

慧管家门禁接入文档

更新时间	更新人	版本号	更新内容
2019-9-19	卞俊杰	1.1.8.23	第一次更新
2019-9-20	卞俊杰	1.1.8.23	优化了文档框架，新增了安防部分文档

概念介绍

- familyId : 设备与底座配对的唯一id，由底座部署完后提交到服务器
- buttonKey : 用于区分多分机时的设备，对应各自底座中的buttonKey，默认主机是 01
- door : 家庭与室外机（单元门口机，小区围墙机，物业中心管理机等室外机）
- p2p : 户户通(社区间 家庭呼叫家庭)
- ext : 室内通(家庭中 分机呼叫分机)

一 接入

1.项目的build.gradle 中加入仓库url

```
maven { url "https://raw.githubusercontent.com/s-tree/JxRepository/master/releases/" }
```

2.在app的build.gradle 中导入入门禁

```
compile "com.jingxi.smartlife.pad.sdk:doorAccess:1.1.8.23"
compile "com.jingxi.smartlife.pad.sdk:utils:1.0.2"
```

3.导入 fastJson 、 lite-orm 、 gson 库(sdk 内部必要库)

```
compile 'com.alibaba:fastjson:1.2.59'
compile 'com.google.code.gson:gson:2.8.5'
```

4.所需权限

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.CAMERA"/>
```

二 初始化

1.在application 初始化的时候设置全局 application

```
JXContextWrapper.context = application;
```

2.DoorAccessManager 是门禁功能接口类，初始化过程在 getInstance() 中通过构造方法进行，最好预先初始化一次

```
DoorAccessManager.getInstance().init();
```

3.设置回调，具体参考api中的 DoorAccessListener,DoorAccessListUI,DoorAccessConversationUI

```
DoorAccessManager.getInstance().setDoorAccessListener(this);  
DoorAccessManager.getInstance().addSecurityListener(this);  
DoorAccessManager.getInstance().addConversationUIListener(this);
```

3.启动服务

```
DoorAccessManager.getInstance().startFamily(familyId,buttonKey)
```

三 视频通话

1.基本操作

必要session

(1).为该次通话设置Surface

```
DoorAccessManager.getInstance().updateCallWindow(sessionId, surfaceView);
```

(2).开锁

```
DoorAccessManager.getInstance().openDoor(sessionId);
```

(3).录制视频

```
recordSession = DoorAccessManager.getInstance().startRecord(familyID,sessionId);
```

recordSession是该次录制视频的session，在停止录制时需要用到

(4).停止录制视频

```
DoorAccessManager.getInstance().stopRecord(familyID,sessionId,recordSession);
```

(5).挂断通话

```
DoorAccessManager.getInstance().hangupCall(sessionId);
```

2.门禁外呼(door)

(1).获取设备列表

DoorDevice 的设备类型请查看 DoorDevice.getMyDeviceType()

```
DoorAccessManager.getInstance().getDevices(familyID);
```

(2).查看该设备视频/音频

```
String sessionId = DoorAccessManager.getInstance().monitor(familyID,doorDevice);
```

sessionId 是此次会话session，需要保存，之后的操作中需要使用

(3).事件更新，具体参考 DoorAccessConversationUI 及 demo

3.户户通外呼

(1).判断是否支持户户通

```
boolean isSupportP2P = DoorAccessManager.getInstance().isSupportP2P(familyID);
```

(2).生成拨号号码

例如 一栋二单元304室，则生成的外呼号码为 01020304

(3).外呼

```
sessionId = DoorAccessManager.getInstance().monitorP2P(familyID,number,true);
```

(4).事件更新，具体参考 DoorAccessConversationUI 及 demo

4.室内通通话

(1).判断是否支持室内通

```
boolean isSupportExt = DoorAccessManager.getInstance().isSupportExt(familyID);
```

(2).设置室内通设备 备注,设置后需要重启服务才能生效

```
DoorKit.Options options = DoorKit.getOptions();  
options.alias = alias;  
DoorKit.init(options);  
DoorAccessManager.getInstance().unInit();
```

```
//延迟200毫秒执行以下操作
DoorAccessManager.getInstance().init();
DoorAccessManager.getInstance().startFamily(familyID,buttonKey);
```

(3).获取室内通设备

```
List<ExtDeviceBean> extDevices = DoorAccessManager.getInstance().getExtDevices(familyID,buttonKey);
```

需要注意的是，这个方法获取到结果会有重复，需要手动去重

(4).外呼

室内通外呼有两种模式

- 直接呼叫
 - isMonitor == false
呼叫时对方会收到ringing 消息,需要对方接听才可通话
- 监控模式
 - isMonitor == true
呼叫时不需要对方接听，可以监控对方的画面与声音，不会将本机的画面与声音传递过去，同时对方页面应该没有任何反应

```
chatSession = DoorAccessManager.getInstance().callExt(familyID,deviceBean,isMonitor);
```

(5).事件更新，具体参考 DoorAccessConversationUI 及 demo

(6).禁止将本地摄像头发送给对方

```
DoorAccessManager.getInstance().enableLocalToRemoteVideo(familyID,sessionId,false);
```

(7).禁止将本地音频发送给对方

```
DoorAccessManager.getInstance().enableLocalToRemoteAudio(familyID,sessionId,false);
```

(8).禁止对方发来的音频

```
DoorAccessManager.getInstance().enableRemoteToLocalAudio(familyID,sessionId,false);
```

5.收到呼叫

(1).获取呼叫来源(门禁、户户通、室内通)

参考DoorAccessListener.onRinging()

```
int callType = DoorSessionManager.getRingingType(sessionId);
if(callType == DoorEvent.TYPE_DOOR){
    //门禁呼叫
}
```

```
else if(callType == DoorEvent.TYPE_P2P){
    //户户通呼叫
}
else if(callType == DoorEvent.TYPE_EXT){
    //室内通呼叫
}
```

(2).接听

```
DoorAccessManager.getInstance().acceptCall(sessionId);
```

(3).挂断

```
DoorAccessManager.getInstance().hangupCall(sessionId);
```

四.回放记录

1.获取来访纪录

- 外呼时 needSaveRecord 参数传入true 或者 在通话时录制了视频才会有记录
- 接听时 可在 DoorAccessListener.onSnapshotReady() 或者 DoorAccessConversationUI.startTransPort() 中自动开启录制

```
//获取门禁的通话纪录，从0到50（按时间倒序）
DoorAccessManager.getInstance().getHistoryListByType(familyID,DoorRecordBean.RECORD_TYPE_DOOR,0,50);
//获取指定设备的通话纪录，从0到50（按时间倒序）
DoorAccessManager.getInstance().getHistoryListByDevice(familyID,doorDevice,true,0,50);
```

2.获取来访纪录中的视频文件

```
List<RecordVideoBean> recordVideoList = doorRecordBean.getRecordList();
```

3.回放

需要传入播放回调 RecordPlayer.RecordPlayerHandler

```
DoorAccessManager.getInstance().startPlayBack(playHandler,playSession,recordVideoBean.videoPath);
```

4.停止回放

```
DoorAccessManager.getInstance().pausePlayBack(playSession);
```

5.调整回放的进度(progress 是 百分比进度)

```
DoorAccessManager.getInstance().seekPlayBack(playSession,progress);
```

6.设置回放的SurfaceView

```
DoorAccessManager.getInstance().updatePlayBackWindow(playSession,surfaceView);
```

五.安防

1.是否支持安防

```
DoorAccessManager.getInstance().isSupportSecurity(familyID);
```

2.设置安防监听

```
DoorAccessManager.getInstance().addSecurityListener(listener);
```

3.查询安防状态

```
DoorAccessManager.getInstance().querySecurityStatus(familyID);
```

4.切换安防状态

```
DoorAccessManager.getInstance().switchSecurityStatus(familyID);
```

5.关闭安防报警

```
DoorAccessManager.getInstance().cancelSecurityWarning(familyID);
```