奇虎 360 公司

360 智能摄像机

开放平台 AndroidSDK 开发文档

dev.jia.360.cn

V3.2.0

2017/8/1

详细阐述了 360 智能摄像机开放平台 AndroidSDK 的接入流程和开发指导

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

目录

1. 名词解释.....2
2. 开发说明.....2
2.1. 关于 sn 号入库.....2
2.2. 关于开放平台和 360 智能摄像机 APP.....2
2.3. 业务逻辑.....3
3. 准备工作.....3
3.1. Manifest 配置.....3
3.1.1. 添加 manifest 子节点,权限.....3
3.1.2. 添加 application 子节点，app_id、sdk_Key、Service.....4
4. 开始使用 SDK.....5
4.1. 初始化 SDK.....5
4.2. 激活摄像机.....5
4.2.1. 生成携带 wifi 信息的声波文件.....5
4.2.2. 将 wifi 信息通过声波发送给摄像机.....5
4.2.3. 向服务端查询摄像机是否激活上线.....6
4.3. 视频直播.....6
4.3.1. 实时直播.....6
4.3.2. 卡录像播放.....7
4.4. 云录.....8
4.4.1. 获取云录像列表信息.....8
4.4.2. 播放.....9
4.5. 摄像机信令.....9
4.6. 安防.....10
4.6.1. 开启与关闭.....10
4.6.2. 拉取数据.....10
4.6.3. 接受报警推送.....10
4.7. 登出与销毁.....10
4.7.1. 登出 sdk.....10
4.7.2. 销毁 SDK.....10
5. 附录.....11
5.1. 开发常见问题.....11
5.2. 错误码说明.....11
5.3. 版本修订说明.....11

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

1. 名词解释

IPC：360 摄像头设备端

app_id： 开放平台分配给开发者的业务 id
app_server_key： 开放平台分配给开发者的服务端请求秘钥，和 app_id 进行匹配
app_sdk_key： 开放平台分配给开发者的 SDK 请求秘钥，和 app_id 进行匹配

uid：开发者的应用中的用户 id，开放平台用于标识最终用户
sn：摄像机的设备序列号，可以在摄像机的底座或包装盒的背面找到

usid：uid 的身份认证信息，和 app_id、uid 有关，和 sn 无关。通过调用开放平台服务端 API 获取，SDK 需要。

sn_token：用户和 sn 在开放平台处的标识，用来验证用户对 sn 的权限，和 app_id，uid，sn 都有关系，由开发者生成，SDK 需要；
计算公式：aes 加密，秘钥是 app_server_key，加密内容是 expire+’,’+app_id+’,’+uid+’,’+sn

usid，sn_token，开发者可以缓存。过期后：a)usid 过期，调用开放平台服务端 API 更新数据；b）sn_token，调用开发者服务端 API 更新数据

2. 开发说明

2.1. 关于 sn 号入库

所有要使用的设备，sn 号都必须进行入库，前期的开发测试使用的，可以联系对接人进行入库，当正式运营时，发货后我们会自动入库。

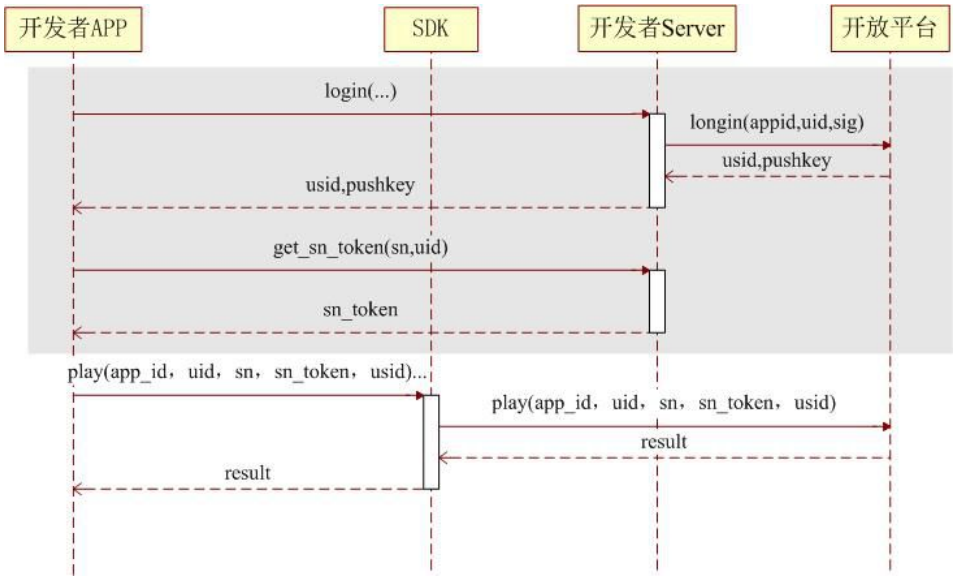
2.2. 关于开放平台和 360 智能摄像机 APP

摄像机在开放平台和 360 智能摄像机 APP 两个系统中是互相冲突的，入库的设备，只能在开放平台下以 SDK 的方式使用，不能在 360 智能摄像机的 app 中使用。没有入库的设备只能在 360 智能摄像机 APP 中使用，不能在开放平台中使用。

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

2.3. 业务逻辑

开放平台不负责维护终端用户和摄像机的权限关系，对开放平台来说所有的摄像机都归属 app_id，并拥有最高权限。
需要开发者 Server 的参与是出于安全的考虑，用户的身份以及对摄像机的权限需要经过开发者 Server 确认后才能使用 SDK 的功能。



1 业务逻辑图

3. 准备工作

从官网上下载 Android SDK 开发包，其中的 demo 工程对 sdk 的功能做了基本的演示，且有一个演示摄像头可供实际操作（注意，演示摄像头可能会有多人同时操作，互相影响），请运行 demo 工程，进行初步了解。
需要在官网上创建应用，审核通过后，在“应用详情”中能够拿到 app_id，sdk_key，sdk 中需要这些信息。

3.1. Manifest 配置

3.1.1. 添加 manifest 子节点,权限

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

```
<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.WAKE_LOCK" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.READ_LOGS" />
```

3.1.2. 添加 application 子节点，app_id、sdk_Key、Service

注意，下面代码示例中的 app_id、sdk_key，请替换为官网上“应用详情”中的相应信息。

```
<!-- **以下是必需配置** -->
<!-- appId 和 appSdkKey 需要接入方在官网上申请，此处为测试数据 -->
<meta-data android:name="appId" android:value="BCSQOMKSQOMKSQOM" />
<!-- 如果 value 是纯数字的话，前面加\0 -->
<meta-data android:name="appSdkKey"
android:value="15ef5f9ae2cd724214355470ef910f52"/>
<!-- 以下这一项是固定的，拷贝就可以了 -->
        <meta-data android:name="DC_APPKEY"
android:value="9766527f2b5d3e95d4a733fcfb77bd7e"/>
<service
android:name="com.qihoo.jiasdk.service.MessageService"
android:exported="false" >
```

```
<intent-filter android:priority="1000" >
</intent-filter>
</service>
```

4. 开始使用 SDK

4.1. 初始化 SDK

SDK 使用前请先执行下面两个动作来初始化 SDK 环境。

```
Qihoo360Camera.init(Context application,QihooCallback qihooCallback);
```

返回值解释详见 javadoc

QihooCallback 为全局回调，包括全局的错误回调，详见 javadoc

```
Qihoo360Camera.loginSDK(UserToken userToken);
```

返回值解释详见 javadoc

UserToken 需要的数据详见 javadoc

4.2. 激活摄像机

4.2.1. 生成携带 wifi 信息的声波文件

```
方法：Qihoo360Camera.createWav(String filePath, String fileName, String ssid,String
psw)
```

方法介绍详见 javadoc

4.2.2. 将 wifi 信息通过声波发送给摄像机

长按摄像机 reset 按钮，直到“滴”声发出，将手机靠近摄像机，需要自行

用 MediaPlayer 等播放器播放上一步生成的 wav 文件，摄像机接收成功会发出“接收成功”的提示音后，如果没有成功，可以重复播放 wav 文件直到成功

4.2.3. 向服务端查询摄像机是否激活上线

当摄像机“接收成功”后会自动连接服务器，app 端可以定时轮询服务器摄像机是否成功激活上线，并设置超时时间，如果超时就提示用户重试

询问摄像机是否激活上线接口：

```
CameraCmdApi.adkIfCameraActivated(String wavId)
```

方法介绍详见 javadoc

4.3. 视频直播

4.3.1. 实时直播

A、创建播放器

```
Qihoo360Camera.createCameraPlayer(Camera camera,PlayerCallback playerCallback)
```

Camera 包含 sn 和 snToken，sn 是摄像机编号，snToken 需要通过调用服务端 360api 获得，详见本文档第三节；PlayerCallback 是播放器的回调，所有的播放状态更新都依赖这个回调，详细介绍详见 javadoc。

B、创建 CameraVideoView，这个类派生自 Android 原生 View，可以在布局文件中配置，也可以直接 new 出来，360 智能摄像机图像宽高比都是 16:9 的，所以 VideoView 最好也设置为 16:9。

C、将 CameraVideoView 绑定给 CameraPlayer

```
mCameraPlayer.setVideoView(CameraVideoView videoView);
```

方法介绍详见 javadoc

D、生命周期相关：

```
@Override
public void onResume() {
```

```
            mVideoView.onResume();
            mCameraPlayer.startPlay();
            super.onResume();
        }
        @Override
        public void onPause() {
            mCameraPlayer.stopPlay();
            mVideoView.onPause();
            super.onPause();
        }
        @Override
        public void onDestroyView() {
            if (mCameraPlayer != null) {
                mCameraPlayer.onDestroy();
            }
            mVideoView.onDestroy();
            super.onDestroyView();
        }
```

E、此后就可以通过 CameraPlayer 实例控制播放了
CameraPlayer 的功能包括：播放、停止播放、静音、切换分辨率、截屏、录像、对讲。详见 CameraPlayer 的 javadoc，使用详见 demo。
PlayerCallback 会回调播放状态的改变：连接中、开始播放、播放失败等，会回调视频时间戳、帧率、码率、录像的时间、对讲时麦克风的音量，详见 PlayStatus 的 javadoc。
F、设置通话全双工/半双工:CameraPlayer.setTalkMode(int mode)；1 是半双工，2 是全双工，详见 javadoc 和 demo

注意：CameraPlayer 只支持同时播放一路视频，只支持同时渲染一个 CameraVideoView，不要尝试同时播放多个。

4.3.2. 卡录像播放

卡录功能其实是直播功能的扩展
A、只要摄像机里插上内存卡，并保证内存卡没有损坏有剩余空间，摄像机就会自动录像在存储卡中，当存储卡满时，会循环删除最旧的视频文件，录制新视频。播放卡录的方法与直播基本相同，只是开始播放时，可以调用 CameraPlayer 的 startPlay(long time)，参数 time 是想要播放的录像时间戳，假如此时间戳没有录像，将播放此时间后最近的录像，假如此后没有录像，则播放直播。
播放过程中可以调用 CameraPlayer 的 setPlayTime(long time)重新定位要播放的录像位置。

B、如何知道摄像机里有哪些时间段的录像呢？
通过 CameraCmdApi 的 getSDRecordData(Camera camera)方法获取，内容介绍详见 SDRecordData 的 javadoc。
然后根据 SDRecordData 中的数据给摄像机 setPlayTime，实现效果可以参考“360 智能摄像机”app。

注：卡录其他功能的使用方法同直播。

4.4. 云录

云录像的视频资源存储在服务器，播放逻辑与直播/卡录有所不同。

云端录像产生逻辑：摄像机检测到画面动作的时候向服务端上传视频。所以有两个条件：1、画面有动作，2、网络正常。

购买：云录服务需要购买，第三方合作的云录需要到 web 上购买，购买业务请咨询 BD ，购买后可以在开放平台官网为设备开通云存。

云录像的 sdk 功能不需要购买，但是只有购买了云录像服务才会生成云录像。

SDK 云录使用方法：

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

4.4.1. 获取云录像列表信息

```
CameraCmdApi.loadCloudRecordList(Camera camera,String startDate, String endDate, int pageCount, int page, int sort)
```

方法介绍详见 javadoc。

4.4.2. 播放

A. 创建云录播放器 CameraRecordPlayer

```
Qihoo360Camera.createCameraRecordPlayer(Camera camera,PlayerCallback playerCallback)
```

参数解释同直播，方法介绍详见 javadoc；

B. 创建 CameraVideoView，方法同直播；

C. 将 CameraVideoView 绑定给 CameraRecordPlayer，同直播。

接下来就可以通过 CameraRecordPlayer 实例进行播放控制了 CameraRecordPlayer 的功能包括播放、停止播放、暂停、继续、截屏（云录像暂不支持录像）、静音，使用方法详见 demo 程序，方法介绍详见 javadoc。

4.5. 摄像机信令

解释：对摄像机发送实时指令，摄像机反馈结果。

此功能依赖手机端和摄像机端的长连，所以过程比一般 http 请求稍长，有一定概率失败。

功能：切换视频清晰度（CameraPlayer 中有集成）、开关指示灯、设置摄像机扬声器音量、开关摄像机上线提示音、开关摄像机离线提示音、开关有人查看摄像机时的提示音、设置摄像机画面倒转、开关摄像机麦克风；

这些功能调用以下接口：

```
CameraCmdApi.setCmd(Camera camera, String key, int value)
```

方法介绍详见 javadoc

获取摄像机存储卡信息（卡录像信息）

```
接口：CameraCmdApi.getSDRecordData(Camera camera)
```

方法介绍详见 javadoc

获取摄像机当前所有设置信息：

```
接口：CameraCmdApi.getCameraSetting(Camera camera)
```

方法介绍详见 javadoc

开关安防功能：

```
接口：CameraCmdApi.setMovingAlarm(Camera camera,int toggle)
CameraCmdApi.setMovingAlarm(Camera camera,int toggle,int mode)
CameraCmdApi.setMovingAlarm(Camera camera,int toggle,float p1x,float p1y,float p2x,float p2y)
CameraCmdApi.setMovingAlarm(Camera camera,int toggle,int beginHour,int beginMinute,int endHour,int endMinute)
```

注：CameraCmdApi 的方法都是同步的耗时操作，需要开子线程调用。

提示：以上功能的实现效果都可以从 360 安全市场下载“360 智能摄像机”app 做参考。

4.6. 安防

解释：360 智能摄像机目前前的安防检测技术有移动侦测、人脸识别、哭声检测，安防产物有图片和视频，目前 SDK 提供的功能只是移动侦测的图片产物。

4.6.1. 开启与关闭

开关移动侦测功能：

```
接口：CameraCmdApi.setMovingAlarm(Camera camera,int moving)
```

方法介绍详见 javadoc

4.6.2. 数据的拉取与删除

1、按天拉取用户（有 1 个或多个摄像机）安防数据索引

```
接口：CameraMasterApi.loadSafeEventListDayIndex(List<Camera> cameras, String endDate, int dayCount)
```

方法介绍详见 javadoc

2、拉取某一台摄像机某一天的安防数据

```
接口：CameraMasterApi.loadSafeEventListOneDay(Camera camera, String date, long timestamp, int count)
```

方法介绍详见 javadoc

3、删除某一台摄像机某一天全天的数据

```
接口：CameraMasterApi.deleteOneDayEvent(Camera camera, String date)
```

方法介绍详见 javadoc

4、逐条删除安防数据

```
接口：CameraMasterApi.deleteEvents(Camera camera, String date, List<Event> events)
```

方法介绍详见 javadoc

5、浏览安防数据

安防数据分为 3 个类型（详见 Event.SubType）

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

IMG：图片类型

VIDEO_CLOUD/VIDEO_MP4:都是视频报警，但是播放方式不一样

VIDEO_CLOUD：先通过 CameraMasterApi.*getCloudRecordInfo* 获取云录播放信息，然后启动播放器播放（详见 demo 中 SecurityDetailActivity）

VIDEO_MP4：先通过 CameraMasterApi.*downloadMotionMp4* 下载小视屏，然后启动播放器播放（详见 demo 中 SecurityDetailActivity）

4.6.3. 接受报警推送 *此功能废弃*

1、接受推送消息
设置消息推送回调：

```
接口：com.qihoo.jiasdk.CameraPushApi.setQihooPushCallback(QihooPushCallback)

回调：com.qihoo.jiasdk.QihooPushCallback.onEventArrived(Event)
```

登陆 SDK 以后就可以通过 onEventArrived 收到推送的回调了。
使用方法详见 demo

2、推送服务的保活：
登陆 SDK 后长连服务自动被启动，但是当手机网络不可用时长连服务会自动停掉，想要重启服务需要开发者自己监听手机网络状态，重启推送服务。
监视长连状态：

```
回调：com.qihoo.jiasdk.QihooPushCallback.onMsgSocketStateChanged(int, String)
```

启动长连服务：

```
接口：com.qihoo.jiasdk.CameraPushApi.startPushClient()
```

4.7. 登出与销毁

4.7.1. 登出 sdk

如果要登出用户或者切换用户，需要登出 SDK，重新登入，否则登出不是必需的。

北京朝阳区酒仙桥路 6 号院 2 号楼 邮编: 100015
电话: +86 10 5682 2690 cameraopen-dev@360.cn

```
Qihoo360Camera.logoutSDK();
```

4.7.2. 销毁 SDK

当不再使用 SDK 或者要退出程序的时候，需要销毁 SDK：

```
Qihoo360Camera.destroy();
```

5. 附录

5.1. 开发常见问题

5.2. 错误码说明

详见 javadoc 中 ErrorCode 的注释

5.3. 版本修订说明

V3.1.0 新增 4.6 安防

V3.2.0

1、增加了安防报警定时的设置、报警区域的设置、报警灵敏度的设置；
2、增加了报警小视屏；
3、支持了全双工
4、更新了播放器，涉及到的文件有：增加 libdistort.so、libffmpeg.so、libjplay_neno.so、更换 libjplayer.so、JPlayer.jar
5、底层网络框架切换到 okhttp，去掉自签名证书