

内部资料，请勿外传

趣看公有云平台

Android 播放器 SDK 手册

20151228

目录

一、简介.....	2
二、Android 播放器接口.....	3
2.1 PlaySdkUtils 类的接口.....	3
2.1.1 获取 sdk 版本号.....	3
2.1.2 初始化 sdk.....	3
2.2 IMediaPlayer 接口.....	3
2.2.1 设置显示的 surface.....	3
2.2.2 设置数据源.....	4
2.2.3 异步准备.....	4
2.2.4 开始播放.....	4
2.2.5 停止播放.....	4
2.2.6 暂停播放.....	4
2.2.7 设置播放时屏幕常亮.....	5
2.2.8 获取视频的宽和高.....	5
2.2.9 视频是否正在播放.....	5
2.2.10 拖动视频.....	5
2.2.11 获取当前播放进度条的位置.....	6
2.2.12 获取当前回放视频的总长度.....	6
2.2.13 释放播放器资源.....	6
2.2.14 重置播放器状态.....	6
2.2.15 设置播放器的侦听器.....	7
三 Q&A.....	8

一、简介

QkMediaPlayer 是趣看公司开发的手机播放器,提供与 android 原生播放器类似的接口,移植和修改都会比较容易。

Sdk 文件包括 `sample/libs/qkplayer-1.x.x.jar`

`sample/src/main/jniLibs/libqkffmpeg.so` (与直播 sdk 中的同名文件,内容一样)

`sample/src/main/jniLibs/libqkplayer.so`

`sample/src/main/jniLibs/libqksdl.so`

提供的样例程序是横屏播放,并且有进度控制,如果要竖屏播放,修改 `sample` 里的界面就可以。QkMediaPlayer 支持多种协议的视频播放,比如 `rtmp`, `hdi`, `hls` 等协议,其中 `rtmp`、`hdi` 协议的延时比较小, `hls` 协议的延时比较大。

二、Android 播放器接口

2.1 PlaySdkUtils 类的接口

2.1.1 获取 sdk 版本号

类型	描述
函数	<code>public static String getVersion();</code>
参数	
返回值	版本号字符串，例如“1.1.0.150814”
说明	该函数获取播放器 SDK 的版本号字符串

2.1.2 初始化 sdk

类型	描述
函数	<code>public static void init(Context ctx, int level);</code>
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 Sdk 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。

2.2 IMediaPlayer 接口

2.2.1 设置显示的 surface

类型	描述
函数	<code>void setDisplay(SurfaceHolder sh);</code>
参数	[in]sh: SurfaceHolder 对象
返回值	
说明	设置播放器显示的 surface，sh 在 SurfaceHolder.Callback 接口的 surfaceCreated() 中获取实例

2.2.2 设置数据源

类型	描述
函数	<code>void setDataSource(String path);</code>
参数	[in]path: 网络视频的 url
返回值	
说明	设置播放器的网络 url, 例如 rtmp, hdl, hls 等协议的链接

2.2.3 异步准备

类型	描述
函数	<code>void prepareAsync();</code>
参数	
返回值	
说明	播放器在设置了数据源后, 调用该函数获取部分视频数据, 解析出视频信息。当解析完成后, 回调 OnPreparedListener 接口的 onPrepared() 函数

2.2.4 开始播放

类型	描述
函数	<code>void start();</code>
参数	
返回值	
说明	当 OnPreparedListener 的 onPrepared() 函数被回调后, 如果播放按钮是打开的, 那么开始播放视频

2.2.5 停止播放

类型	描述
函数	<code>void stop();</code>
参数	
返回值	
说明	停止播放视频

2.2.6 暂停播放

类型	描述
----	----

函数	<code>void pause();</code>
参数	
返回值	
说明	暂停播放视频。建议在直播的时候，不要调用该函数；因为 <code>pause</code> 的时候，播放器会在缓冲区中继续缓存部分视频数据，当恢复播放后，先播放该缓存的数据，会造成实时视频出现延迟

2.2.7 设置播放时屏幕常亮

类型	描述
函数	<code>void setScreenOnWhilePlaying(boolean screenOn);</code>
参数	<code>[in]screenOn</code> : 屏幕是否常亮
返回值	
说明	设置播放时屏幕是否常亮，一般都要设置为 <code>true</code>

2.2.8 获取视频的宽和高

类型	描述
函数	<code>int getVideoWidth();</code> <code>int getVideoHeight();</code>
参数	
返回值	视频的宽和高
说明	

2.2.9 视频是否正在播放

类型	描述
函数	<code>boolean isPlaying();</code>
参数	
返回值	视频是否正在播放
说明	该函数一般用于判断是否能按下 <code>pause</code> 按钮的；如果要判断是否是否结束，应该添加其他的 <code>listener</code>

2.2.10 拖动视频

类型	描述
函数	<code>void seekTo(long msec);</code>
参数	<code>[in]msec</code> : 毫秒数
返回值	

说明	拖动视频到指定的位置，视频的总时长通过 <code>getDuration()</code> 获取
----	---

2.2.11 获取当前播放进度条的位置

类型	描述
函数	<code>long getCurrentPosition();</code>
参数	
返回值	当前回放的播放位置，单位毫秒
说明	获取当前播放进度条

2.2.12 获取当前回放视频的总长度

类型	描述
函数	<code>long getDuration();</code>
参数	
返回值	当前回放视频的总长度，单位毫秒
说明	回放视频才有意义

2.2.13 释放播放器资源

类型	描述
函数	<code>void release();</code>
参数	
返回值	
说明	释放播放器资源，具体使用参看样例

2.2.14 重置播放器状态

类型	描述
函数	<code>void reset();</code>
参数	
返回值	
说明	重置播放器的状态，具体使用参看样例

2.2.15 设置播放器的侦听器

设置侦听器的函数

```
// 视频信息解析完成的侦听定时器
public abstract void setOnPreparedListener(OnPreparedListener listener);

// 视频回放完成的侦听器
public abstract void setOnCompletionListener(OnCompletionListener listener);

// 视频缓存进度的侦听器
public abstract void setOnBufferingUpdateListener(OnBufferingUpdateListener listener);

// 拖动进度条执行完成的侦听器
public abstract void setOnSeekCompleteListener(OnSeekCompleteListener listener);

// 视频尺寸播放过程中发生变化的侦听器
public abstract void setOnVideoSizeChangeListener(OnVideoSizeChangeListener listener);

// 视频出错的侦听器，比如服务器无法连接、直播已经结束
public abstract void setOnErrorListener(OnErrorListener listener);

// 提示信息的侦听器
public abstract void setOnInfoListener(OnInfoListener listener);
```

侦听器接口，都是 `IMediaPlayer` 的内部接口类

```
/*-----  
 * Listeners  
 */  
public static interface OnPreparedListener {  
    public void onPrepared(IMediaPlayer mp);  
}  
  
public static interface OnCompletionListener {  
    public void onCompletion(IMediaPlayer mp);  
}  
  
public static interface OnBufferingUpdateListener {  
    public void onBufferingUpdate(IMediaPlayer mp, int percent);  
}  
  
public static interface OnSeekCompleteListener {  
    public void onSeekComplete(IMediaPlayer mp);  
}  
  
public static interface OnVideoSizeChangedListener {  
    public void onVideoSizeChanged(IMediaPlayer mp, int width, int height, int sar_num, int sar_den);  
}  
  
public static interface OnErrorListener {  
    public boolean onError(IMediaPlayer mp, int what, int extra);  
}  
  
public static interface OnInfoListener {  
    public boolean onInfo(IMediaPlayer mp, int what, int extra);  
}
```

如何使用，请参看 `com.kukan.playsdk.sample.VideoActivity`

三 Q&A

1: 播放器开始播放视频前有一段时间黑屏,什么时候知道画面开始出现?

答: 可以参考 `Sample` 中的 `QkVideoView` 的 `mInfoListener` 的消息处理。

当接收到 `IMediaPlayer.MEDIA_INFO_VIDEO_RENDERING_START` 消息的时候, 说明播放器已经准备好解码好视频数据, 可以送显卡渲染了

2: 因为网络不正常, 导致播放器视频卡顿, 有没有对用户更友好的提示?

答: 可以参考 `Sample` 中的 `QkVideoView` 的 `mInfoListener` 的消息处理。

1) 当接收到 `IMediaPlayer.MEDIA_INFO_BUFFERING_START` 消息的时候, 说明播放器的队列中已经没有数据, 需要从本地或网络加载数据

2) 当接收到 `IMediaPlayer.MEDIA_INFO_BUFFERING_QUEUE_INFO` 的时候, 消息中 `arg2` 参数中有当前队列缓存的数据百分比

3) 当接收到 IMediaPlayer. *MEDIA_INFO_BUFFERING_END* 消息的时候, 说明播放器的队列缓存数据已经足够, 可以开始播放了