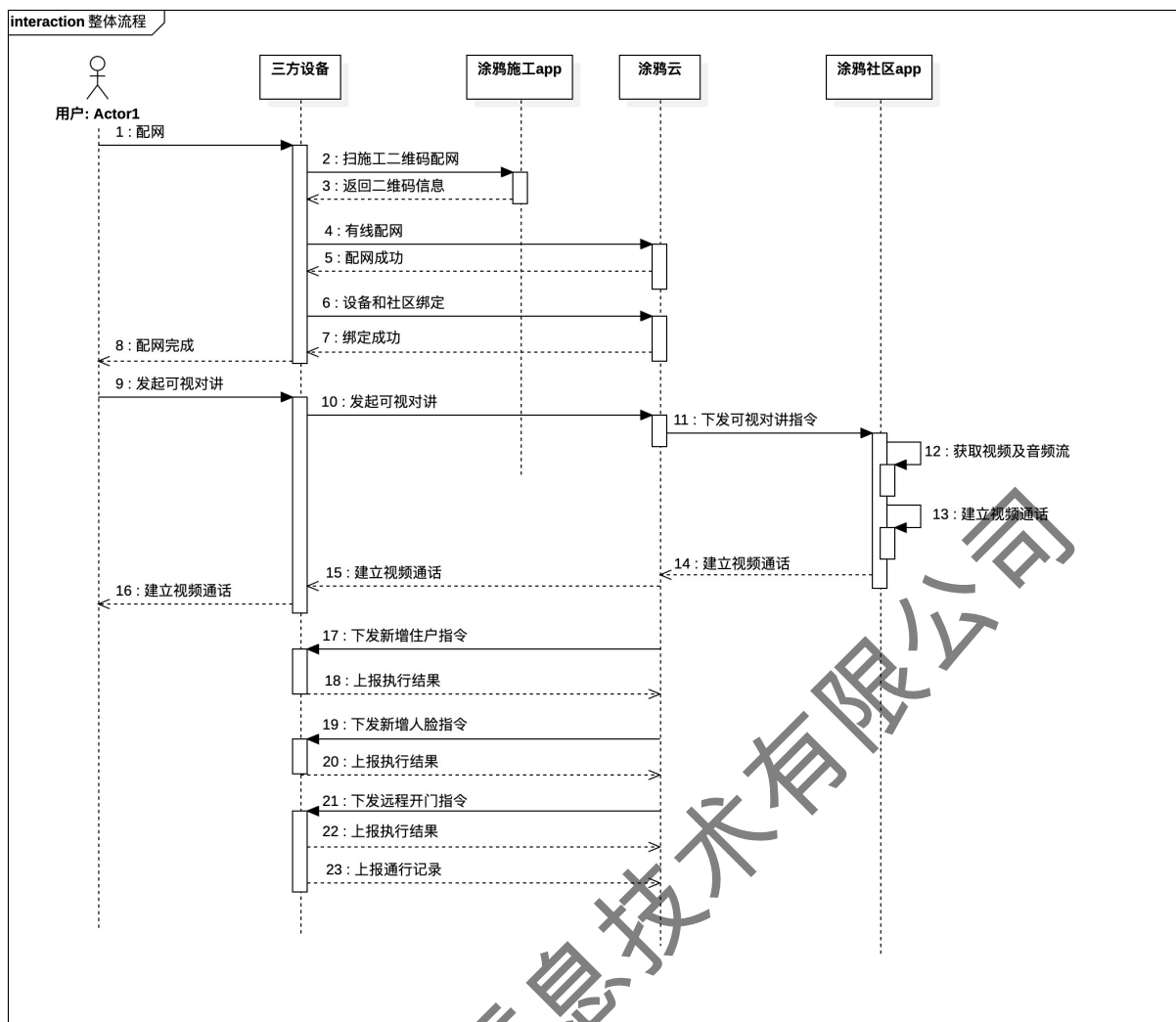


# 涂鸦可视门禁接入API

## 1. 关键术语

序号	术语	说明
1	cid	三方设备id,设备的唯一编号
2	sn	下发dp点的唯一编号
3	atop接口	涂鸦对外提供的https接口
4	下发	涂鸦云端指令通过mqtt消息通知给设备
5	上报	设备端消息通过atop接口或mqtt消息通知给涂鸦云

## 2. 整体流程



## 3. 对接流程

### 3.1. 依赖配置

```
implementation 'com.squareup.okhttp3:okhttp:3.14.0'
```

### 3.2. 下载涂鸦智能app

iOS: <https://testapp.tuya.com/tuyaSmart?version=&id=1237645658022772736>

安卓: <https://testapp.tuya.com/tuyaSmart?version=&id=1236845822000267264>

### 3.3. 三方设备引入涂鸦EDGE IPC SDK

涂鸦EDGE IPC SDK及demo地址:

[https://github.com/TuyaInc/tuyasmart\\_android\\_saas\\_edge\\_ipc\\_demo](https://github.com/TuyaInc/tuyasmart_android_saas_edge_ipc_demo)

EDGE IPC SDK基于涂鸦IPC SDK开发,涂鸦IPC SDK文档地址:

## 3.4. 进行设备配网,初始化配置

### 3.4.1. 申请私钥

向涂鸦申请私钥,申请的私钥用来对配网信息进行解密.

### 3.4.2. 设备扫码

设备扫描施工app二维码,进行设备配网.二维码信息格式:

```
{
  "t": "AYRfQAw5nmpr_0",
  "a": "a1-cn.wgine.com",
  "key": "1168813909092601857"
}
```

二维码生成后,10分钟就会过期.

设备根据a、key和cid调用接口,获取二维码的内容并解密,解密结果转为qrcodeMap.

```
//查询配网信息
NetQrcodeVO netQrcodeVO = AtopFacade.getInstance().queryQrcodeInfo(a, key,
cid);
//对配网信息进行解密
String qrcodeInfo = AESUtils.decrypt(netQrcodeVO.getData(), secret);

//组装qrcodeMap;
Map<String, String> qrcodeMap = JSON.parseObject(qrcodeInfo, new
TypeReference<HashMap<String, String>>() {});
qrcodeMap.put(QrcodeEnum.TOKEN.getCode(), t);
```

### 3.4.3. 初始化SDK

```
/**
 * 扫二维码配网及设备重启时调用
 *
 * @param ctx          系统上下文
 * @param cid          三方设备id,不同的设备唯一
 * @param qrcodeMap    二维码信息Map,见1.3.2
 * @param basePath     可写的一个路径,用于存储SDK相关的配置,建议app存
                        储目录.
 * @param recordPath   可写的一个路径,用于存储录像,建议sd卡.
 * @param properties   实现类的配置
 * @param paramConfigCallback 音视频参数设置回调
```

```
*/  
public void initSDK(Context ctx, String cid, Map<String,String> qrCodeMap,  
    String basePath, String recordPath, Properties properties,  
    MediaParamConfigCallback paramConfigCallback)
```

qrCodeMap的key参见QrCodeEnum的code, 其中t、pId、uuid、authKey在配网成功后,需要写入系统.保证设备重启、断电或恢复出厂设置后,还能重新获取,从而连上涂鸦云.

判断设备是否配网成功:

```
SharedPreferences sp =  
ctx.getSharedPreferences("edge_config",Context.MODE_PRIVATE);  
boolean isBind = sp.getBoolean("bind_status",false);
```

如果isBind = true,即为配网成功.

### 3.4.4. 初始化门禁接口实现类

properties的key参见下面的示例, value中的值是由三方具体实现的接口类, 初始化后的类才会被调用.

所有需要实现的接口列表参见目录5.

```
Properties properties = new Properties();  
//住户数据同步  
properties.put("dc_userInfo","com.tuya.ai.ipcsdkdemo.edge.TenementReceiveEvent  
Impl");  
//人脸数据同步  
properties.put("dc_faceInfo","com.tuya.ai.ipcsdkdemo.edge.FaceImageReceiveEven  
tImpl");  
//卡数据同步  
properties.put("dc_cardInfo","com.tuya.ai.ipcsdkdemo.edge.CardReceiveEventImpl  
");  
//二维码数据同步  
properties.put("dc_qrCodeInfo","com.tuya.ai.ipcsdkdemo.edge.QcCodeReceiveEvent  
Impl");  
//开门  
properties.put("dc_door","com.tuya.ai.ipcsdkdemo.edge.DoorReceiveEventImpl");
```

## 3.5. 实现接口

IPC SDK已经将涂鸦云下发的指令封装成了接口, 三方只需要实现具体的接口, 接口列表参见目录5.

## 4. 接口API说明

# 4.1. 视频通话

```
//开启视频通话、接听、挂断时调用
AtopFacade.getInstance().sendControlProtocol(controlProtocolRequest);
```

AtopFacade	接口		
sendControlProtocol	方法		
ControlProtocolRequest	请求参数	描述	是否必传
sn	String	sn号,挂断或接听时需要设置.设置的值为呼叫时返回的sn.	否
mode	String	设备模式.如门铃:ac_doorbell	是
type	String	操作类型,见详细说明	是
deviceId	String	涂鸦设备id	是
targetAddress	String	目标地址,格式:01-01-01-01-01-01,表示01小区01苑01幢01单元01层01号房间.没有传00,比如没有苑,苑的位置传00.	是
extend	String	json字符串.通话后挂断:{"beginTime":1592373736,"endTime":1593373736}.beginTime:开始时间,endTime:结束时间.时间类型为long.	否
	返回结果		
	String	sn号	

操作类型详细说明

操作类型	场景	操作
1	设备呼叫app和室内机	设备发送type=1事件
3	设备呼叫app和室内机,设备主动挂断,APP或室内机不操作	设备发送type=3事件
4	设备呼叫app和室内机,APP或室内机未接听,呼叫时间超过设备限定,设备挂断.设备限定呼叫超时时间默认30s	设备发送type=4事件
5	设备呼叫app和室内机,APP或室内机接听后,APP或室内机挂断.sdk会有回调事件给设备.	设备发送type=5事件
6	设备呼叫app和室内机,社区APP或室内机接听.sdk会有回调事件给设备.	设备发送type=6事件
7	设备呼叫app和室内机,APP或室内机接听后,通话时间超过设备限定,设备挂断.设备限定通话超时时间默认90s	设备发送type=7事件

## 4.2. 通行记录上报

//通过该接口上报门禁通行事件

```
MqttEventSupport.getInstance().publishThirdEvent(thirdEventRequest);
```

MqttEventSupport	接口		
publishThirdEvent	方法		
ThirdEventRequest	请求参数	描述	是否必传
uid	String	用户id	是
operTime	String	事件发生时间	是
way	int	开门方式: 1(卡)/2(密码)/3(二维码)/4(人脸)/5(指纹)/9(远程)/10(其它)	是
imageId	String	图片id,获取图片id的接口参见2.4.	否
isSuccess	boolean	是否成功,true:成功,false:失败	是
	返回结果		
	boolean	true:调用成功,false:调用失败	

## 4.3. 上传图片

```
//通过该接口上报图片,上传的图片格式默认为jpg
String imageId = AtopFacade.getInstance().uploadImageFile(fileBytes);
```

AtopFacade	接口		
uploadImageFile	方法		
	请求参数	描述	是否必传
fileBytes	byte[]	图片的字节数组	是
	返回结果		
	String	图片id	

## 5. 待三方实现的接口说明

### 5.1. 新增住户

TenementReceiveEvent	接口		
addTenement	方法		
TenementRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
name	String	涂鸦用户名称	否
idCard	String	证件id	否
phone	String	手机号	否
beginTime	Long	有效期开始时间, 13位时间戳	否
endTime	Long	有效期结束时间, 13位时间戳	否
EventContext	请求参数	描述	是否必传
sn	String	唯一编号	是
DeviceInfo		设备信息	是
gatewayId	String	涂鸦边缘网关设备id	是
deviceId	String	涂鸦设备id	是

deviceName	String	涂鸦设备名称	否
cid	String	三方设备id	是
commType	String	设备接入类型(MQTT,HTTP)	否
deviceIp	String	设备ip	否
macAddress	String	设备mac地址	否
productType	String	品类	否
productId	String	产品id	是
installLocation	String	设备安装地点	否
deviceDesc	String	设备描述	否
activeStatus	String	设备状态 0:未配网,1:已配网	是
extendData	String	扩展信息,json串	否
BaseResult	返回结果	描述	是否必传
success	boolean	是否返回成功,成功:true,失败:false	是
code	String	返回code	否
msg	String	返回msg	是
data	Object	返回数据	否

该接口需要支持幂等,即同一个住户新增成功后,还能再次新增.

## 5.2. 更新住户



<b>TenementReceiveEvent</b>	接口		
modifyTenement	方法		
TenementRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
idCard	String	证件id	否
phone	String	手机号	否
beginTime	Long	有效期开始时间，13位时间戳	否
endTime	Long	有效期结束时间，13位时间戳	否
EventContext	请求参数		
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

### 5.3. 删除住户

<b>TenementReceiveEvent</b>	接口		
removeTenement	方法		
RemoveTenementRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

### 5.4. 住户通行授权

TenementReceiveEvent	接口		
enableTenement	方法		
EnableTenementRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
enable	Integer	启用/禁用. 0:禁用,1:启用	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.5. 新增人脸

FacImageReceiveEvent	接口		
addFacImage	方法		
FacImageRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
faceId	String	人脸id	是
url	String	人脸照片地址	是
EventContext	请求参数		
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

该接口需要支持幂等,即同一个人脸照片新增成功后,还能再次新增.

## 5.6. 更新人脸

<b>FacelImageReceiveEvent</b>	接口		
modifyFacelImage	方法		
FacelImageRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
facelId	String	人脸id	是
url	String	人脸照片地址	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.7. 移除人脸

<b>FacelImageReceiveEvent</b>	接口		
removeFacelImage	方法		
RemoveFacelImageRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
facelId	String	人脸id	否
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.8. 新增门禁卡

<b>CardReceiveEvent</b>	接口		
addCard	方法		
CardRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
cardNo	String	门禁卡号	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.9. 更新门禁卡

<b>CardReceiveEvent</b>	接口		
modifyCard	方法		
CardRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
cardNo	String	门禁卡号	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.10. 移除门禁卡

<b>CardReceiveEvent</b>	接口		
removeCard	方法		
RemoveCardRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.11. 新增二维码

<b>QcCodeReceiveEvent</b>	接口		
addQcCode	方法		
QrCodeRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
qrCode	String	二维码Code	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

## 5.12. 更新二维码

<b>QcCodeReceiveEvent</b>	接口		
modifyQcCode	方法		
QrCodeRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
qrCode	String	二维码Code	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

### 5.13. 移除二维码

<b>QcCodeReceiveEvent</b>	接口		
removeQcCode	方法		
RemoveQrCodeRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

### 5.14. 远程开门

DoorReceiveEvent	接口		
openDoor	方法		
DoorRequest	请求参数	描述	是否必传
uid	String	涂鸦用户id	是
EventContext	请求参数		是
		同新增用户EventContext	
BaseResult	返回结果		
		同新增用户BaseResult	

远程开门不需要校验用户权限.

杭州涂鸦信息技术有限公司