内部资料，请勿外传

趣看公有云平台

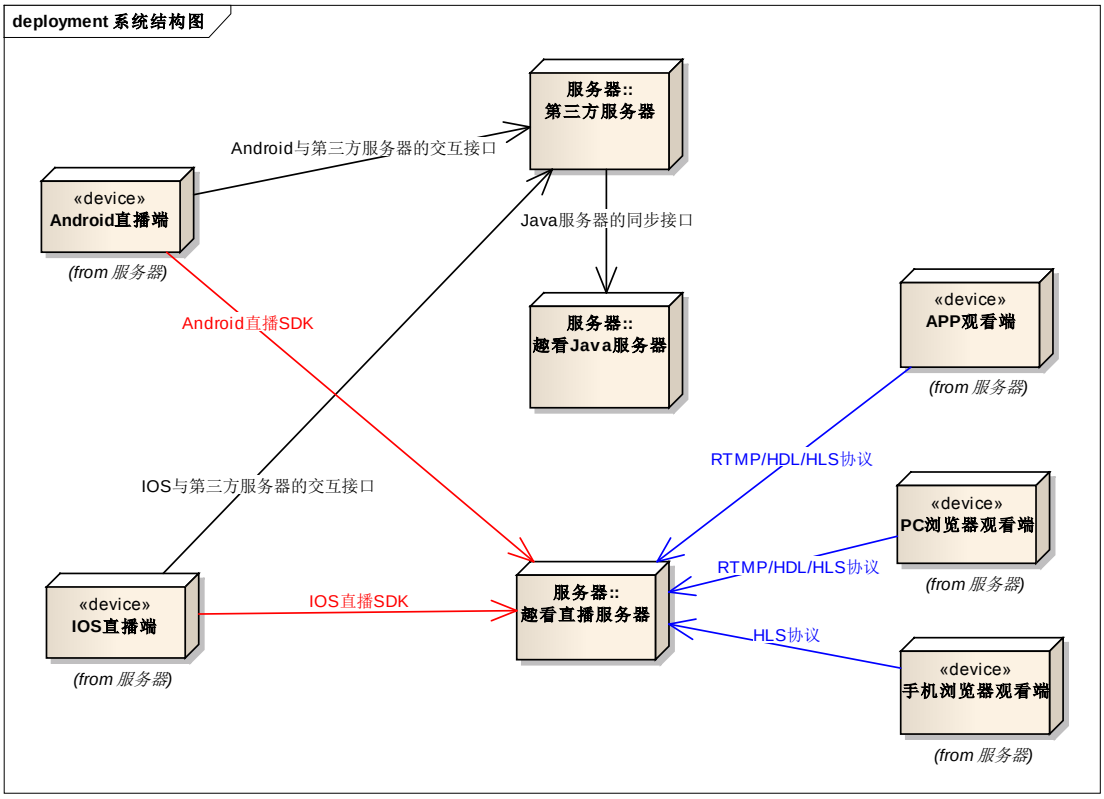
Android 直播 SDK 手册

20151228

目录

一、系统结构图.....2
二、Android 客户端直播接口.....3
2.1 ClientSdk 类的接口.....4
2.1.1 获取当前 SDK 的版本.....4
2.1.2 初始化 sdk 的上下文.....4
2.1.3 添加消息侦听器.....4
2.1.4 删除消息侦听器.....5
2.1.5 设置 appkey.....5
2.1.6 检查直播网络发送队列的状态.....5
2.1.7 网络数据总字节数统计.....5
2.1.8 网络数据发送速度统计.....6
2.2 LiveContext 类的接口.....6
2.2.1 初始化 LiveContext 并申请必须的资源.....6
2.2.2 释放 LiveContext 内申请的资源.....6
2.2.3 初始化摄像机和麦克风.....6
2.2.4 关闭摄像头和麦克风.....7
2.2.5 打开直播开关.....8
2.2.6 关闭直播开关.....8
2.2.7 启动 RTMP 推送.....8
2.2.8 关闭 RTMP 推送.....8
2.2.9 开启或关闭闪光灯.....9
2.2.10 开启或关闭声音.....9
2.2.11 切换前后摄像头.....9
2.2.12 手动聚焦.....9
2.2.13 摄像机画面放大.....10
2.2.14 声音增益.....10
三 Q&A.....10

一、系统结构图



系统接口图

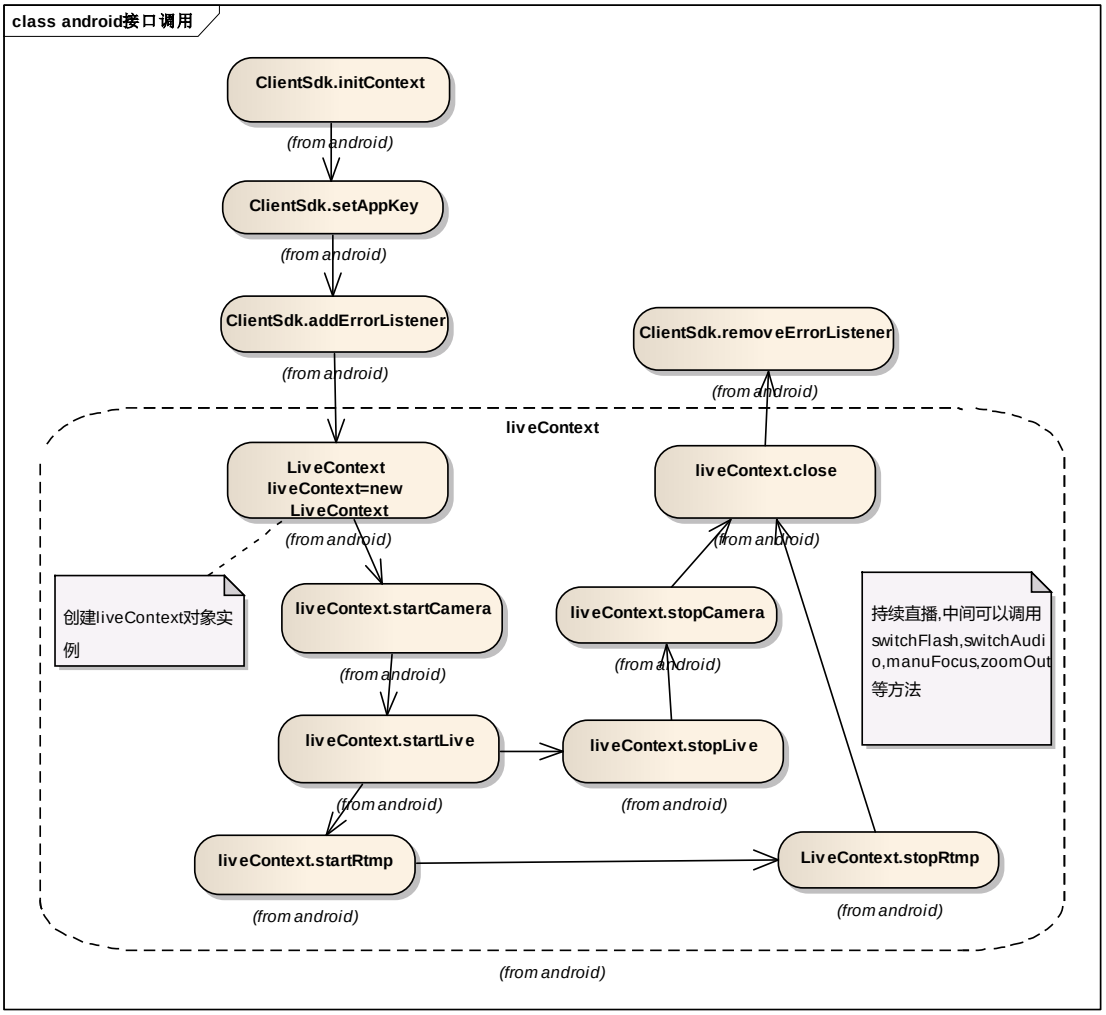
- 在直播端，我司提供 Android 直播 SDK 和 IOS 直播 SDK
- 在观看端，我司提供基于 RTMP、HDL、HLS 等观看协议，并提供 PC 和手机浏览器的内嵌 iframe 页面
- 趣看 Java 服务器提供与第三方服务器之间的 Json 接口，可以完成用户管理以及活动管理等数据同步业务

二、Android 客户端直播接口

提供的 jar 文件: qklive_demo\qksdkdemo\libs 下 fastjson-1.1.27.jar 和 qklivesdk-1.x.x.jar

提供的 so 文件: qklive_demo\qksdkdemo\src\main\jniLibs\armeabi-v7a 下的 libqkffmpeg.so 和 libqkproducer.so

Android 的接口主要分为 2 个类，ClientSdk 是提供服务器登录等基本操作，LiveContext 提供直播相关的接口。



Android 接口函数调用图

2.1 ClientSdk 类的接口

import class com.qukan.clientsdk.ClientSdk; 该类全部是静态方法，可以直接调用。

2.1.1 获取当前 SDK 的版本

类型	描述
函数	public static String getVersion();
参数	
返回值	本 SDK 的当前版本号
说明	

2.1.2 初始化 sdk 的上下文

类型	描述
函数	public static void initContext(Context ctx, int logLevel);
参数	[in]ctx: android.content. Context 对象 [in]logLevel: sdk 的日志等级，即 android.util.Log 的日志等级，DEBUG、INFO、WARN、ERROR 等
返回值	
说明	该函数完成 ClientSdk 的一些全局属性的设置，比如应用程序上下文，日志等级。该函数必须首先被调用，一般在 android 应用程序主 activity 的 onCreate()函数中调用。

2.1.3 添加消息侦听器

类型	描述
函数	public static void addMsgListener(Handler msgHandle);
参数	[in] msgHandle: android.os.Handler 对象
返回值	
说明	当 SDK 内部产生一些消息通知的时候，可以通过注册消息接收句柄来接收这些消息并处理。addMsgListener()方法可以被多个 activity 同时注册，各个 activity 的消息处理函数仅处理自己关心的消息即可。 一般在各个 activity 的 onResume()中调用

消息类型：

```
public class ClientSdk
{
    // 消息通知
    public static final int MSG_INIT_CAMERA_FAILED = 9999001; // 初始化编码器失败
    public static final int MSG_SDCARD_STATUS_ERROR = 9999002; // SD 卡访问失败
```

```
    public static final int MSG_RTMP_CONNECT_FAILED = 9999003; // RTMP 服务器连接断开
    public static final int MSG_APPKEY_CHECK = 9999004;        // appkey 检查
    public static final int MSG_LICENSE_FAILED = 9999005;      // license 检查失败
    public static final int MSG_RTMP_CONNECT_SUCC = 9999006;   // rtmp 服务器连接成功
}
```

2.1.4 删除消息侦听器

类型	描述
函数	public static void removeMsgListener(Handler msgHandle);
参数	[in] msgHandle: android.os.Handler 对象
返回值	
说明	注册了 SDK 消息通知处理函数的 activity 在 onPause()函数中需要调用该函数清除错误侦听器

2.1.5 设置 appkey

类型	描述
函数	public static void setAppKey(String appKey)
参数	[in] appKey: android.content. Context 对象
返回值	
说明	该函数用来设置直播用户的 appkey,appkey 可以在趣看公有云平台查看 在登录页面,会收到 ClientSdk. ***MSG_APPKEY_CHECKE*** 的检查结果

2.1.6 检查直播网络发送队列的状态

类型	描述
函数	public static int getNetPercent();
参数	
返回值	区间[0，5]，0 表示数据完全无法发送出去，5 表示网络数据发送非常通畅
说明	该函数用于检测当前直播时网络数据发送的状态。如果数据发送不流畅，比如返回值大约为 2 或 3，那么视频观看可能会发现略微有些卡顿；如果网络状态为 0 或 1，那么视频可能会非常的卡，或者网络不通。用户可以根据这个值，考虑如何改善现场的网络情况。

2.1.7 网络数据总字节数统计

类型	描述
函数	public static long getTotalDataSendSize();

参数	
返回值	[out]当前直播的网络推送字节数,单位 Byte
说明	获取当前直播推送字节数,直播开始时统计,直播结束时重置为 0

2.1.8 网络数据发送速度统计

类型	描述
函数	public static long getCurrDataSendSpeed(); public static long getCurrDataSendSpeed();
参数	
返回值	[out] 获取当前的发送速度,单位 bps
说明	当前直播的网络数据发送速度, 直播开始时统计,直播结束时重置为 0

2.2 LiveContext 类的接口

import class com.qukan.clientsdk.LiveContext;　　LiveContext 这个类必须要实例化才能使用。

2.2.1 初始化 LiveContext 并申请必须的资源

类型	描述
函数	public LiveContext();
参数	
返回值	
说明	LiveContext 的构造函数。一般在直播的 activity 的 onCreate()方法中调用

2.2.2 释放 LiveContext 内申请的资源

类型	描述
函数	public void close();
参数	
返回值	
说明	该函数在 activity 的 onDestroy()方法中调用，释放 liveContext 申请的各种资源

2.2.3 初始化摄像机和麦克风

类型	描述

函数	public boolean startCamera(SurfaceHolder sh,int videoCameraSize,int videoBitrate,int videoFramerate,int cameraId,int screenOrientation);
参数	[in]sh: 需要显示摄像机预览画面的 surface [in]videoCameraSize: 视频尺寸 public static final int *CAMERA_SIZE_320x240* = 1; public static final int *CAMERA_SIZE_640x480* = 2; public static final int *CAMERA_SIZE_320x180* = 3; public static final int *CAMERA_SIZE_512x288* = 4; public static final int *CAMERA_SIZE_640x360* = 5; public static final int *CAMERA_SIZE_768x432* = 6; public static final int *CAMERA_SIZE_1024x576* = 7; [in]videoBitrate: 视频比特率(也称为视频码流),单位 kbps.一般来说, 低清 320*240 的码流是 200-300 之间; 标清 640*480 的码流是 500-800 之间; 高清 1280*720 的码流是 1200-1500 之间. [in]videoFramerate: 视频的帧率,默认为 10 帧 [in]cameraId: 摄像机 ID, CameraInfo.CAMERA_FACING_BACK 或者 CameraInfo.CAMERA_FACING_FRONT [in]screenOrientation: 屏幕的方向 public static final int *SCREEN_LANDSCAPE* = ActivityInfo.*SCREEN_ORIENTATION_LANDSCAPE*; public static final int *SCREEN_PORTRAIT* = ActivityInfo.*SCREEN_ORIENTATION_PORTRAIT*;
返回值	true: 表示初始化成功 false: 表示初始化失败
错误消息	EventType.INIT_ENCODER_FAILED: 初始化编码器失败，使用硬编码时，某些手机可能不支持，需要切换为软编即可 EventType.SDCARD_STATUS_FAILED: SD 卡存储失败，如果直播时开启了本地录像功能，那么需要检查可用的 sd 卡，如果该 sd 卡无法读取或容量不足，那么会产生该错误消息 EventType.SERVER_OTHER_ERROR: 由直播服务器返回的错误
说明	在 init()函数调用成功后，用户在 surfaceCreate()函数中初始化摄像头和麦克风

2.2.4 关闭摄像头和麦克风

类型	描述
函数	public void stopCamera();
参数	
返回值	
说明	用户结束直播，调用 stopLive()方法之后，调用该方法关闭摄像头和麦克风，将资源归还给 android 系统。注意直播结束后，这个函数必须要调用，否则可能会造成摄像头和麦克风资源被抢占无法归还的情况。

2.2.5 打开直播开关

类型	描述
函数	public void startLive(boolean audioFlag,boolean recordFlag,String recordPath);
参数	[in]audioFlag: 声音开关，true 表示直播开始时有声音，false 表示无声 [in]recordFlag: 录像开关，true 表示直播时本地录像，false 表示无本地录像 [in]recordPath: 录像的指定保存路径
返回值	
说明	该命令在 startCamera 之后调用，它和 stopLive 是可以成对调用的。表示开启直播功能，音视频数据会被编码为 H264 和 AAC 的数据帧

2.2.6 关闭直播开关

类型	描述
函数	public void stopLive();
参数	
返回值	
说明	停止直播。调用该方法后，可以再重新调用 startLive()重新开始将音视频的原始数据编码为 H264 和 AAC。

2.2.7 启动 RTMP 推送

类型	描述
函数	public void startRtmp(String rtmpUrl,int reconnectTimetout);
参数	[in]rtmpUrl: 推送的目标地址,必须是趣看云上的 RTMP 推送地址 [in]reconnectTimeout: 重连 rtmp 服务器的超时时间,单位毫秒
返回值	
说明	如果 rtmpUrl 不合法,handler 会接收到 MSG_LICENE_FAILED 消息 如果 rtmp 服务器无法连接,handler 会收到 MSG_RTMP_CONNECT_FAILED 消息 如果 rtmp 服务器连接成功, handler 会收到 MSG_RTMP_CONNECT_SUCC 消息

2.2.8 关闭 RTMP 推送

类型	描述
函数	public void stopRtmp();
参数	
返回值	
说明	关闭与 rtmp 服务器的连接

2.2.9 开启或关闭闪光灯

类型	描述
函数	public boolean switchFlash(boolean flashFlag);
参数	[in]flashFlag:true 表示开启闪光灯，false 表示关闭闪光灯
返回值	函数调用是否成功，true 表示调用成功，false 表示调用失败
说明	在调用 startCamera()之后，就可以调用该函数开启或关闭闪光灯。调用 startLive()直播的过程中，也可以调用该函数开启或关闭闪光灯。

2.2.10 开启或关闭声音

类型	描述
函数	public boolean switchAudio(boolean audioFlag);
参数	[in]audioFlag:true 表示开启声音，false 表示关闭直播声音
返回值	函数调用是否成功，true 表示调用成功，false 表示调用失败
说明	在调用 startCamera()之后，就可以调用该函数开启或关闭声音。调用 startLive()直播的过程中，也可以调用该函数开启或关闭声音。

2.2.11 切换前后摄像头

没有直接提供对应的接口函数，用户可以通过如下步骤来切换前后摄像头(具体做法参考 demo 的 onSwitchVideo 方法):

- 调用 stopLive 和 stopCamera 来停止当前的直播
- 调用 initSurfaceView 来触发 android.view.SurfaceHolder.surfaceCreate()回调函数，在 surfaceCreate()函数中，再调用 startCamera(other camera id)和 startLive 来达到切换摄像头的目的

2.2.12 手动聚焦

类型	描述
函数	public void manualFocus(float x,float y,int surfaceWidth,int surfaceHeight);
参数	[in] x: 手动聚焦的横坐标，单位像素 [in] y: 手动聚焦的纵坐标，单位像素 [in] surfaceWidth: 整个摄像头预览界面的像素宽度 [in] surfaceHeigth: 整个摄像头预览界面的像素高度
返回值	
说明	手动聚焦的模式，就可以调用该函数使手机的摄像头修改聚焦的焦点位置。 如果需要自动聚焦,可以参看 sample 中提供的使用传感器触发聚焦的代码

2.2.13 摄像机画面放大

类型	描述
函数	public void zoomOut(double zoomRatio);
参数	[in] zoomRatio: 放大缩小的倍数，android camera 的[0,maxZoom]的放大倍数被统一映射为 zoomRatio 的区间[0,1]。zoomRatio=0 表示原有尺寸，不放大；zoomRatio=1 表示放大到 android 相机支持的最大倍数，mCamera.getParameters().getMaxZoom();
返回值	
说明	设置摄像机放大的倍数，前置摄像机不能执行 zoom 操作

2.2.14 声音增益

类型	描述
函数	boolean setAudioGainControl(int ratio) ;
参数	[in] ratio: 声音增益倍数，放大倍数[0,100]; 0 表示不放大(默认情况),100 表示放大一倍
返回值	true: 表示设置成功 false: 表示设置失败
说明	设置麦克风声音放大倍数,因为 android 的内置麦克风普遍存在声音偏小的问题,增加这个函数用来线性放大麦克风的声音.

三 Q&A

1：如果导入 SDK 后在 build.gradle 中设置 targetSdkVersion 为 23,为什么 app 会立刻崩溃

答: SDK 使用到了 x264 的开源库,当前的开源库编译出的 so 文件中,有 text relocation 的问题;在 android level21,22 等版本,对这个问题仅仅是 warning,但是在 android 23 版本就直接报 error,导致 app 崩溃.建议如下设置:

```
ext {
    compileSdkVersion = 23
    targetSdkVersion = 22
    buildToolsVersion = "23.0.2"

    versionCode =  151218
    versionName =  "2.0.0"
}
```

只要 targetSdkVersion 小于 23 即可