

# 物联网智能视频服务(消费版) 应用端接入手册





【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

应用端接入手册 低代码小程序应用端接入手册 概述 准备工作 授权部署发布小程序 设备端接入(摄像机) 设备端接入(图片流门锁) 设备端接入(视频流门锁) 低代码小程序蓝牙配网指南 音视频模块 APP SDK概述 Android 应用端接入手册 快速开始 iOS 应用端接入手册 快速开始 错误码 小程序应用端接入手册 物联网模块 APP 应用端接入手册 小程序应用端接入手册 应用端 API 列表



# 应用端接入手册 低代码小程序应用端接入手册 概述

最近更新时间: 2022-11-17 12:50:16

搭建低代码的自主品牌小程序工作主要分为3个部分:准备工作、授权发布小程序、设备端功能开发。

## 流程图



# 流程简介

## 准备工作

作为由 loT Video 代开发的小程序,其所有权归属于注册小程序的企业。因此需要登录 微信公众平台 注册小程序账号,并完成相关的配置后,开通订阅消 息以实现微信消息强提醒推送。

#### 授权发布小程序

在完成准备工作后,用户可在 loT Video 消费版控制台 上创建产品、添加设备,并在应用开发模块将注册小程序授权给 loT Video,然后配置强提醒消 息等信息后,即可体验生成的小程序,如满足要求则可提交微信审核并正式发布小程序。

## 设备端接入

在获得小程序后,则可根据在控制台添加设备得到的设备三元组信息对提供的相关版本设备端 SDK 进行相关功能的适配和调试。目前支持摄像机、图片流 和视频流门锁设备。



# 准备工作

最近更新时间: 2024-09-06 16:43:51

# 1. 注册微信小程序

访问小程序注册,按指引完成微信小程序账号注册。

	① 帐号信息 — ② 邮箱激活 — ③	)信息登记
每个邮箱(	双能申请一个小程序	已有微信小程序?立即登录
邮箱	请输入正确的邮箱地址 作为登录纸号,请项写未被做信公众平台注册,未被做信开放平台注册,未被个人做信号 频定的邮箱	小程序上线需要开发 不会开发? 找小程序代开发服务 学习开发课程。前往做信学室 创建测试号、免注册快速体验小程序开 发。立即申请
密码	请填写密码 字母、数字或者英文符号,最短8位。区分大小写	
确认密码	请再次输入密码	
验证码	しんがわ	
	你已阅读并同意《微信公众平台服务协议》及《微信小程序平台服务条款》 注册	

# 2. 设备接入小程序

需要确保小程序主体公司的营业执照经营范围含有**智能硬件销售及服务**或**物联网设备销售及服务**等描述,并且在国家工商官网可查询到,否则无法通过微信 审核。

# 2.1 添加服务类目

登录 小程序管理后台,选择左侧最下方**设置 > 基本设置 > 基本信息 > 服务类目**后,点击右侧**详情**,进入下方页面。添加"工具 > 设备管理"类目(该类目 无需资质,如已有该类目可忽略此流程。)

设置		
基本设置 第三方设置 关联设置 关注公众号		
基本设置 / 服务类目		
<b>服务类目</b> 服务类目最多添加5个,本月可添加5次,联系客服		设置主营类目 添加类目
服务类目	状态	操作
工具 > 设备管理 主营	已通过	删除
工具 > 视频客服	已通过	删除

# 2.2 微信认证

#### △ 注意:

完成微信认证是开通设备管理的必要操作,请务必先完成微信认证后,再申请添加硬件设备。

1. 签署《微信公众平台认证服务协议》,勾选同意,单击**下一步**。



① 同意的成 — ② 描写资料 — ③ 确认名称 — ④ 描写边景 — ⑤ 支付费用 联系家根
双股协调用做得公众平台!
申请你曾认证是一项严肃的法律行为,找从证主体持此申请你曾(公众号/14组序(开放平台) 桥号(原始10) 认证服务,已行通阅读并同意以下内容及《你曾公众平台认证服务协议》。
1.在米号资质重线成功之后,帐号的使用仅属于通过资质重构的执认还主体,该帐号自注册时其产生的一切权利义务均由设主体理组,该帐号所获得的所有收益,权限均但认证后的主体享有,且所有运营活动能必须以该主体对外开展。
2 然以证主体理改组编书的认证资料真实无畏,并不可能确地投机的展现及其更任的任何属三方审核机构的理论的资料出行预测标实。一经申请和产生编制及其更任的属三方审核机物的审规成本,故所改成的认证审核服务费用库不因认证结果, 然以证主体是否接出规定申请等回款而通图。
3.8以证主体在申请附留以证股税过程中编码并附随而建改的认证联系人(包括后端不可支置的人员)为6以认证主体检查的认证联系人员,并特出最权用款指定人员以6以认证主体的名义负责该的指帐号的内容编中、开放编中及运营管理。认证 联系人的所销量(时方,均代期以以证主体,拟认证主体均需资程一切责任。
4、本认证医务仅对这种考销是在公认证资料的复杂性。会社性担行非互联股损失,其功能,仍就是否开通,称号给百次市等均需者守力应业务平台为货将用运的专项规则(如期借公众号示着盾(回省公众平台服务协议); 即置引程序标号质量 價(回省公众平台服务协议)及(回答:程序平台服务条款); 即提开放平台所受需量商(回答开放平台开发者服务协议)), 而不与认证率构成局界存在重要关系。
5.秋从证主体改造板上达影谱,均需得超一切展任、对确讯成属三方造成组织的,须题做提供,在初始国际集团新给从证服务之前,本公园持续有效。
■ 我已同意并善守上述内容及《物理公众于会认证服务协议》
<b>下一步</b>

2. 填写资料:选择认证主体类型,提交相应的认证材料。

 发票填写:目前只支持电子发票和纸质增值税专用发票。其中增值税专用发票还需提交《税务登记证》(办理三证合一的企业直接上传新的营业执照)、 《银行开户证明》,审核公司会对资质进行审核。资质审核通过后由腾讯公司开具并寄送发票。

#### △ 注意:

- 发票资料提交后不能修改,请填写正确的发票类型和寄送地址,若填写错误造成发票开具错误、寄送错误或选择不开具发票,后续将无法重新开具并寄送发票。
- 增值税专用发票的抬头为认证申请机构的全称,电子发票内暂不支持开具地址、电话、开户行及账号信息,腾讯云目前开具的发票也是可以
   正常报销的,如您需要开具以上信息,建议后续选择纸质增值税专用发票。
- 订单完成后(包括认证成功和失败),腾讯会在认证审核完成后8个工作日左右开具电子发票,30个工作日开具并寄出增值税专用发票。
- 4. 支付费用: 支付方式目前仅支持微信支付。

#### 2.3 开通设备管理功能

进入小程序管理后台的**功能 > 硬件设备**页面,阅读设备使用条件和接入流程等后,单击**开通**并由小程序管理员扫码确认后,显示开通成功,即可进入设备管 理页面 。

J	小程序									文档	社区~	We分析	工具~	<b>(14</b> )	$\Phi^{-1}$
+	首页	设备	管理												
	管理														
	版本管理 成员管理 用户反馈		硬件	设备											
¢	统计		PCTT												
	功能		功能说明 为用户提 序设备消	 供设备联网、设备技 息等。	空制等智能硬件相9	关服务的小程序,	可申请开通「硬	牛设备」,开道后	,可基于实际诉求	中请平台提供	的小程序硬件都	氻,如:小程			
	城市服务		使用条件												
	微信技一提 微信支付 物流服务		<ul> <li>必须是</li> <li>面向智</li> <li>小程序</li> <li>设音需</li> </ul>	通过微信认证的小程 能硬件生产企业或开 类目需为"工具一设f 要接入微信ID件系	IFF账号 F发音 备管理"共日										
	硬件设备 客服		接入流程 添加"工」	儿一设备管理"类日	漆加设备并获得	得设备ID申讨	青设备能力——开	发调试发布上	6						
	订阅消息 旧订阅消息		п	P											
	直播		217.8	2											
	页面内容接入														

## 2.4 添加硬件设备

进入**硬件设备**,单击**添加设备**,按照每个字段对应的说明填写设备相关信息,如门锁品类,则可添加一个门锁类的硬件设备。此步骤需要微信审核,审核时 间为1天 – 3天。



✔ 小程序				文档	社区~	服务 We	析 工具-	۵	0
	晋记设备								
合 首页	1000								
0.992									
		品牌名称 (中文)	101512.00233338.000 0354						
52-F 52-19		(第2)							
国産業権									
		生产广商 (中文)	说意的法则可能的 0.84						
a 功能									
REE101-W/F		生产广奏 (英文)	0,64						
劳信我一想		产品6称 (中文)							
器装支付			WARDERARPOAR BUARA						
指統订单			产品包裹上的名称一致,不要这长。						
物此服务		B B S D (1977)							
使作证备 + 22		PROMIN UPAU							
R.R. Controll		말만	0/84						
10000 1000 1000 1000 1000 1000 1000 10			10不同想点导致的型号不一样,兼算校						
实置内容统入			小、大胃腫肪情耳起号,因因每个型号部 菌量单性中国modeLid						
小程序描件									
SHIA		PARA	252*252px;近转底, prg格式;流产格 压模此格式上传影片;该图片为展示绘用						
小程序取图			户的图片,不能且现与产品无关信息。						
N 1749			+						
- 7164			上特型片						
ff发展1度									
HX1A									
1.007 022104		产品简合							
* 成长									
小理序注册			6/500						
法规记录			(The research						
d 10.0		DOD-REAL	CID + HARRY						

# 2.5 获取设备 model\_id

设备添加成功后,可在**设备管理**的列表中,获得平台分配的 model\_id , model\_id 是调用小程序设备能力相关接口的重要凭证。获取 model\_id 后,小 程序须为设备申请**消息能力**等设备能力。此步骤需要微信审核,审核时间为1天 - 3天。

设备	管理					帮助指引
						添加设备
	设备类型	设备型号	model_id	设备能力 ?	状态	操作
	家居安防-摄像头	高清云台2K版	d7	-	设备能力审核中 ⊘	
	家居安防-摄像头	云台2K版	zlk	-	设备信息未通过 🥜	删除 详情
	家居安防-门锁	K20 Pro	рХ	消息能力	已通过	删除 申请设备能力

# 3. 获取消息通知模板 ID

1. 登录小程序管理后台**功能 > 订阅消息 > 公共模板库 > 长期订阅**,查看可选用的设备消息模板。

Introduction         Status         Status <thstatus< th="">         Status         <thsta< th=""><th>▶ 小程序</th><th></th><th>文档 社区 &gt; 服务 We分t</th><th>ff 工具~ 〇</th></thsta<></thstatus<>	▶ 小程序		文档 社区 > 服务 We分t	ff 工具~ 〇
Pire         Control (Control (Contro) (Control (Contro) (Control (Control (Control (Con	) 首页			
Alia da	管理	3255/98/20 223+21227 年181277 第171277		
KH         CMB100         CMB100 <thcmb100< th=""> <thcmb100< th=""></thcmb100<></thcmb100<>	成员管理用户反馈	公共模板库 你可从下方的公共模板里中选用你所需的模板		
功能         加加	▶ 统计	<b>太性订阅</b> <-	提索模板	Q.
	1 功能 <sub>附近的小程序</sub>	摄像头盆崩异常报誓 报繁时间,所在位置、耀示说明	设备管理	选用
電気協分         単環境のなり構成         単原型の内し、作法10点、単の化物         単原型         2011           適合         「10分り30歳         「10分り30歳         必要者で、         26者管理         26者         26者         26者         26者         26         26         2	衛信提一握 衛信支付		20.60 MPTB	14.77
著作 (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2) (1)(2)(2)(2) (1)(2)(2)(2)(2) (1)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)(2)	物流服务 硬件设备	11396-453/9540388 金3/9893/99, 711年112,28, 38(7)15493	32(bit 10:746	12.41
页面内积极入 小程序邮件	客服 订阅消息	门铃锣叫耀耀 呼叫时间,继示说明	设备管理	逃用
	页面内容接入 小程序插件			

2. 单击需要接入模板右侧的**选用**,则可以进入具体页面,选择设备消息模板中需要的关键词,并提交。



#### 设备消息模板的关键词内容由平台生成,为枚举值,开发者不能够自定义内容。

עטרדע שי				
你可以在同一个公共模板下,使用不同的关	键词。提交启	5,关键词的内容和)	原序将无法修改。	
p2p player		关键词	呼叫时间	
门铃呼叫提醒			提示说明	
		已选择(0/5)	请先从上方选择关键词	
		场景说明	说明订阅消息服务场景	0/15
			提交	
查看详情	>			

#### 3. 提交后,可在我的模板中找到对应模板的模板 ID ,每个模板以 template\_id 标记。

♠ 首页	若启用小程序消息推送,用户右	E手机客户端使用订阅消息功能的相关事	B件,将会推送至所配置的服务器地址。宣看 消息推送配	π.			
	订阅消息						
🗅 管理							
版本管理	我的模板 公共模板库 审核	记录 操作记录					
成员管理							
用户反馈	我的模板						
C 统计	至效中 2 已删除 2			一次性订	1闾可选用50个,	长期订阅可选用25个	送用
<b>…</b> 功能	标题	关键词	模板ID		类型	操作人	操作
開始に行われた第日の学							
的加速一個	门铃野叫提醒	呼叫时间、提示说明	ljokSaWyMM4fdBCxHu7H0vrImAGL9-3Ss3v	复制	长期订阅	ha****tong *	洋情 删除
潮信支付							
物资服务							
硬件设备	摄像头监测异常报警	报警时间、所在位置、提示说明	ByASrNUB0bmEjlFZeGdrkgpTyZl8hygL4BK8	复制	长期订阅	ha****tong *	洋情 删除
30.02							
10 UA							

#### △ 注意:

• 摄像头品类的设备消息在订阅消息时,暂不支持勾选所在位置、剩余电量,否则会导致消息推送失败。

摄像头监测异常按照 报照时间 2021年10月21日 12:00:00 组示说明 摄像头监测频序常,请及 时处理       已选择 (2/5)     ● 投號时间	
相關時间 2021年10月21日 12:00:00 設示说明 補機失送測明序系、確及 的处理 已选择 (2/5) ご 报警时间	
已透择 (2/5) 📝 报警时间	
☞ 提示说明	
<b>场限说明</b> 说明订阅消息服务场限	
查看详情 >	



• 门锁品类的设备消息在订阅门锁电量不足提醒时,暂不支持勾选剩余电量,否则会导致消息推送失败。

你可以在同一个公共模板下,使用不同的关键词。	提交后,关键词的内容和顺	厚将无法修改。		
摄像头助手	关键词	☑ 提醌时间		
门锁电量不足提醒 提醒时间 2022年2月22日 12:00 提示说明 设备电量低,请及时更换		☑ #550℃		
电池或充电	已选择(2/5)	✓ 提醒时间	=	
		✓ 提示说明	=	
	场景说明	说明订阅消息服务场景	0/15	
查看详情				
		提夺		

# 4. 接入小程序插件

- 申请 X-P2P插件
- 申请 p2p-player插件
- 申请 腾讯连连小程序插件
- 申请 腾讯位置服务地图选点插件
- 创建地图选点应用
  - 微信登录 腾讯位置服务 后,在应用管理中创建一个地图选点的应用。

🗾 腾讯位置船	务	产品 - 解决方案 - 开发文档 -	春户案例 生态合作 服务升级 -	Q	11 控制台 🏾 🌀 -
Ξ 首页		我的应用			+ 创建应用
<b>11</b> 应用管理	*				
我的应用		家庭选点 1988年4			∠ 編辑 + 添加Key * 收起
签 配赖管理	v	KeyBla	Key	NUK	86
⊜ 数据管理	¥	家庭地址选点	G2	4	<b>编辑</b> 移动
二 个人中心	¥			_	
⊜ 订单	÷		创建应用		
45 IØ	*		<b>应用名称*</b> 名称可包含汉字、数字、字母, 7	继过16个字	
!猫 个性化地图	×		<b>应用类型*</b> 请选择应用类型	•	
前 地图商户中心	٣		10.0	612	
念 数据图展	٣				
-8. 轨迹云	*				
					្
		100 m			
		LINE CAS			12

○ 对创建的应用单击添加 key。





○ 将注册小程序的 APP ID 填入后,单击添加后,可在应用列表获得对应的 Key 字段。

💉 腾讯位置)	服务	产品 - 解决方案 - 开发文档 ·	客户案例 生态合作 <b>温务升级 -</b> Q	11 拉那台 🍥 -
EE 首页	*	我的应用		+ 台(建成两
我的应用		家庭选点 1988年4	添加key到「家庭选点」应用	∠ 编辑 + 添加Kay * 收起
二 配额管理	*	Kay名称	Kay名称" 示例家庭地址选点 雅道	制作
⊜ 叙想管理	٠	家庭地址透点	<b>描述</b> 例: 能结成划, 获取用户信重	1987) 
▲ 个人中心	×		常用作品 Jarvascript API、Jarvascript API GL、地图铝件、Android差位SDK、IOS定位 SDK在成功会IMMery历、即可使用	
	×		U WebServiceAPI	
11 个性化地图	¥		SDK	
言 地图商户中心	*		図 数位小程序 提択 APP ID* wo 4	
◆ 数据图度	Ť		前读并词意《病讯位重服务开放API服务协议》和《病讯位重服务输私协议》	
2 # <u>8</u> 2	Ť		80	
				9
				联系管理



# 授权部署发布小程序

最近更新时间: 2024-09-06 16:43:51

# 1. 注册登录

登录 腾讯云官网。如果您之前未注册腾讯云账号,请参见 账号注册教程 完成注册。

# 2. 创建产品和设备

登录 物联网智能视频服务(消费版 ) 控制台(以下简称 Video 控制台),创建产品及设备,产品为一类真实设备的集合,创建的设备则对应真实设备。

创建新产品		×
产品名称 *		
	支持中文、英文、数字、下划线的组合,最多不超过20个字符	
产品品类 *	儿童手表 摄像头 门锁	
配网方式 *	Wi-Fi 配网 4G 配网	
产品有效期 *	1年 5年	
操作系统*	通用	•
芯片厂商 *	通用	
芯片型号*	通用	•
描述	请输入描述信息	
	最多不超过128个字符	
	确定 取消	

# 3. 创建应用

1. 单击控制台左侧导航栏的 应用开发,进入应用列表页面。按需单击新建应用或选择已有应用。

应用开发				
	新建应用			应用名称C
	应用名称	创建时间	操作	
	loT	2022-11-02 11:42:19	编辑删除	
	Je	2022-10-19 15:10:07	编辑删除	
	π	2022-09-28 10:14:25	编辑 删除	

<sup>2.</sup> 进入应用详情页后,关联在 步骤2 中创建的产品。如不关联,则无法在小程序上成功手动添加设备。



🔗 腾讯云 🛛 🕸 🗌	云产品 ~			搜索产品、文档	Q. @ 小程序	── <sup>──</sup> 集团账号 ~	备案 工具 ~	支持 ~	费用 🗸 🛛 👻 🔻
物联网智能视频服务	← Tyrnayrill 188							物联网智能	视频服务帮助文档 IZ
○ 产品管理		应用信息							
<ul> <li>① 回仟升级</li> <li>⑦ 物模型</li> </ul>		App ID Tr To							
山 AI数据模型		应用名称 Tu							
♦ 应用开发		应用描述 -							
⊘ 数据统计		创建时间 2022-09-28 10:14:25							
☑ 资源包管理		Android应用 ①		iOS应用 ①					
		АРР Кеу ау		АРР Кеу ір 🌆					
		APP Secret Tti IRDW		APP Secret U VUHynl					
		小程序应用 ① 去授权 部署发布							
		APP Key mc							
		APP Secret YV 3/bhJ							
		关联产品							
		① 自主品牌应用只有与产品关联后,用户才可有权限对设备进行配网、绑定	2,控制等操作。						9
		产品名称	产品品类	关联	l l				
		xu	儿童手表						<b>1</b>
		xu	门锁		J				=

# 4. 授权小程序

单击小程序应用右侧的**去授权**,则可弹窗显示授权窗口,小程序管理员使用手机微信扫码后,选择授予权限给 loT Video。支持将注册小程序重复授权和授 权给其他应用,但仅对最新授权的应用生效。

於 購訊云 ∩ 总定	±#	- B. Y					搜索产品、文档	Q. @ 小程	¢ ⊠ <mark>'</mark>	集团账号 ~	备案 工具	、 支持  费用
物联网智能视频服务	÷	Turnkey1用_不	更动我									物联网智能视频剧
◎ 产品管理												
③ 固件升级			应用信息	L.								
☞ 物模型			App ID	TT TO								
山 AI数据模型			应用名称	Tu								
♦ 应用开发			应用描述									
⊘ 数据统计			创建时间	2022-09-28 10:14:25								
☑ 资源包管理			Android	应用③		iOS应用 (i	)					
			APP Key	a' 🗇		APP Key	ij G					
			APP Sech	et T 3RDW		APP Secret	L VUHyni 🗖					
			小程序应	用 ① 去授权 部署发布								
			APP Key	ma 🗖								
			APP Secn	et Yv SfbhJ 🗖								
			关联产品									
			0	自主品牌应用只有与产品关联后,用户才可有权限对设备进行配网	]、绑定,控制等操作。							
			产品名称	8	产品品类		关联					
			×	1. Th	儿童手表							

# 5. 部署发布小程序

待授权完毕后,可单击**部署发布**,完成部署发布小程序。

## 5.1 小程序配置



物联网智能视频图络 ← IoT_Video低代码示6。 都豐放布		
9 <i>F&amp;</i> ##	() (	4 1920
© 1097.948		
0 1041	<b>最新構成版本: 3.2.1</b>	
- A A 放送検知		
\$ 应用开发	基础配置	
© mmmit	家庭位置近现boy ·	
□ 资源包管理	保護維治息配置	
	792 <b>4</b> -	
	Difference id*	
	\$U.750.	
	138287.922898 •	A CONTRACTOR OF
	(1999-88 <b>8</b> ·	
	(1852a)#+	
	开关门跳器。	
	1184.87.8.9M*	
	()研研の問題。	
	售后与反馈配置	
	然后电话 (法则)	
	0(002239091) (2540)	
	APP下檔二線码(含複)	
		國家上做,peg婚式的二曲時, 國內策度讓沒不經过150px
	小程序使导说明配置	
	>-R#0008091 (30)	还来选择图时,请上约 prog 格式文件
		双种医疗
	产品使用设施配置 (改集)	建立上供_pag格式的透射非常整。把片发现建设不能过278px
	#0^&\$8	確你不ら診察性 操作
		通4世》77第30年品種種類的
		Internet of the Second Second
		下一步。至成体验分程序

- 显示支持的小程序最新版本。
- 基础配置

家庭位置应用 key: 填写准备工作第4步 创建地图选点应用 时,获取得到的 Key 值。

- 强提醒消息配置
  - **产品品类:**目前支持摄像头、门锁,共两种品类,支持多选。
  - 默认产品 ID: 填写在 第2步 中,指定品类下已创建的产品 ID。
  - model\_id: 填写在小程序管理后台添加的指定品类的 model\_id。
  - 消息模板:填写小程序管理后台订阅的消息模板 id,注意检查是否填写正确,若填写错误,则小程序接收不到微信推送的强提醒消息。
- 售后与反馈
  - 售后电话 (选填): 配置在小程序售后反馈的电话,不填写则不会在小程序上显示相关内容。
  - 问题反馈指引(选填):配置问题反馈的文案内容。
  - App下载二维码(选填):本地上传二维码图片,建议宽度不超过150px。
- 小程序使用说明配置

小程序使用说明书(选填):本地上传说明书的图片,建议宽度不超过370px。

● 产品使用说明配置(选填):

单击下方添加产品使用说明后,输入硬件产品的名称,以及上传对应的产品使用说明书,支持添加多个。

# 5.2 小程序 UI 配置

物联网智能视频服务	← 授权测试_摄像头	助: ####################################				物戰阿智能視頻則
♀ 产品管理 ④ 固件升级 ♀ 物模型		<ul> <li>小程序配置 &gt; 2 小程序UI配置</li> <li>最新概版版本: 3.3.0</li> </ul>	★ > ③ 生成体验小程序 > ④ 微信审核	> (5) 上线发布		
·II AI数据模型       ② 应用开发       ② 数据统计       ① 资源包管理		小程序主面配色选择。 首页度图图片。	10 · ( #FTEDAC ) III E21 · ( #FCDAC ) III E22 · ( AEFFEA ) III	<sup>1150</sup> 現点用户的家→ ・・・ ① Hi 微信用户! >	RAREA MITAL	
		门场交流被损失。	王政上校 単単 建议上校開み代寸かがたね。大小不相対1MB 建築上校町の大寸かが5440、大小不相対1MB 建築上校町の大寸かが5541、大小不相対1MB		□ 「」」「」」」」」」」 □ 「」」」」」 □ 「」」」」 □ 「」」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」」 □ 「」」 □ 「」」」 □ 「」 □ 「」 □	
8893					ECEN	



- 小程序主题配色选择:显示小程序默认的配色和预览效果,支持对主色和配色进行修改和重置。
- 首页底部图片: 支持替换首页底部的图片,可按照建议本地上传。
- 门锁顶部图片: 支持替换门锁设备页顶部的图片,可按照建议本地上传。

#### 5.3 生成体验小程序

1. 单击小程序配置页下方的**下一步: 生成体验小程序**可跳到对应页面,如下:

✓ 小程序配置	>	2 生成体验小程序	>	③ 微信审核	>	<ol> <li>上紙服布</li> </ol>
最新模版版本: 3.2.1						
准备构建小程序体验	板					
						取消胡春发布 立即构建

2. 单击**立即构建**后,3分钟内生成体验二维码,通过手机扫描二维码进入体验版小程序进行功能体验并与设备端开启功能调试。在单击**取消部署发布**后,则可回到小程序配置步骤,进行重新的配置步骤。

→ 小程序配置 > 2 生成体验小程序	> ③ 微信审核 > ④ 上线发布
最新模版版本: 3.2.1	
构建完成,请扫二维码体验,模板版本号:3.2.1	1011/042/465/24101
	取消部署发布 下一步:微信审核

## 5.4 微信审核

单击**下一步:微信审核**,则可进入微信审核页面,在单击**立即审核**后,则可等待微信审核,时间为1天 – 7天。期间可关闭弹窗,并可再次单击**部署发布**按钮 进入弹窗查看审核结果,同时审核结果也会在审核完毕后第一时间以微信服务通知发送给小程序管理员。

•	小程序	配置 >	< 生成体验小	N程序 〉	3 微信审核	> (4	<ol> <li>上越发布</li> </ol>
	()	准备提交微信 点击下方按钮	<b>言审核</b> 田即可将小程序内容提交	至微信审核。			
							取消器 <b>素</b> 发布 开始银线

## 5.5 上线发布

待微信审核通过后,可单击**下一步:上线发布**进入上线发布页面。单击**发布**后,即可正式发布小程序,并支持微信中搜索发布的正式版小程序并开始使用。

# 设备端接入(摄像机)

最近更新时间: 2024-09-06 15:42:31

#### 摄像头扫码配网

设备侧需自行完成门锁摄像头扫码解析小程序二维码的功能,具体如下:

- 1. 启动小程序,选择**手动添加设备**,确认设备重置后,选择家庭 Wi−Fi 并输入密码,则可使用摄像头扫描小程序生成的配网二维码,其中二维码解析后的 格式如下:
  - 内容格式: len str len str len str。其中 len: 2个字符,代表紧随内容 utf-8 解码的长度, str 顺序: ssid、password、token。

○ 示例:

以 Tencent-wifi, 密码为 I, token 为 v3\_00d······为例, 设备侧需要解析的字符串为:

12Tencent-WiFi01132v3\_00d02b0248546d5a5fbf95a5812e4

其对应的3个字段如下:

- { "SSID": "Tencent-wifi",
   "password": "1",
   "token": "v3\_00d02b0248546d5a5fbf95a5812e4" }
- 2. 设备侧解析完 wifi ssid 和密码后,自行完成联网操作。
- 3. 连接完成后调用 系统模块 iv\_sys\_init() 和 物模型模块 iv\_dm\_init() 初始化必要的功能模块。
- 4. 在确认初始化接口返回成功且确保设备上线后,调用 iv\_ad\_send\_bind\_info() 配网接口,上报上述配网 token。
- 5. 上报成功后,刷新小程序主页面即可看到刚添加的设备。

# 云台控制

设备侧需完成小程序云台信令的响应,参考 交<mark>互信</mark>令 。

# 语音对讲

- 设备对接协议:设备到小程序上行仅支持 AAC,小程序到设备下行支持 AAC 16K 采样率或者设置任意采样率的 PCM,但下行 PCM 不支持小程序回 音消除,推荐使用 AAC 16K。
- 采样率:为了有更好的延时效果,设备到小程序的 AAC 格式请设置为 8K 采样率以上。
- 语音对讲设备侧 响应信令。

#### 本地录像回放

- 设备侧需响应 本地录像回放指令。
- 获取某月有本地录像文件的日期:设备侧按需返回日期。
- 获取某一天所有本地录像文件的列表:设备侧需返回 json 列表。
- 点播某个录像文件:设备侧需 推送音视频流。
- 暂停播放某个录像文件:设备侧需停止推送音视频流。
- 恢复播放某个录像文件:设备侧需继续推送音视频流。
- 实时显示某个录像文件的播放进度:设备侧需实时响应当前播放进度请求,并返回相对时间戳。

#### 云存回放

- 设备端对接设备端 SDK,某一个事件发生时,传入事件 ID、图片,并根据回调进行音视频流推送。
- 云存储模块说明
- 事件类型:



- 有人活动: event id = 1。
- 画面发生变化: event id = 2。

## 相关物模型

所需的物模型需要在控制台的物模型页面进行操作,主要包括需要导入的相关物模型和添加的必选标准功能的物模型,同时也支持添加自定义功能的物模 型。



#### 导入相关物模型 json 内容:

```
{
    "version": "1.0",
    "profile": {
        "ProductId": "FT92000JFG",
        "categoryId": "113"
    },
    "properties": [
        {
            "id": "_sys_xp2p_info",
            "name": "xp2p信息",
            "desc": "",
            "mode": "rw",
            "define": {
               "type": "string",
               "min": "0",
               "max": "64"
            },
            "required": false
        },
        {
            "id": "_sys_cs_days",
            "name": "云存全时天数",
        "define": {
               "id": "rw",
              "define": {
               "id": "rw",
               "define": {
               "type": false
               },
               "required": false
               },
               "desc": "",
               "name": "云存全时天数",
              "define": {
                "udefine": {
                "typ",
              "define": {
               "id": "rw",
              "define": {
               "id": "rw",
              "define": {
               "id": ".sys_cs_days",
               "name": "云fet时天数",
               "define": {
               "id": "rw",
              "define": {
               "id": "rw",
              "define": {
               "define": {
               "udefine": {
               "type": "rw",
              "define": {
               "define": {
               "define": {
               "type": "rw",
              "define": {
               "type": "rw",
              "define": {
               "type": "rw",
              "define": {
               "type": "rw",
               "define": {
                "type": "rw",
               "define": {
               "type": "rw",
              "define": {
               "define": {
               "type": "rw",
              "define": {
                "define": {
               "type": "rw",
              "def
```



```
"name": "云存开关",
"name": "云存类型",
  "1":"全时",
  "2": "事件"
"name": "本地录像使能",
```





"mapping":	{		
"0": "天			
ι · · π			
},			
"required":			
}			
],			
"events": [],			
"actions": []			
}			

# 设备端接入(图片流门锁)

最近更新时间: 2024-09-06 15:42:31

## 门锁摄像头扫码配网

腾讯云

设备侧需自行完成门锁摄像头扫码解析小程序二维码的功能,具体如下:

- 1. 启动小程序,选择**手动添加设备**,确认设备重置后,选择家庭 Wi−Fi 并输入密码,则可使用摄像头扫描小程序生成的配网二维码,其中二维码解析后的 格式如下:
  - 内容格式: len str len str len str。其中 len: 2个字符,代表紧随内容 utf-8 解码的长度, str 顺序: ssid、password、token。

○ 示例:

以 Tencent-wifi, 密码为 I, token 为 v3\_00d ······为例, 设备侧需要解析的字符串为:

12Tencent-WiFi01132v3\_00d02b0248546d5a5fbf95a5812e4

其对应的3个字段如下:

{ "SSID": "Tencent-wifi",
 "password": "1",
 "token": "v3\_00d02b0248546d5a5fbf95a5812e4" }

- 2. 设备侧解析完 wifi ssid 和密码后,自行完成联网操作。
- 3. 连接完成后调用 系统模块 iv\_sys\_init() 和 物模型模块 iv\_dm\_init() 初始化必要的功能模块。
- 4. 在确认初始化接口返回成功且确保设备上线后,调用 iv\_ad\_send\_bind\_info() 配网接口,上报上述配网 token。
- 5. 上报成功后,刷新小程序主页面即可看到刚添加的设备。

#### 语音对讲

- 设备对接协议:设备到小程序上行仅支持 AAC,小程序到设备下行支持 AAC 16K 采样率或者设置任意采样率的 PCM,但下行 PCM 不支持小程序回 音消除,推荐使用 AAC 16K。
- 采样率:为了有更好的延时效果,设备到小程序的 AAC 格式请设置为 8K 采样率以上。
- 相关信令

#### 云存回放

- 购买云存储,需要通过 云 API 后台操作,为设备配置云存储套餐。
- 设备端对接设备端 SDK,某一个事件发生时,通过事件上报接口 iv\_cs\_event\_start()或者 iv\_cs\_event\_directly\_report()向控制台后台 传入事件 ID、图片,并根据回调进行音视频流推送。
- 云存储模块说明
- 事件类型:目前支持事件类型见下表,注意 ID 和事件名称的对应关系,后续将持续拓展。

事件类型	事件 ID
有人按门铃	event ID = 3
撬锁告警	event ID = 4
密码/指纹错误冻结告警	event ID = 5
胁迫告警	event ID = 6
徘徊告警	event ID = 7
开门通知	event ID = 8

关门通知	event ID = 9
指纹禁试告警	event ID = 10
密码禁试告警	event ID = 11
卡片禁试告警	event ID = 12
人脸禁试告警	event ID = 13
逗留侦测	event ID = 14
系统禁试告警	event ID = 15
假锁告警	event ID = 16
防遮挡告警	event ID = 17

## 消息上报

设备端对接设备端 SDK: 当事件发生时,设备端通过 iv\_msg\_send\_notice()传入通知 ID 到控制台后台,进行消息模板匹配,将通知参数传入小程序 {{time1.DATA}}

{{enum\_string2.DATA}} 后,即可进行消息通知推送。

• 通知类型:目前支持通知类型见下表,注意 ID 和通知的对应关系,后续将持续拓展。通知将显示在小程序设备详情页中,保存有效期30天。

通知类型	通知 ID
开门通知	notice ID = 1
关门通知	notice ID = 2
忘拔钥匙	notice ID = 3
反锁	notice ID = 4
低电量告警	notice ID = 5
有人按门铃	notice ID = 6
钥匙开门	notice ID = 7
临时密码开门	notice ID = 8
门未关提醒	notice ID = 9
未上锁提醒	notice ID = 10
门虚掩提醒	notice ID = 11
未知情况	notice ID = 12

#### 🕛 说明:

- 设备端通过 iv\_msg\_send\_notice (notice\_id: 1, ext: {unlock\_type: xx})上报开门通知。不传入 unlock\_type,消息显示
   为 "开门通知"; 传入unlock\_type,消息显示对应的 unlock\_type。
- 设备端通过 iv\_msg\_send\_notice (notice\_id: 12, ext: {unknown\_info\_id: xx}) 上报消息。不传入 unknown\_info\_id, 消息显示为 "未知情况"; 传入unknown\_info\_id, 消息显示 id 对应的消息通知。

unlock\_type 有下表几种:

unlock_type	通知
1	密码开门

2	人脸开门
3	指纹开门
4	卡片开门
5	门内开锁
6	App 蓝牙开门
7	指纹+密码开锁
8	人脸+指纹开锁
9	人脸+密码开锁

#### unknown\_info\_id 有下表几种:

unknown_info_id	通知
1	锁舌无法伸出告警
2	锁舌无法收回告警
3	防遮挡告警
4	布防告警
5	假期模式关闭告警
6	假锁告警

# 落锁状态

- 设备端对接设备端 SDK,通过 mqtt 消息上报"落锁状态"属性物模型,同步门锁开关信息。
- 物模型模块介绍

功肯	能类型	功能名称	标识符	数据类型	读写类型	数据定义
属性	性	电池电量	battery_percentage	整数型	读写	数值范围: 0-100 初始值: 0 步长: 1 单位: 百分比
属性	性	落锁状态	lock_motor_state	布尔型	只读	0 - 未落锁 1 - 已落锁

# 远程解锁

设备端对接设备端 SDK,通过 SDK 接收到服务器下发的远程解锁行为物模型,接收解锁行为通知后,开锁指令下发成功,待小程序获取到落锁状态物模 型由"已落锁"转为"未落锁",则说明锁已开,提示用户"开锁成功"。



*	行为	远程解锁		unlock_remote -	-	
	调用参数					
	参数名称		参数标识符		数据类型	数据定义
					暫无调用参数	
	返回参数					
	参数名称		参数标识符		数据类型	数据定义
	开锁成功状态		result		布尔型	0 - 开锁失败 1 - 开锁成功

# 电池电量信息上报

设备端对接设备端 SDK,通过上报"电池电量"属性物模型,同步门锁电量信息。

功能类型	功能名称	标识符	数据类型	读写类型	数据定义
属性	电池电量	battery_percentage	整数型	读写	数值范围: -1-100 初始值: 0 步长: 1 单位: 百分比

# 临时密码解锁

- 原理: RFC4226
  - perio:密码有效期间隔,30秒一个。
  - 将设备 PSK 密钥按照 base64 解码后得到原始二进制,得到 secret。
  - 将时间戳除以 period,向下取整,得到 counter。
  - 对 counter 计算 sha1。
  - 将 sha1 使用 dynamic truncation 计算为6位数字。

#### • 代码参考

设备端需自行实现 RFC4226 算法,设备端在用户输入临时密码时,获取输入密码对应的时间戳后,通过 RFC4226 算法生成密码,并与输入密码进行逐一比对,至对比成功则可正常开锁。

⚠ 注意: 当门锁状态显示已开锁时,临时密码不可使用。

#### 相关物模型

所需的物模型需要在控制台的物模型页面进行操作,主要包括需要导入的相关物模型和添加的必选标准功能的物模型,同时也支持添加自定义功能的物模 型。

# 🔗 腾讯云

物联网智能视频服务	÷	frostLock1027									物联网
◎ 产品管理		1		导入JSON 查看JSON	N 重置物模型 产品品类:门锁						
⑦ 固件升级			-								
			Ċ	标准功能							
III AI数据模型			2	添加标准功能标准功能	能为系统推荐,您可按需选择						
◊ 应用开发				功能类型	功能名称	标识符	数据类型	读写类型	数据定义	操作	
❷ 数据统计				属性	xp2p信息 必选	_sys_xp2p_info	字符串	读写	字符串长度: 0-64个字符	编辑 删除	
□ 资源包管理				属性	云存全时天数 必选	_sys_cs_days	整数型	读写	数值范围:0-100 初始值:0 步长:1 单位:	编辑 删除	
				属性	云存开关 必选	_sys_cs_status	布尔型	读写	0-关 1-开	编辑 删除	
				属性	云存类型 必选	_sys_cs_type	枚举型	读写	1 - 全时 2 - 事件	编辑 删除	
				属性	落锁状态 可选	lock_motor_state	布尔型	只读	0 - 未落锁 1 - 已落锁	编辑 删除	
				属性	电池电量 可选	battery_percentage	整数型	读写	数值范围: -1-100 初始值: 0 步长: 1 单位: 百分比	编辑删除	
		3	•	自定义功能 添加自定义功能	以通过自定义功能按需定义功能						1
				功能类型	功能名称	标识符	數据类型	读写类型	数据定义	操作	

导入相关物模型:

```
"name": "电池电量",
"desc": "-1表示未获取到电量",
 "unit": "百分比"
"name": "门锁状态",
 "mapping": {
"0": "未落锁",
   "1": "已落锁",
   "3": "门锁异常"
```



```
"name": "xp2p<mark>信息",</mark>
"name": "云存全时天数",
"name": "云存开关",
   "0":"关",
"name": "云存类型",
   "1": "全时",
   "2": "事件"
```



```
"mapping": {
"0": "关",
"1": "开"
"name": "图片流标识",
"name": "远程解锁",
    "name": "开锁成功状态",
       "0": "开锁失败",
        "1": "开锁成功"
```

# 设备端接入(视频流门锁)

最近更新时间:2024-09-06 15:42:31

# 1. 门锁摄像头扫码配网

设备侧需自行完成门锁摄像头扫码解析小程序二维码的功能,具体如下:

- 1.1 启动小程序,选择**手动添加设备**,确认设备重置后,选择家庭 Wi-Fi 并输入密码,则可使用摄像头扫描小程序生成的配网二维码,其中二维码解析 后的格式如下:
  - 内容格式: len str len str len str。其中 len: 2个字符,代表紧随内容 utf-8 解码的长度, str 顺序: ssid、password、token。

○ 示例:

以 Tencent-wifi, 密码为I, token 为 v3\_00d ······为例, 设备侧需要解析的字符串为:

12Tencent-WiFi01132v3\_00d02b0248546d5a5fbf95a5812e4

#### 其对应的3个字段如下:

{ "SSID": "Tencent-wifi",
 "password": "1",
 "token": "v3\_00d02b0248546d5a5fbf95a5812e4" }

1.2 设备侧解析完 wifi ssid 和密码后,自行完成联网操作。

- 1.3 连接完成后调用 系统模块 iv\_sys\_init() 和 物模型模块 iv\_dm\_init() 初始化必要的功能模块。
- 1.4 在确认初始化接口返回成功且确保设备上线后,调用 iv\_ad\_send\_bind\_info() 配网接口,上报上述配网 token。
- 1.5 上报成功后,刷新小程序主页面即可看到刚添加的设备。

# 2. 语音对讲

- 设备对接协议:设备到小程序上行仅支持 AAC,小程序到设备下行支持 AAC 16K 采样率或者设置任意采样率的 PCM,但下行 PCM 不支持小程序回 音消除,推荐使用 AAC 16K。
- 采样率:为了有更好的延时效果,设备到小程序的 AAC 格式请设置为 8K 采样率以上。
- 相关信令

#### 3. 云存回放

- 购买云存储,需要通过 云 API 后台操作,为设备配置云存储套餐。
- 设备端对接设备端 SDK,某一个事件发生时,通过事件上报接口 iv\_cs\_event\_start()或者 iv\_cs\_event\_directly\_report()向控制台后台 传入事件 ID、图片,并根据回调进行音视频流推送。
- 云存储模块说明
- 事件类型:目前支持事件类型如下表,注意 ID 和事件名称的对应关系,后续将持续拓展。

事件类型	事件 ID
有人按门铃	event ID = 3
撬锁告警	event ID = 4
密码/指纹错误冻结告警	event ID = 5
胁迫告警	event ID = 6
徘徊告警	event ID = 7
开门通知	event ID = 8



关门通知	event ID = 9
指纹禁试告警	event ID = 10
密码禁试告警	event ID = 11
卡片禁试告警	event ID = 12
人脸禁试告警	event ID = 13
逗留侦测	event ID = 14
系统禁试告警	event ID = 15
假锁告警	event ID = 16
防遮挡告警	event ID = 17

# 4. 消息上报

• 设备端对接设备端 SDK: 当事件发生时,设备端通过 iv\_msg\_send\_notice() 传入通知 ID 到控制台后台,进行消息模板匹配,将通知参数传入小 程序 {{time1.DATA}}

{{enum\_string2.DATA}} 后,即可进行消息通知推送。

• 通知类型:目前支持通知类型如下表,注意 ID 和通知的对应关系,后续将持续拓展。通知将显示在小程序设备详情页中,保存有效期30天。

通知类型	通知 ID
开门通知	notice ID = 1
关门通知	notice ID = 2
忘拔钥匙	notice ID = 3
反锁	notice ID = 4
低电量告警	notice ID = 5
有人按门铃	notice ID = 6
钥匙开门	notice ID = 7
临时密码开门	notice ID = 8
门未关提醒	notice ID = 9
未上锁提醒	notice ID = 10
门虛掩提醒	notice ID = 11
未知情况	notice ID = 12

#### () 说明:

- 设备端通过 iv\_msg\_send\_notice (notice\_id: 1, ext: {unlock\_type: xx})上报开门通知。不传入 unlock\_type,消息显示
   为 "开门通知"; 传入unlock\_type,消息显示对应的 unlock\_type。
- 设备端通过 iv\_msg\_send\_notice (notice\_id: 12, ext: {unknown\_info\_id: xx}) 上报消息。不传入 unknown\_info\_id, 消息显示为 "未知情况"; 传入 unknown\_info\_id, 消息显示 id 对应的消息通知。

unlock\_type 类型如下:

unlock_type	通知
1	密码开门

2	人脸开门
3	指纹开门
4	卡片开门
5	门内开锁
6	Арр 蓝牙开门
7	指纹+密码开锁
8	人脸+指纹开锁
9	人脸+密码开锁

#### unknown\_info\_id 类型如下:

unknown_info_id	通知
1	锁舌无法伸出告警
2	锁舌无法收回告警
3	防遮挡告警
4	布防告警
5	假期模式关闭告警
6	假锁告警

# 5. 落锁状态

- 设备端对接设备端 SDK,通过 mqtt 消息上报"落锁状态"属性物模型,同步门锁开关信息。
- 物模型模块介绍

功能类型	功能名称	标识符	数据类型	读写类型	数据定义
属性	电池电量	battery_percentage	整数型	读写	数值范围: 0-100 初始值: 0 步长: 1 单位: 百分比
属性	落锁状态	lock_motor_state	布尔型	只读	0 - 未落锁 1 - 已落锁

# 6. 远程解锁

设备端对接设备端 SDK,通过 SDK 接收到服务器下发的远程解锁行为物模型,接收解锁行为通知后,开锁指令下发成功,待小程序获取到落锁状态物模 型由"已落锁"转为"未落锁",则说明锁已开,提示用户"开锁成功"。



*	行为	远程解锁		unlock_remote -	-	
	调用参数					
	参数名称		参数标识符		数据类型	数据定义
					暫无调用参数	
	返回参数					
	参数名称		参数标识符		数据类型	数据定义
	开锁成功状态		result		布尔型	0 - 开锁失败 1 - 开锁成功

# 7. 电池电量信息上报

设备端对接设备端 SDK,通过上报"电池电量"属性物模型,同步门锁电量信息。

功能类型	功能名称	标识符	数据类型	读写类型	数据定义
属性	电池电量	battery_percentage	整数型	读写	数值范围: -1-100 初始值: 0 步长: 1 单位: 百分比

# 8. 临时密码解锁

- 原理: RFC4226
  - perio: 密码有效期间隔,30 秒一个。
  - 将设备 PSK 密钥按照 base64 解码后得到原始二进制,得到 secret。
  - 将时间戳除以 period,向下取整,得到 counter。
  - 对 counter 计算 sha1。
  - 将 sha1 使用 dynamic truncation 计算为6位数字。

#### • 代码参考

 设备端需自行实现 RFC4226 算法,设备端在用户输入临时密码时,获取输入密码对应的时间戳后,通过 RFC4226 算法生成密码,并与输入密码进行 逐一比对,至对比成功则可正常开锁。

# ▲ 注意:

当门锁状态显示已开锁时,临时密码不可使用。

# 9. 主动唤醒

#### 实现流程如下:

1.1 小程序获取唤醒状态物模型的值为0,即休眠状态,则会下发唤醒门锁行为。

属性	唤醒状态 可选	wakeup_state	布尔型	只读	0 - 休眠状态 1 - 唤醒状态



•	行为	唤醒门锁可选	wake_up		-
	调用参数				
	参数名称		参数标识符	数据类型	数据定义
				暂无调用参数	
	返回参数				
	参数名称		参数标识符	数据类型	数据定义
	唤醒状态		wakeup_state	布尔型	0 - 未唤醒 1 - 已成功唤醒

1.2 设备 Wi−Fi 收到唤醒命令,唤醒主控。待主控启动,调用 loT Video SDK,待初始化成功后,设备 Wi−Fi 上报**唤醒状态**为"唤醒状态"。

1.3 小程序获取唤醒状态为"唤醒状态"时,出图显示门锁实况。

# 10. 相关物模型

所需的物模型需要在**控制台 > 物模型** 页面进行操作,主要包括需要导入的相关物模型和添加的必选标准功能的物模型,同时也支持添加自定义功能的物模 型。

物联网智能视频服务	÷	frostLock1027							物联网
◎ 产品管理		1 [	与入JSON 查看JSOI	N 重重物模型 产品品类:门锁					
⑦ 固件升级									
分 物模型			标准功能						
∎ AI数据模型		2	添加标准功能标准功	能为系统推荐,您可按需选择					
☆ 应用开发			功能类型	功能名称	标识符	数据类型	读写类型	数据定义	操作
❷ 数据统计			属性	xp2p信息 必选	_sys_xp2p_info	字符串	读写	字符串长度: 0-64个字符	编辑 删除
□ 资源包管理			属性	云存全时天数 参选	_sys_cs_days	整数型	读写	数值范围: 0-100 初始值: 0 步长: 1 単位:	编辑 把印度
			属性	云存开关 必送	_sys_cs_status	布尔型	读写	0 - 关 1 - 开	编辑 删除
			属性	云存类型 必送	_sys_cs_type	枚举型	读写	1 - 全时 2 - 事件	编辑 删除
			属性	落锁状态 可选	lock_motor_state	布尔型	只读	0 - 未落锁 1 - 已落锁	编辑 删除
			愿性	电池电量 可选	battery_percentage	整数型	读写	数值范围: -1-100 初始值: 0 步长: 1 单位: 百分比	编辑 删除
		3	自定义功能	以通过自定义功能按需定义功能					
			功能类型	功能名称	标识符	數据类型	读写类型	数据定义	操作

#### 导入相关物模型:





```
"name": "电池电量",
"desc": "-1表示未获取到电量",
 "unit": "百分比"
"name": "音量",
  "0": "静音",
  "1": "低",
   "3": "高"
"name": "唤醒状态",
"desc": "设备主动上报是否处于唤醒状态",
  "0": "休眠状态",
  "1": "唤醒状态"
"name": "面容",
```



"id": "id",	
In . Ince_name ,	
"type": "string",	
"ld": "caras",	
"name": "卞万",	
"desc": "",	
"mode": "rw",	
"mode": "rw", "define": {	
"mode": "rw", "define": { "arrayInfo": {	
"mode": "rw", "define": { "arrayInfo": { "type": "struct",	
"mode": "rw", "define": { "arrayInfo": { "type": "struct", "specs": [	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         {         }         {</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         "id": "id",         "id"; "id",         "id"; "id",         "id"; "id",         "id"; "id",         "id"; "id",         "id"; "id",         "</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [             {                  "id": "id",                     "name": "id",                     "dataType": {</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                 "type: "string",                "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                      "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",                 "type: "string",</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                 "min": "0" </pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                   "min": "0",                  "name": "1400"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         "id": "id",         "name": "id",         "dataType": {             "type": "string",             "min": "0",             "max": "100"         ``</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                "type": "string",                  "min": "0",                 "max": "100"         } </pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [             {                 "id": "id",                 "name": "id",                 "dataType": {                     "type": "string",                     "min": "0",                     "max": "100"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [             {                 "id": "id",                 "name": "id",                 "dataType": {                     "type": "string",                     "min": "0",                     "max": "100"                 }</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [             {                 "id": "id",                 "name": "id",                 "dataType": {                     "type": "string",                     "min": "0",                     "max": "100"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                      "min": "0",                     "max": "100"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                      "min": "0",                     "max": "100"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                      "min": "0",                     "max": "100"         }         },         {</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {             "id": "id",             "name": "id",             "dataType": {                 "type": "string",                      "min": "0",                     "max": "100"         }         },         {</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [             {                 "id": "id",                 "name": "id",                 "dataType": {                     "type": "string",                     "min": "0",                     "max": "100"</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         "id": "id",         "name": "id",         "dataType": {             "type": "string",             "min": "0",             "max": "100"         }       },       {         "id": "card_name",         "name": "card_name",         "dataType": {             "type": "string",             "dataType": {                 "type": "string",                 "max": "20"         }        </pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         "id": "id",         "name": "id",         "dataType": {             "type": "string",             "min": "0",             "max": "100"         }       },       {         "id": "card_name",         "name": "card_name",         "dataType": {             "type": "string",             "type": "string",             "min": "0",             "max": "20"       }     } }</pre>	
<pre>"mode": "rw", "define": {     "arrayInfo": {         "type": "struct",         "specs": [         {         "id": "id",         "name": "id",         "dataType": {             "type": "string",             "min": "0",             "max": "100"         }       },       {         "id": "card_name",         "name": "card_name",         "dataType": {             "type": "string",             "type": "string",             "min": "0",             "max": "20"       }       }       } }</pre>	



"name": "指纹",



```
"name": "落锁状态",
   "0": "未落锁",
   "1": "已落锁"
"name": "童锁状态",
 "mapping": {
"0": "关闭",
"1": "开启"
"name": "布防模式",
   "1": "开启"
"name": "反锁状态",
```



"mode": "rw",

"0": "关闭", "name": "绑定终端用户列表", "name": "主人", "name": "被分享者", "name": "门锁状态", "0": "未落锁", "1": "已<mark>落锁</mark>", "3": "门锁异常"



```
"name": "xp2p信息",
"name": "云存全时天数",
"name": "云存开关",
"name": "云存类型",
  "1": "全时",
  "2": "事件"
```



"name": "唤醒门锁", "name": "唤醒状态", "0**": "未唤**醒", "1": "已成功唤醒" "name": "远程解锁", "name": "开锁成功状态", "0": "开锁失败", "1": "开锁成功"

# 低代码小程序蓝牙配网指南



最近更新时间: 2023-03-31 15:15:52

# 一、低代码小程序在控制台完成蓝牙配网配置

#### 1. 先在 控制台创建新产品,选择蓝牙辅助配网。

创建新产品		
产品名称*		
产品品类*	支持中文、英文、数字、下划线的组合,最多不超过20个字符	
配网方式*	Wi-Fi(二维码)配网 4G配网 蓝牙辅助配网	
产品有效期*	1年 5年	
操作系统*	通用	V
芯片厂商 *	通用	
芯片型号 *	通用	~
描述	请输入描述信息	
	最多不超过128个字符	
	确定 取消	

2. 进入对应产品页面进行配网设置,可编辑设备配网的文案和展示图片。



蓝牙配网		
配置硬件引导页	τ.	
引导文案标题	配置教程	预览 TChyst 22.4E 10265
	请输入配置硬件引导标题,10字符内	蓝牙辅助配网 ••• ④
配置教程	请根据设备使用说明书上的指引进行设备重置	
	请输入引导文案,65字符内	項稅總以前10月以由15月以前11以11以會重直 重置设备款程 >
特殊处理标题	重置设备教程	
	请输入特殊处理方法标题,10字符内	
重置设备教程	<ol> <li>接通设备电源。</li> <li>长按复位键(开关),指示灯慢闪。</li> <li>点击"下一步"开始篮牙辅助配网。</li> </ol>	0
	请输入特殊处理教程, 130字符内	
配网引导图		
	重新选择	○ 我已确认上述操作
	• 取消	
	建议尺寸: 670px * 670px, 支持png、jpg、gif格式, 大小不超过500KB	
按钮文案	下一步	
	按钮文案,如:"下一步"、"设备已开启配网模式",长度10个字符内	

- 3. 进入 应用开发,创建应用。搭建低代码的自主品牌小程序可参考 文档。
- 4. 若创建的应用需要使能蓝牙配网功能,在步骤3的基础上,进行如下操作:
  - 4.1 在控制台 应用开发 页面关联刚刚创建的蓝牙辅助配网产品。

关联产品			
<ul> <li>自主品牌应用只有与产品关联后,用户才可有权限对设备进行</li> </ul>	配网、绑定,控制等操作。		
产品名称	产品品类	关联	
蓝牙配网	摄像头		

4.2 进入部署开发,对低代码小程序的功能项和 UI 进行详细配置。



应用信息	
App ID	
应用名称	
应用描述 -	
创建时间	
Android应用 (j)	iOS应用 ①
АРР Кеу	APP Key
APP Secret	APP Secret
小程序应用 ① 再次授权 部署发布	
APP Key	

4.3 在小程序 UI 配置 > 默认产品配置中,选择创建的蓝牙配网产品;若修改该产品对应的配网文案和推案,该产品下的配网配置信息将同步更新。

	₩Ŋ广 茄 蚁: 1	
产品ID *	W34T5KJKC6 👻	
产品展示名称★	蓝牙测试	
产品展示图标	还未选择图片, 请上传 png 格式文件 选择图片	
	建议上传.png格式的透明底背景图,图片尺寸建议为64x64,最大50	10kb
产品品类	门锁	
蓝牙配网		
配置硬件引导页		
<b>配置硬件引导页</b> 引导文案标题	配置教程	预览
<b>配置硬件引导页</b> 引导文案标题	<b>配置教程</b> 请输入配置硬件引导标题,10字符内	预览 ••••• TCloud 22:05 100% 蓝牙辅助配网 ••• ④
<b>配置硬件引导页</b> 引导文案标题 配置教程	<b>配置教程</b> 请输入配置硬件引导标题,10字符内 请根据设备使用说明书上的指引进行设备重置	预览
配置硬件引导页 引导文案标题 配置教程	<b>配置教程</b> 请输入配置硬件引导标题,10字符内 请根据设备使用说明书上的指引进行设备重置	
配置硬件引导页 引导文案标题 配置教程	<b>配置教程</b> 请输入配置硬件引导标题,10字符内 <b>请根据设备使用说明书上的指引进行设备重置</b> 请输入引导文案,65字符内	研究
配置硬件引导页 引导文案标题 配置教程 特殊处理标题	<b>配置教程</b> 请输入配置硬件引导标题,10字符内 <b>请根据设备使用说明书上的指引进行设备重置</b> 请输入引导文案,65字符内 <b>哈哈</b>	预览 蓝牙辅助配网 ( 配置硬件 设置家庭WEP 连接设备查牙 开始配 配置数程 请根据设备使用说明书上的指引进行设备重置 哈哈 >
配置硬件引导页 引导文案标题 配置教程 特殊处理标题	配置教程 请输入配置硬件引导标题,10字符内 请根据设备使用说明书上的指引进行设备重置 请输入引导文案,65字符内 G哈 请输入特殊处理方法标题,10字符内	研究
配置硬件引导页 引导文案标题 配置教程 特殊处理标题 哈哈	配置教程 请输入配置硬件引导标题,10字符内 请根据设备使用说明书上的指引进行设备重置 请输入引导文案,65字符内 哈哈 请输入持殊处理方法标题,10字符内 哈哈	一 一 一 一 一 一 一 一 一 一 一 一 一

# 二、设备端完成蓝牙配网模块开发

• 小程序整体蓝牙配网流程可参考 blufi 蓝牙辅助配网开发。





● 设备端配合小程序配网,除了需要集成我们邮件发送的 loT Video 设备端 SDK,还需要集成蓝牙 SDK,可参考说明。

• 设备端可参考 LLSync SDK 辅助配网功能接入指引。

完成蓝牙配网功能适配,在设备连接网络完毕后,再调用 IoT Video 设备端 SDK中的上报配网 Token 接口,完成蓝牙配网流程。

# 音视频模块 APP SDK概述

最近更新时间: 2024-04-24 10:32:01

# SDK介绍

IoT Video X-P2P SDK集成了音视频通讯模块,具体包含 iOS 版本 或 Android 版本 ,用户可参考 SDK Demo 快速接入 App SDK。

# App SDK版本更新

# V2.4

- 发布日期: 2022-08-30
- 系统平台: iOS/Android
- 开发语言: OC 语言/Java
- •开发环境: Mac、Xcode/Android studio
- 更新内容:
  - 支持MJPEG类型设备出图。
  - 底层XP2P通讯协议升级,支持中转和打洞并行,快速出图。

## V2.3

- 发布日期: 2021-09-30
- 系统平台: iOS/Android
- 开发语言: OC 语言/Java
- •开发环境: Mac、Xcode/Android studio
- 更新内容:更新底层XP2P通讯协议,从而支持小程序插件

## V2.2

- 发布日期: 2021-07-30
- 系统平台: iOS/Android
- 开发语言: OC 语言/Java
- 开发环境: Mac、Xcode/Android studio
- 更新内容:
  - 增加支持多通道设备预览。
  - Demo App 支持视频清晰度切换。
  - Demo App 支持边录边播、截图、云台控制。
  - Demo App 支持多路同屏。

服务提供方:腾讯云计算(北京)有限责任公司

《SDK接入指引》

《开发者合规指南》

《IoT Video X-P2P SDK个人信息保护规则》



# Android 应用端接入手册 快速开始

最近更新时间: 2023-04-25 10:02:03

本文为您介绍 Android 应用端通过 so 库和 aar 库方法集成 SDK。

## 使用动态库 so

#### 下载路径

#### 稳定版

进入 so 下载地址,选择对应 Latest Version (例如2.0.3),单击右侧 Downloads > aar。

```
• SNAPSHOT 版
```

```
进入 so下载地址,单击 Repositories > Snapshots > Browse Storage 的 Path Lookup 输入框中输入
```

```
com/tencent/iot/thirdparty/android > xp2p-sdk >选择对应版本(例如1.0.2-SNAPSHOT),选择最新的 .aar 单击右键
Downloads。
```

() 说明:

建议使用稳定版本,SNAPSHOT 版仅供开发自测使用。

#### 引用工程

1. 解压上一步骤下载下来的 aar,目录结构如下:



2. 将头文件和 so 动态库放在您工程目录下,确保 CMakeList.txt 文件可以找到对应的路径即可。使用示例如下:



#### 在 CMakeLists.txt 中加上以下代码即可。

```
add_library(test-lib SHARED IMPORTED)
set_target_properties(test-lib PROPERTIES IMPORTED_LOCATION
${PROJECT_SOURCE_DIR}/../jniLibs/${ANDROID_ABI}/libxnet-android.so)
include_directories(${PROJECT_SOURCE_DIR}/include)
target_link_libraries( native-lib test-lib ${log-lib})
```



# 使用 Android aar 库

## 引用稳定版

在应用模块的 build.gradle 中配置版本号,具体版本号请参见 版本号列表。

```
dependencies {
    implementation 'com.tencent.iot.video:video-link-android:*.*.*'
}
```

# 引用 SNAPSHOT 版

1. 在工程的 build.gradle 中配置仓库 url。



2. 在应用模块的 build.gradle 中配置版本号。

```
dependencies {
    implementation 'com.tencent.iot.video:video-link-android:*.*.*-SNAPSHOT'
}
① 说明:
建议使用稳定版本, SNAPSHOT 版仅供开发自测使用。
```

# SDK 接入指南

具体请参见 SDK 接入指引。

# App 接入 SDK

第三方 App 在接入 Video SDK 时,**建议将 secretId 和 secretKey 保存到自建后台,并请务必保证不被泄露**;而 xp2p info 需要 App 侧从本地的业 务后台获取,获取到 xp2p info 后,可以通过上述的 startServiceWithXp2pInfo 接口将该 info 传给 SDK,示例代码如下:





# iOS 应用端接入手册

# 快速开始

最近更新时间: 2021-08-03 16:17:43

本文为您介绍 iOS 应用端通过 C++ .a 库和 iOS 库方法集成 SDK。

# 使用 C++ .a 库方法

1. 库 git 地址。

2. 工程引用步骤如下:将所有 .a 与 AppWrapper.h 文件加入工程中,同时加入 libc++,libsqlite3 和 libz 系统库。

# 使用 iOS 库方法

- 1. 库 git 地址。
- 2. 工程引用步骤如下:
  - 2.1 安装 Xcode 开发工具。
  - 2.2 集成 SDK 方式。

执行如下命令采用 Cocoapods 方式集成。

pod 'TIoTLinkVideo'

# SDK 接入指南

具体请参见 SDK 接入指引。



# 错误码

最近更新时间: 2021-06-22 11:40:21

# 错误码

错误码	十进制值	含义
XP2P_ERR_NONE	0	成功。
XP2P_ERR_INIT_PRM	-1000	入参为空。
XP2P_ERR_GET_XP2PINFO	-1001	SDK 内部请求 XP2P info 失败。
XP2P_ERR_PROXY_INIT	-1002	本地 P2P 代理初始化失败。
XP2P_ERR_UNINIT	-1003	数据接收或发送服务未初始化。
XP2P_ERR_ENCRYPT	-1004	数据加密失败。
XP2P_ERR_TIMEOUT	-1005	请求超时。
XP2P_ERR_REQUEST_FAIL	-1006	请求错误。
XP2P_ERR_VERSION	-1007	设备版本过低。
XP2P_ERR_APPLICATION	-1008	服务 application 初始化失败。
XP2P_ERR_REQUEST	-1009	服务 request 初始化失败。
XP2P_ERR_DETECT_NOTREADY	-1010	P2P 探测未完成。
XP2P_ERR_P2P_ININED	-1011	当前 ld 对应的 P2P 已完成初始化。
XP2P_ERR_P2P_UNININ	-1012	当前 ld 对应的 P2P 未初始化。
XP2P_ERR_NEW_MEMERY	-1013	内存申请失败。
XP2P_ERR_XP2PINFO_RULE	-1014	获取到的 XP2P info 格式错误。
XP2P_ERR_XP2PINFO_DECRYPT	-1015	获取到的 XP2P info 解码失败。
XP2P_ERR_PROXY_LISTEN	-1016	本地代理监听端口失败。
XP2P_ERR_CLOUD_EMPTY	-1017	云端返回空数据。
XP2P_ERR_JSON_PARSE	-1018	JSON 解析失败。
XP2P_ERR_SERVICE_NOTRUN	-1019	当前 ld 对应的服务没有在运行。
XP2P_ERR_CLIENT_NULL	-1020	从 map 中取出的 client 为空。

# 解决方案

# P2P 初始化接口失败

• 当前传入的 ld 已经完成了 P2P 初始化,返回错误码: XP2P\_ERR\_P2P\_ININED ,需要先销毁 P2P 资源或使用新 ld,日志如下:

p2p service is running with id:cam01, please stop it first

<sup>•</sup> 向云端请求 XP2P info 时云端回复数据为空,返回错误码: XP2P\_ERR\_GET\_XP2PINFO 。需排查设置的设备三元组和云 API 账号信息是否正确,日 志如下:



quest xp2p\_info failed, errmsg:empty reply from cloud

 向云端请求 XP2P info 时云端回复数据中没有指定 JSON 字段,返回错误码: XP2P\_ERR\_GET\_XP2PINFO 。需排查设备是否上报了 XP2P info 到 云端,日志如下:

request xp2p\_info failed, errmsg:parse reply error

• 请求到的设备 XP2P info 格式错误,返回错误码: XP2P\_ERR\_XP2PINFO\_RULE 。一般为设备 SDK 版本过低所致,日志如下:

emote xp2p\_info rule wrong:\$xp2p\_info

• 设备 SDK 版本与 App SDK 版本不匹配,返回错误码: XP2P\_ERR\_VERSION 。需升级设备 SDK, 日志如下:

The xp2p\_device\_sdk is low, Please upgrade the device version to at least \$version

• 解码获取到的设备 XP2P info 失败,返回错误码: XP2P\_ERR\_XP2PINFO\_DECRYPT 。需排查传入的设备三元组信息是否正确,日志如下:

decrypt xp2p\_info error

• 获取到的设备 XP2P info 信息无效,返回错误码: XP2P\_ERR\_PROXY\_INIT 。需检查设备端网络,确保网络正常,日志如下:

emote xp2pinfo is invalid

• 本地代理无法监听 TCP 端口,返回错误码: XP2P\_ERR\_PROXY\_INIT 。需检查 App 端网络,确保网络正常,日志如下:

proxy listen failed! 、、或 、、shell cannot listen a port

#### 启动数据传输服务失败

• P2P 未成功初始化便启动数据传输服务,返回错误码: XP2P\_ERR\_P2P\_UNININ 。需确保 P2P 初始化成功后再进行后续操作,日志如下:

p2p service is not running with id:cam01, please run it first

• P2P 探测未完成便启动数据传输服务,返回错误码: XP2P\_ERR\_DETECT\_NOTREADY 。需等待 ready 回调触发后再进行后续操作,日志如下:

p2p detect is not ready, state:0

• 创建 Application 失败。需检查 App 网络环境,确保网络正常,日志如下:

create AudioStream application failed

• 创建 Request 失败。需检查 App 网络环境,确保网络正常,日志如下:

create AudioStream request failed

#### 语音数据发送失败

• 语音发送服务未启动,返回错误码: XP2P\_ERR\_UNINIT 。需先启动语音发送接口,日志如下:



connot found request with service:AudioStream

• 语音服务已关闭,返回错误码: XP2P\_ERR\_UNINIT 。需重新启动语音发送服务或停止语音发送接口调用,日志如下:

application is invalid

#### 无法收到回调消息

需要注册回调函数到 SDK。

# 小程序应用端接入手册

最近更新时间: 2023-12-29 16:37:21

# 介绍

通过腾讯云 IoT Video 小程序 P2P 服务,引入 IoT Video X-P2P 插件和 P2P-Player 插件,可实现摄像头和小程序直接打洞传输视频流;配合云端 的 Server SDK,可实现小程序和小程序,小程序和 App 之间的数据共享。

# 准备工作

- 申请腾讯云 IoT Video P2P 服务,获取访问密钥(联调阶段可直接使用 demo 里的密钥,正式发布时请使用我们邮件提供给您的正式密钥)。
- 向腾讯云 IoT Video 团队申请使用 IoT Video X-P2P 插件 和 IoT Video P2P-Player 插件。
- 有使用 live-player 的权限,详见 官方文档。
- 如果使用 1v多 模式,需要将 flv 流的域名加到小程序的 request 合法域名 和 tcp 合法域名 配置中,详见 服务器域名配置官方文档。

## 微信版本限制

微信 8.0.10 以上,基础库 2.19.3 以上。

# 开发指引

# 开发文档

- IoT Video X-P2P 插件开发文档
- IoT Video P2P-Player 插件开发文档
- Demo 源码

# 🕛 说明:

- 若未申请过使用物联网智能视频服务(消费版),请先在物联网智能视频服务(消费版)申请单登记相关信息,申请产品的使用。
- 若已与售前架构师对接使用物联网智能视频服务(消费版),请先联系产品经理授权,后方可访问此 Demo 源码。

原理介绍





#### • 0: 初始化阶段:

小程序在启动时就可以调用 IoT Video P2P 插件的 init 接口,初始化 p2p 模块,入参填写腾讯云 IoT Video 分配的 appKey 和 appKeySecret 等信息。

• 1: 小程序用户选择某个摄像头。

1-1: 小程序引用 IoT Video P2P-Player, 返回 livePlayerContext , 可以直接对 live-player 的 context 进行操作。

1-2: 如果是1v1,小程序调用 IoT Video P2P 插件的 startP2PService 接口,开始建立 p2p 连接。

- 2: 小程序利用 livePlayerContext 触发播放。
- 3: 插件启动播放。
  - 3-1: P2P-Player 插件抛出拉流事件 playerStartPull 给小程序应用。

3-2:小程序应用调用 XP2P 插件的 startStream 接口,传入需要播放的摄像头 ID 和播放 URL,并设置消息接收回调和数据接收回调。

- 4: 数据流会通过第3步设置的数据接收回调,传递给 P2P-Player 插件播放。
- 5: 小程序用户停止播放。
- 6: 插件终止播放。

6-1: 小程序调用 P2P 插件的 stopStream、stopServiceByld 终止传输数据。

6-2: 小程序 Demo 或自有小程序操作 live-player 的 context,停止播放。

•7:消耗。

小程序退出时,调用 IoT Video P2P 插件的 destroy 接口,销毁 p2p 模块。

# 小程序 Demo

#### Demo 地址

• 源码



#### 🕛 说明:

- 若未申请过使用物联网智能视频服务(消费版),请先在物联网智能视频服务(消费版)申请单登记相关信息,申请产品的使用。
- 若已与售前架构师对接使用物联网智能视频服务(消费版),请先联系产品经理授权,后方可访问此 Demo 源码。
- 该体验版二维码对应 3.x.x 版本的 xp2p 插件。

#### Demo 使用

## ▲ 注意:

Demo UI 交互可能更新,但主要流程不变。

1. 进入主页面。



○ "X-P2P Demo IPC" 演示1V1 P2P 直连摄像头场景。

- "X-P2P Demo 1vN-xntp" 和 "X-P2P Demo 1vN-tcp" 演示 1V 多 P2P 场景。
- "多播放器" 演示多播放器的调用。



## 2. 1V1 P2P 直连摄像头场景:

8:59		::: † <b>63</b>	9:00		:::
<	X-P2P Demo	••• 0	<	X-P2P Demo	•••   0
initModule moduleState:	destroyModule re inited	esetP2P	mode: ipc productld:		
player: iot-p	e: 25Q9LF63g6S1WG o2p–player	10	deviceName: sp01_32820	237_10	
			xp2pInfo: 25QPD2ozz( flvFile:	Dp8RLMpLa	
			ipc.flv?actior	startPlay stopPlay	stopAll
mode: ipc productld:			语音 <sup>不加密对讲</sup> 信令	加密对讲 挂断	
AQTV2839G deviceName:	71		command: action=user_	_define&cmd=xxx	
sp01_32820 xp2pInfo: 25QPD2ozz	237_10 0p8RLMpLa		sendCommand 状态		
flvFile: ipc.flv?actio	n=live		targetld: flvUrl:		
prepare	startPlay stopPlay	stopAll	log:		

○ 如果 x−p2p 插件还未初始化,先单击 initModule 初始化。

- 填写 "productId"、"deviceName"、"xp2pInfo", xp2pInfo 即设备属性中的 "sys\_p2p\_info" 字段,可通过 获取设备属性数据的 API 获 取。
- "flvFile"字段可设置播放清晰度,取值参考信令交互文档。
- 单击 prepare 和 startPlay 即可开始播放。
- 单击**不加密对讲**和**加密对讲**可演示小程序语音对讲,单击**挂断**停止对讲。
- 修改信令 command 的 "cmd=xxx" 可演示自定义信令。



腾讯云

# APP 应用端接入手册

最近更新时间:2024-07-11 10:35:11

本文介绍开发者如何基于物联网开发平台 APP SDK(下称 SDK)构建自主品牌 APP,通过 SDK 使用平台提供的用户管理,设备管理等能力。

# App 源码获取

- iOS 版本 App 可通过 腾讯连连-iOS 下载源码。
- Android 版本 App 可通过 腾讯连连-Android 下载源码。

#### 创建应用

- 1. 登录物联网智能视频服务(消费版)控制台,在左侧导航栏选择应用开发。
- 2. 单击新建应用,创建应用成功后,即可获取系统自动生成的 AppKey 与 AppSecret。

← 001				
	应用信息			
	App ID wsE To			
	应用名称 001			
	应用描述 -			
	创建时间 2022-11-08 14:48:27			
	Android应用 ①	iOS应用()		
	APP Key	APP Key		
	APP Secret	APP Secret		
	小程序应用 ① 去授权 部署发布			
	APP Key			
	APP Secret			

3. 将App Key 写入配置文件,将 App Secret 保存在 自建后台,将 appapi调用由设备端发起切换为由自建后台服务发起。

# 搭建后台服务

将 App API 调用由设备端发起切换为由自建后台服务发起,关于应用端 API,请参见 应用端 API 简介 。

#### ▲ 注意:

- 登录前所使用的 API URL 为 https://iot.cloud.tencent.com/api/exploreropen/appapi , 不建议在设备端调用,需要替换为自建的后台服务,以避免密钥的泄漏。
- api/studioapp/\* 为公版 App 专用, OEM 的 App 使用的是应用端 API(api/exploreropen/),当在 App 参数写入配置文件中配置 TencentlotLinkAppkey 后,api/studioapp 调用将自动切换为应用端 API 调用。
- App API (api/exploreropen/appapi)请在自建后台进行调用,Token API (api/exploreropen/tokenapi)可安全在设备端调用。

• iOS 版本 可通过 TIoTAppEnvironment.m 的 selectEnvironmentType 方法中设置此 API。

登录前所使用的 API URL 在 environment.oemAppApi 配置,请务必替换成自建的后台服务地址。







# // 请在 [environment setEnvironment]; 之后设置 oemAppApi 以免被覆盖。 environment.oemAppApi = @"需要替换为自建后台服务地址"; } TIoTCoreAppEnvironment.m 的 setEnvironment方法中默认 API 配置说明。 - (void) setEnvironment { //公版:开源体验版使用 当在 app-config.json 中配置 TencentIotLinkAppkey TencentIotLinkAppSecret 后,将自 动切换为 OEM 版本。 self.studioBaseUrl = @"https://iot.cloud.tencent.com/api/studioapp"; self.studioBaseUrlForLogined = @"https://iot.cloud.tencent.com/api/studioapp/tokenapi"; //OEM App 使用 self.oemAppApi = @"https://iot.cloud.tencent.com/api/exploreropen/appapi"; // 需要在 TIoTAppEnvironment.m 的 -selectEnvironmentType: 中替换为自建后台服务地址。 self.oemTokenApi = @"https://iot.cloud.tencent.com/api/exploreropen/tokenapi"; // 可安全在设备端调 R。 self.wsUrl = @"wss://iot.cloud.tencent.com/explorer"; self.hSUrl = @"https://iot.cloud.tencent.com/explorer-h5"; }

#### • Android 版本 可通过 HttpRequest 中设置API。

登录前所使用的 API URL 在 OEM\_APP\_API 配置,请务必替换成自建的后台服务地址。

接口请求文件	
class HttpRequest private constructor() {	
companion object {	
// <b>公版&amp;开源体验版使用 当在</b> app-config.json <b>中配置</b> TencentIotLinkAppkey TencentIotLinkAppSecret 后,将	
自动切换为 ОЕМ 版本。	
const val STUDIO_BASE_URL = "https://iot.cloud.tencent.com/api/studioapp"	
const val STUDIO_BASE_URL_FOR_LOGINED = "https://iot.cloud.tencent.com/api/studioapp/tokenapi"	
// OEM App <b>使用</b>	
const <b>val OEM_APP_API = "</b> https://iot.cloud.tencent.com/api/exploreropen/appapi" // <b>需要替换为自建后</b>	
台服务地址	
const val OEM_TOKEN_API = "https://iot.cloud.tencent.com/api/exploreropen/tokenapi" // 可安全在设	
备端调用。	
<pre>const val APP_COS_AUTH = "https://iot.cloud.tencent.com/api/studioapp/AppCosAuth"</pre>	
const val BUSI_APP = "studioapp"	
const val BUSI_OPENSOURCE = "studioappOpensource"	

# App Demo、SDK Demo 和 SDK 的关系

#### iOS

工程中已经包含 App Demo、SDK Demo 和 SDK,无需额外引入。

## Android

工程中已经包含 App Demo 、SDK Demo 和 SDK,无需额外引入。

# 小程序应用端接入手册

最近更新时间: 2024-11-26 15:09:22

本文介绍开发者如何基于物联网开发平台小程序 SDK (下称 SDK )构建自主品牌小程序,通过 SDK 使用平台提供用户管理,设备管理等能力。

# 前提条件

> 腾讯云

- 1. 使用 SDK 需要您的运行环境包含 Node.js 以及 npm。
- 2. 登录物联网智能视频服务(消费版),进入应用开发。
- 3. 新建应用,创建应用成功后,即可获取系统自动生成的 AppKey 与 AppSecret。

001		
	应用信息	
	App ID wsElzuXIHKFe I	
	应用答称 001	
	应用描述 -	
	创建时间 2022-11-08 14:48:27	
	Android应用①	iOS应用 ①
	APP Key	APP Key
	APP Secret	APP Secret
	小程序应用 ① 去授权 部署发布	
	APP Key	
	APP Secret	

# 接入 SDK

# 安装 SDK

1. 在微信小程序目录下,通过 npm 安装 SDK 。

npm install qcloud-iotexplorer-appdev-sd}



2. 在 微信开发者工具 的项目界面中,单击界面右上角的详情,选择本地设置,勾选"使用 npm 模块"。

				-		×
		止 上传	い 版本管理		三	
基本信息	本地设置	项目	配置		腾讯云	
调试基础库 ⑦		2.10.1	•	•	推送	
该基础库支持微信客	户端					
iOS			7.0	.9 及니	以上版本	
Android			7.0	.9 及L	以上版本	
MacOS				音	雪不支持	
Windows				音	雪不支持	
<ul> <li>✓ ES6 转 ES5</li> <li>✓ 增强编译</li> <li>✓ 使用 npm 模</li> </ul>	块					
🗌 上传代码时样	試自动补全					

3. 在微信开发者工具的项目界面中,选择菜单栏的工具 > 构建 npm,构建成功后界面提示完成构建。



#### 配置小程序服务器域名

小程序 SDK 通过以下域名连接到物联网开发平台。

- request 域名: https://iot.cloud.tencent.com
- socket 域名: wss://iot.cloud.tencent.com

在接入小程序 SDK 时,需要将上述域名添加到小程序的服务器域名列表中,步骤如下。



1. 登录小程序后台。

#### 2. 选择**开发管理 > 开发设置**。

✔ 小程序	
▲ 首页	开发管理
□ 管理	运维中心 监控告警 开发设置 接口设置 安全中心
版本管理 成员管理 用户反馈	开发者ID
♥ 统计	开发者ID
■■ 功能	AppID(小程序ID)
微信搜一搜 客服 订阅消息	AppSecret(小强序密钥)
↓□□IPAge入 	小程序代码上传 开发者可基于配置信息调用微信开发者工具提供的代码
开发管理 开发工具	配置信息

3. 在"服务器域名"页面单击**开始配置**(若曾经配置过则单击修改),根据页面指引完成身份确认。

ţ	服务器域名
	尚未配置服务器信息,查看小程序域名介绍 使用官方推出的小程序云开发,无需服务器及域名配置即可上线小程序,立即开通 云开发 如霜购买服务器资源及域名,可 前往腾讯云购买。
	开始配置

#### 4. 填写服务器信息。

配置服务器信息		$\times$
	<ol> <li>身份认证</li> <li>2 配置服务器信息</li> </ol>	
如需购买服务器资源及域	名,可 <b>前往腾讯云</b> 购买。或开通 云开发,无需服务器及域名配置即可上线小	程序。
request合法域名	https://iot.cloud.tencent.com;	
socket合法域名	wss://iot.cloud.tencent.com;	
〇 <b>request 合法域名:</b> https://	://iot.cloud.tencent.com	

- O socket 合法域名: wss://iot.cloud.tencent.com
- 5. 单击**保存并提交**。
- 接入微信登录



使用物联网开发平台的 微<del>信号注册登录</del> 应用端 API,可以让小程序用户通过微信注册登录到物联网开发平台。出于安全考虑,不建议在小程序端直接调用 微信号注册登录 API,请在自建的后端服务调用该 API,以避免密钥的泄露。 小程序接入微信登录需要实现以下流程:

- 1. 小程序调用 wx.login 获取临时登录凭证 code 。
- 2. 小程序将临时登录凭证 code 传递给自建的后端服务。
- 3. 后端服务调用微信服务端 API auth.code2Session ,通过临时登录凭证 code 获取小程序用户的 OpenID。
- 4. 后端服务调用物联网开发平台应用端 API 微信号注册登录 ,通过小程序用户的 OpenID 获取 AccessToken。
- 5. 后端服务将 AccessToken 返回给小程序。
- 6. 小程序将 AccessToken 传入小程序 SDK 以完成小程序 SDK 的初始化。

#### 示例代码

请按照以下步骤并参考示例代码以实现接入微信登录:

- 1. 在 AppDevSdk 的构造函数参数中,将 appKey 的值修改为您实际获取的 AppKey。
- 2. 补充 getAccessToken 函数的实现。在 getAccessToken 函数中,您需要调用自建后端服务的登录接口,传入登录所需的信息,并将获取到的 AccessToken 放入 getAccessToken 函数的返回值。

```
appKey: '此处填写您的 AppKey',
 获取小程序登录凭证 c
// 调用开发者自建后端服务,传入 code,获取物联网开发平台的 AccessToken
  url: '开发者自建后端服务的 URL',
    nickName: '新用户的默认昵称',
    avatarUrl: '新用户的默认头像 URL'
```



<pre>resolve(res.data);</pre>			
fail(err) {			
reject (err) ;			
Token: resp.Token,			
ExpireAt: resp.ExpireAt,			

# 使用 SDK

## 调用应用端 API

应用端 API 是物联网开发平台为了满足智能家居场景,为用户开发自有品牌的小程序或 App 而提供的云端服务,用户无需实现用户管理、设备管理、设备 定时、家庭管理等基础能力,可通过调用应用端 API 快速完成移动应用端的开发。关于应用端 API 的更多信息,请参见 应用端 API 简介。 SDK 对应用端 API 的调用过程进行了封装,发送请求时会自动带上公共参数 AccessToken 与 RequestId 。 以下示例代码以调用 获取用户绑定设备列表 应用端 API 为例。

```
sdk.requestApi(
    'AppGetFamilyDeviceList', // 请求应用端 API 的 Action 名
    { FamilyId: 'default' } // 请求应用端 API 的数据
)
    .then(data => {
        // 请求成功
        console.log(data);
    })
    .catch(err => {
        // 请求失败
        console.error(err);
    });
```

## 控制设备

通过调用 用户控制设备 应用端 API,可以对用户已绑定的设备发起控制操作。

```
// 指定要控制的设备的属性数据
const deviceData = {
    light_switch: 0,
};
sdk.requestApi('AppControlDeviceData', {
    ProductId: '要控制的设备的产品 ID',
    DeviceName: '要控制的设备的设备名称',
    Data; JSON.stringify(deviceData) // 控制报文 JSON
})
.then(data => {
    // 请求成功
    console.log(data);
```



```
// .catch(err => {
    // 请求失败
    console.error(err);
});
```

#### 长连接通信能力

SDK 支持通过 WebSocket 为用户订阅所绑定设备的上报数据以及状态信息,SDK 初始化时默认自动连接 WebSocket 服务端。

```
通过设备 ID 列表订阅
```

```
sdk.subscribeDevices([
   'Product1/Device1',
   'Product1/Device2',
   'Product2/Device3'
]);
```

#### 通过设备列表订阅

subscribeDevices 函数支持传入设备列表(如 获取用户绑定设备列表 应用端 API 返回的设备列表)以订阅设备信息,要求参数数组中的设备信息包 含 DeviceId 字段。

```
sdk.requestApi('AppGetFamilyDeviceList', { FamilyId: 'default' })
.then(data => {
   sdk.subscribeDevices(data.DeviceList);
})
.catch(err => {
   // 请求失败
   console.error(err);
});
```

#### 监听设备上报数据及状态信息

调用 SDK 的 监听事件 接口,可以监听 WebSocket 订阅的设备上报数据及状态信息,请参照以下示例代码。

```
const { EventTypes } = require('qcloud-iotexplorer-appdev-sdk').AppDevSdk.
// 监听设备上报数据推送
sdk.on(EventTypes.WsReport, ({ deviceId, deviceData }) => {
   console.log('websocket device report', deviceId, deviceData);
});
// 监听设备在线状态变更推送
sdk.on(EventTypes.WsStatusChange, ({ deviceId, deviceStatus }) => {
   console.log('websocket device status change', deviceId, deviceStatus);
});
// 监听设备控制推送
sdk.on(EventTypes.WsControl, ({ deviceId, deviceData }) => {
   console.log('websocket device control', deviceData }) => {
   console.log('websocket device control', deviceData }) => {
   console.log('websocket device control', deviceId, deviceData);
});
```

#### 获取用户基本信息

通过读取 SDK 对象上的属性,可以获取用户的基本信息,请参照以下示例代码。



// sdk.isLogin: boolean; **用户是否登录** console.log(sdk.isLogin);

// sdk.userId: string; 用户 ID console.log(sdk.userId);

// sdk.nickName: string; 用户昵称 console.log(sdk.nickName);

// sdk.userInfo: UserInfo; 用户详细信息
console.log(sdk.userInfo);
console.log(sdk.userInfo.Avatar);

// sdk.getDefaultFamilyId: () => Promise< string >; 获取用户默认的家庭 ID sdk.getDefaultFamilyId().then(familyId => console.log(familyId));

#### UserInfo 数据结构说明

属性名	属性描述	类型
Avatar	头像	string
CountryCode	国家代码	string
Email	邮箱	string
NickName	昵称	string
PhoneNumber	电话号码	string
UserID	用户 ID	string

# **SDK Demo**

• Demo 项目 GitHub 地址。

#### () 说明:

- 若未申请过使用物联网智能视频服务(消费版),请先在物联网智能视频服务(消费版)申请单登记相关信息,申请产品的使用。
- 若已与售前架构师对接使用物联网智能视频服务(消费版),请先联系产品经理授权,后方可访问此 Demo 源码 。

• Demo 使用指引:请参见 快速入门。



# 应用端 API 列表

最近更新时间: 2023-07-06 10:01:13

# 简介

本文档为您介绍应用端 API 相关使用内容。

# 用户管理

# 相关接口

接口名称	备注
微信号注册登录	用于微信用户注册登录,获取开发平台的用户访问 AccessToken,首次调用时,自动为该微信号注册对应 账号 。
手机号注册用户	提供手机号码方式的用户注册。
随机发送手机短信	用于手机号注册、绑定、重置密码时,发送验证码。
邮箱账号注册用户	提供邮箱账号方式注册用户。
随机发送邮箱验证码	用于邮箱注册、绑定、重置密码和登录时,发送验证码。
手机号、邮箱账号登录	用于手机号码、邮箱账号登录,获取用户访问 AccessToken。
使用手机号重置密码	用于使用手机号重置密码。
使用邮箱重置密码	用于使用邮箱重置密码。
用户注销	用于用户退出登录态。
修改用户信息	用于修改用户信息。

# 配网管理

# 相关接口

接口名称	备注
生成 Wi-Fi 设备配网 Token	用于生成 Wi-Fi 配网任务的随机 Token。
查询配网 Token 状态	用于查询配网 Token 的当前状态。
用户绑定 Wi-Fi 设备	用于小程序或 App 用户绑定 Wi-Fi 类设备。

# 设备管理

# 相关接口

接口名称	备注
获取产品信息	获取产品信息。
用户删除设备	为用户提供删除设备的功能,删除后用户需要重新配网进行绑定。
获取用户绑定设备列表	用于获取用户已绑定设备列表。
获取设备当前状态	用于查询设备状态。
修改设备名称	用于修改设备名称。



获取设备详情	用于查询设备详情。
设备更换房间	设备更换房间。
配网绑定设备	用于小程序或 App 进行 Wi-Fi 设备配网绑定操作。

# 设备控制

# 相关接口

接口名称	备注
用户控制设备	用于用户对绑定的设备下发属性物模型数据,实现设备远程控制。
同步调用设备行为	用于用户向已成功绑定的设备同步下发行为物模型请求,实现设备远程控制。
异步调用设备行为	用于向用户所绑定的设备异步下发行为物模型请求,实现设备远程控制。

# 数据查询

# 相关接口

接口名称	备注
获取设备物模型数据	用于获取设备物模型数据,可用于【_sys_xp2p_info】的获取。
获取设备物模型历史数据	用于获取设备物模型历史数据。
获取设备的历史事件	用于获取设备的历史事件。

# 云存服务

接口名称	备注
拉取云存事件列表	用于获取云存时间列表。
拉取云存事件缩略图	用于拉取云存事件缩略图地址。
获取具有云存的日期	用于获取具有云存的日期。
获取某一天云存时间轴	用于获取某一天云存时间轴。
获取视频防盗链播放 URL	用于获取视频防盗链播放 URL。
拉取图片流数据	用于拉取图片流数据。

# 设备分享

接口名称	备注
App 端发送设备分享	用于发送设备分享邀请。
获取设备分享 Token	用于获取设备分享 Token。
获取设备分享 Token 信息	用于获取设备分享 Token 信息。
绑定用户分享的设备	用于绑定用户分享的设备。
查询用户分享设备列表	用于查询用户分享的设备列表。
删除用户分享的设备	用于删除用户分享的设备。
查询设备的用户列表	用于查询设备用户列表。



删除设备分享的用户

用于删除设备分享的用户。

# 家庭管理

接口名称	备注
创建家庭	用于创建家庭。
删除家庭	用于删除家庭。
修改家庭	用于修改家庭。
获取家庭列表	用于获取家庭列表。
获取家庭详情	用于获取家庭详情。
新建房间	用于新建房间。
修改房间	用于修改房间。
删除房间	用于删除房间。
获取房间列表	用于获取房间列表。
邀请家庭成员	用于向微信好友发送邀请加入家庭请求。
成员加入家庭	用于成员加入家庭。
删除家庭成员	用于管理员删除家庭成员。
成员退出家庭	用于成员主动退出某个家庭。
获取家庭成员列表	用于获取家庭成员列表。
App 端邀请家庭成员	用于发送家庭分享邀请。

# 固件升级

接口名称	备注
查询设备固件是否升级	用于查询设备固件升级信息。
确认固件升级任务	用于用户确认升级后,云端向设备发起固件升级请求。
查询设备固件升级状态	用于查询设备固件升级状态及进度。

# 长连接通信

接口名称	备注
注册监听	用于向 WS 服务端进行订阅,订阅成功后可以通过 ws.onmessage 的监听获取到 WS 服务端实时推送设 备上下线状态及属性数据。
心跳	设备保活,通过该接口更新设备时间戳,保持设备监听状态,同时保持 WebSocket 连接。可以多次调用, 推荐每60s调用一次。
设备状态推送	设备状态推送用于在小程序或 App 实时获取用户绑定设备的上下线状态、设备上报的属性与事件以及设备行 为执行结果。需要成功调用注册设备监听接口,通过监听 WebSocket 的 OnMessage 获取设备状态与属 性值的实时推送,若获取到的 event.data.push 为 true,则代表该条消息为设备状态变更的主动推送。