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=xxxxxxx
```

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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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1帧，2表示从相邻的两个1帧中取一个1帧播放，3表示从相邻的三个1帧中取一个1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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&action=inner_define&cmd=call_timeout
```

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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&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=xxxxxxx&action=user_define&cmd=ptz_left_speed_1
```

```
http://ipc.p2p.com/command?_token=123456789&_peername=xxxxxxx&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=xxxxxxx&action=user_define&cmd=36D7F336FCEC9E57318ECA0323BA94970296A776CE
```

```
028D19F5CA5AC44E92952B4CE77642354CBAE7036F4DA954317462C61A1BC033DAC882FAAB9EDCBFEE47DA
```

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/xxxxxxxxxxxx/proxy.sample.server/feedback?
```

```
_token=123456789&product_id=xxxxxxx&device_name=xxxxxxx
```

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

说明：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

说明：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**

说明：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：2022-06-09 11:30:15

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	<ul style="list-style-type: none"> 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。 <ul style="list-style-type: none"> 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
}
```