

全民 K 歌 TV 版硬件音频组件接口文档

(ver 0.5.0)

目录

全民 K 歌 TV 版硬件音频组件接口文档.....	1
1. 文档介绍.....	3
1.1 文档目的.....	3
1.2 术语与缩写解释.....	3
1.3 更新历史.....	3
2. 总体说明.....	6
3. 接入说明.....	7
3.1 代码模块接入说明.....	7
3.2 接入形式.....	7
4. 各模块接口说明.....	8
4.1 音频输出模块.....	8
4.1.1 AudioOutputInstaller.....	9
4.1.2 AudioOutput.....	11
4.2 音频接收模块.....	13
4.2.1 AudioReceiverInstaller.....	14
4.2.2 AudioReceiver.....	18
4.3 其他一些辅助功能的模块.....	21
4.4 其他封装类.....	22
4.4.1 AudioParams	22
4.4.2 AudioFrame.....	23
4.4.3 AudioPlayState	23
4.4.4 AudioEnv	24
4.4.5 DebugPrinter	24
4.4.6 ReceiverChannelType	25
4.4.7 AdaptDeviceType	25
4.4.8 DeviceInfo.....	26
4.4.9 ErrorCode.....	27
5. 示例.....	27

1. 文档介绍

1.1 文档目的

本文档是描述音频组件提供方接入全民 K 歌 TV 版时的使用说明文档

1.2 术语与缩写解释

缩写、术语	解释
K 歌	全文中提到的 K 歌均指全民 K 歌 TV 版本
麦克风	只是用户用来输入声音的设备，包括但不限于话筒麦克风，遥控器麦克风，手机麦克风

1.3 更新历史

版本	修订日期	作者	描述
0.5.0	2020-2-26	zoroweili(李伟)	<ol style="list-style-type: none"> 1. 新增获取当前接收器类型的接口。 详见 <code>AudioReceiverInstaller.getReceiverChannelType</code> 2. 新增获取接收器连接状态的接口。 详见 <code>AudioReceiverInstaller.getCanCheckReceiverDevice/</code> <code>isReceiverDeviceConnected/</code> <code>notifyReceiverDeviceConnectChanged/</code> <code>getCanCheckMic/ isMicOpen/</code> <code>notifyMicOpenChanged</code> 等方法

			<p>3. 新增错误信息反馈的接口。 详见 AudioReceiverInstaller. notifyErrorInfo</p> <p>4. 新增获取硬件信息的接口。 详见 AudioReceiverInstaller. getDeviceInfo</p> <p>5. 新增当前设备已适配可用的外设推荐。 AudioReceiverInstaller. getAdaptDeviceTypeList</p> <p>6. 在回调的基础上新增获取已经准备好手机 server 的端口号的方法 getPreparedPhoneSocketPorts</p>
0.4.0	2019-5-9	Daviddwchen(陈定伟)	<p>1. 新增回调接口</p> <p>2. 新增调试接口</p> <p>3. 新增全局 Context 获取接口</p>
0.3.1 beta	2019-5-8	zoroweili(李伟)	<p>1. 增加 3.2 处接入形式的说</p> <p>2. 对 AudioReceiver 增加一些注意事项的说明</p>
0.3 beta	2019-4-16	zoroweili(李伟) zoroweili@tencent.com	<p>AudioFrame 增加 byte[] byteBuffer 属性, 传入人声同时支持 short 数组类型和 byte 数组类型; isByteData() 接口判断传入的数据是 byte 格式还是 short 格式; size 参数指定数据长度, 不需要做 short byte 格式长度转换</p>
0.2 beta	2019-2-25	zoroweili(李伟) zoroweili@tencent.com	<p>将 AudioFrame 中的 buffer 类型修改为 short[]</p> <p>给 AudioOutput 增加了 getPlayState 方法</p> <p>修改了 AudioOutput 中 notifyPlayStateChanged 方法的描述</p> <p>给 AudioOutput 的 write 方法增加了 int 型的返回值</p>
0.1.1 beta	2019-2-14	zoroweili(李伟) zoroweili@tencent.com	<p>删除了 AudioFrame 中一处多余的变量</p>

0.1 beta	2018-12-28	zoroweili(李伟) zoroweili@tencent.com	初版接口设计，主要包括音频输出和音频接收的模块接口设计，其他功能的模块已计划待补充
----------	------------	---	---

2. 总体说明

随着智能 TV（OTT 盒子）的快速普及，利用家庭 TV 进行 K 歌的需求渐渐浮出水面，而 TV 目前主流操作系统为 Android 系统，针对外设音频设备的语音的录入播放若采用 Android 系统原生的播放方式，延迟很大，基本达不到正常 K 歌的体验效果，这时需要一些硬件音频组件提供方帮助来实现底层的低延时声音传输方案。

为了便于各硬件音频组件提供方接入 K 歌，特此制定一套统一的接口，按照这个接口标准来实现底层功能。

3. 接入说明

3.1 代码模块接入说明

音频接口主要包括两大模块：音频输出和音频输入

音频输出，是指 K 歌把歌曲的声音数据传给音频组件底层来播放。如果底层要播放 K 歌的声音数据，则必须实现该模块的接口。

音频输入，是指音频组件接收麦克风的人声数据传给 K 歌。如果底层要给 K 歌提供人声数据，则必须实现该模块的接口。

此外，还有一些辅助的功能模块。

3.2 接入形式

K 歌侧会以 aar 包的形式提供接口 sdk。

音频组建提供方，需要接入该 aar，并按照后续说明实现其中的接口，并打成 aar 包提供给 K 歌侧。

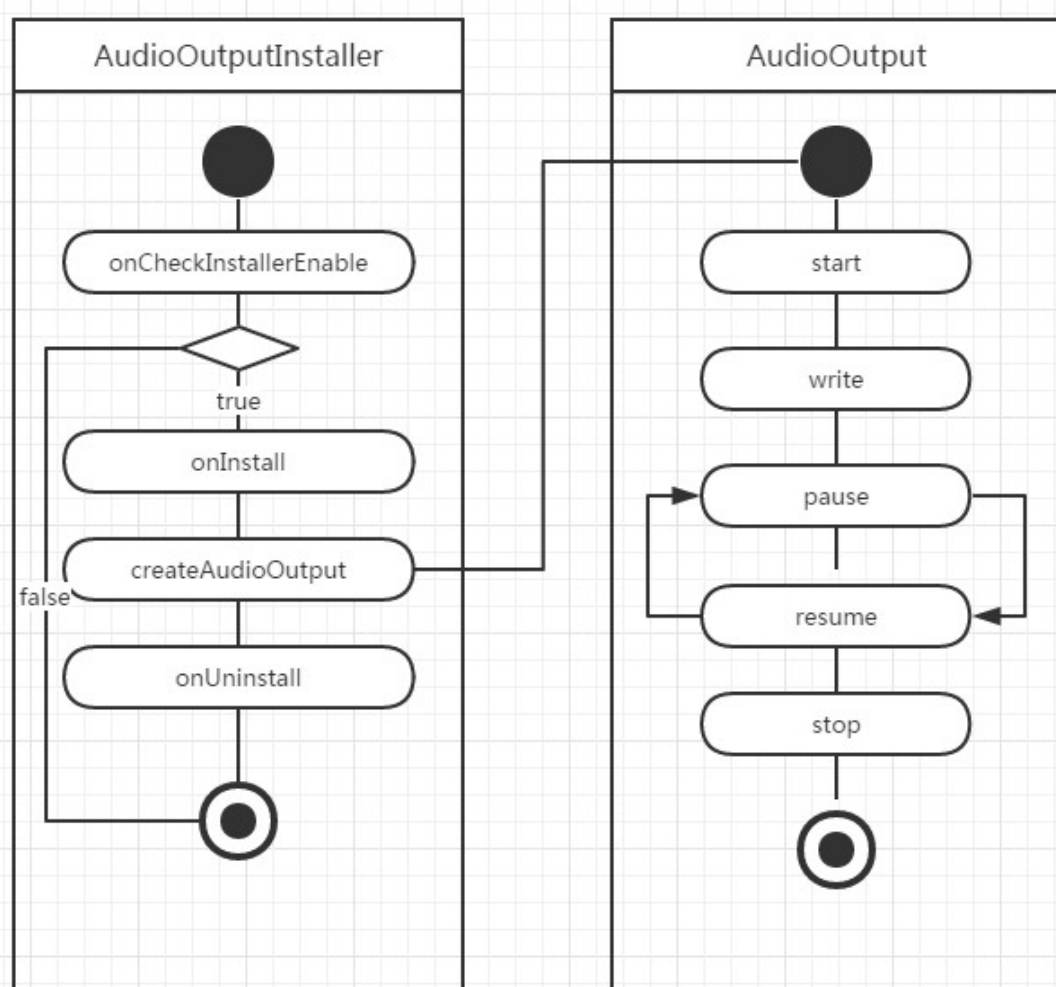
需要注意几点：

- 1) 请不要混淆 K 歌的接口类
- 2) 最终打的 aar 包里面，请不要把 K 歌的接口包打进去
- 3) 不想被 K 歌侧混淆的类，请实现 NoProguard 接口
- 4) Jdk 版本 1.6+

4. 各模块接口说明

4.1 音频输出模块

实现音频输出，需要实现 `AudioOutputInstaller` 和 `AudioOutput` 两个接口，这两个接口的调用逻辑如下：



以下是各接口的主要方法说明：

4.1.1 AudioOutputInstaller

音频输出组件安装器

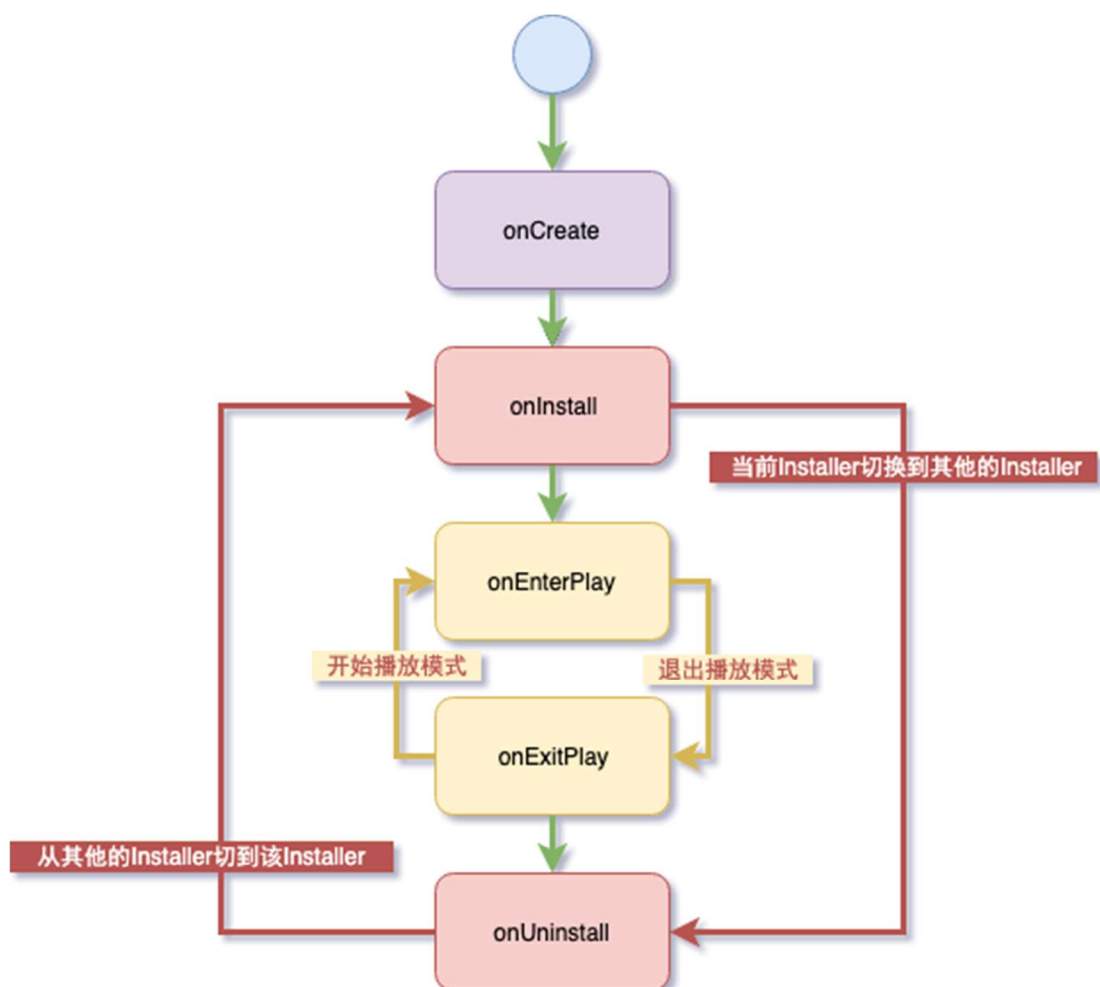
K 歌通过此安装器来获取 AudioOutput，并且做一些底层环境搭建工作

public 方法	
void	<div>getVersion()</div> <div>获取 SDK 版本</div>

protected 方法	
boolean	<div>onCheckInstallerEnable()</div> <div>是否可使用该安装器</div> <div>返回值： boolean : true 可以使用该安装器构建；false 不可以使用</div>
void	<div>onCreate()</div> <div>Installer 构建时候回调，每次进程启动只会回调一次</div>
void	<div>onInstall()</div> <div>当采用该安装器的时候回调,每次装载只会回调一次</div>
void	<div>onUninstall()</div> <div>该安装器被其他的安装器所替代，需要卸载该安装器的时候调用</div>
void	<div>onEnterPlay()</div> <div>当 K 歌进入播放模式的时候回调该函数，该函数调用发生在</div>

	onInstall 之后
void	onExitPlay() 当 K 歌退出播放模式之后回调该函数,该函数调用发生在 onUnInstall 回调之前
AudioOutput	onCreateAudioOutput(AudioParams params) 创建 AudioOutput 入参: AudioParams params : 音频参数 返回值: AudioOutput : 音频输出器

AudioOutputInstaller 回调接口关系图:



4.1.2 AudioOutput

音频输出器

完成音频数据传输到底层输出组件进行播放的功能

public 方法	
void	<p>start()</p> <p>打开音频输出，之后就可以通过 write 方法播放音频数据了</p>
void	<p>stop()</p> <p>关闭音频输出</p>
void	<p>resume()</p> <p>如果之前调了暂停，则调用此方法继续音频输出</p>
void	<p>pause()</p> <p>暂停音频输出</p>
void	<p>flush()</p> <p>冲掉剩余的播放队列中的数据</p>
void	<p>release()</p> <p>释放资源</p>
int	<p>write(AudioFrame frame)</p> <p>写音频数据，用于播放</p> <p>入参：</p> <p>AudioFrame frame ： 一段音频数据，详见 AudioFrame</p>

	<p>返回值:</p> <p>int 成功写入的字节数</p>
void	<p>setVolume(float volume)</p> <p>设置输出音频数据的音量, 0.0f 到 1.0f</p> <p>这是在设备本身的音量基础之上设置的音量增益, 并不是最终的实际音量</p> <p>入参:</p> <p>float volume : 音量值, 0.0f 到 1.0f</p>
void	<p>setMicVolume(float volume)</p> <p>设置麦克风的音量, 0.0f 到 1.0f</p> <p>这是在设备本身的音量基础之上设置的音量增益, 并不是最终的实际音量</p> <p>入参:</p> <p>float volume : 音量值, 0.0f 到 1.0f</p>
int	<p>getPlaybackHeadPosition()</p> <p>获取当前播放的 PCM 的 sample 位置, 即当前播放的采样点的位置</p> <p>返回值:</p> <p>int : 当前播放的 PCM 的 sample 位置, 即采样点的位置</p>
int	<p>getAudioSessionId()</p> <p>返回音频 session ID</p>

	<p>返回值:</p> <p>int : 音频 session ID</p>
int	<p>getPlayState()</p> <p>获取播放状态</p> <p>返回值:</p> <p>int : 播放状态, 详见 AudioPlayState</p>

这些方法是由 K 歌来调用的, 无需另起线程来实现, 由 K 歌来同步调用

protected 方法	
void	<p>notifyPlayStateChanged(int state)</p> <p>通知播放状态的改变</p> <p>入参:</p> <p>int : 播放状态, 详见 AudioPlayState</p> <p>注: 这个方法父类已经实现, 无需子类实现, 但是需要子类在合适的地方调用</p>

4.2 音频接收模块

音频接收, 用于接收通过麦克风输入的音频数据。

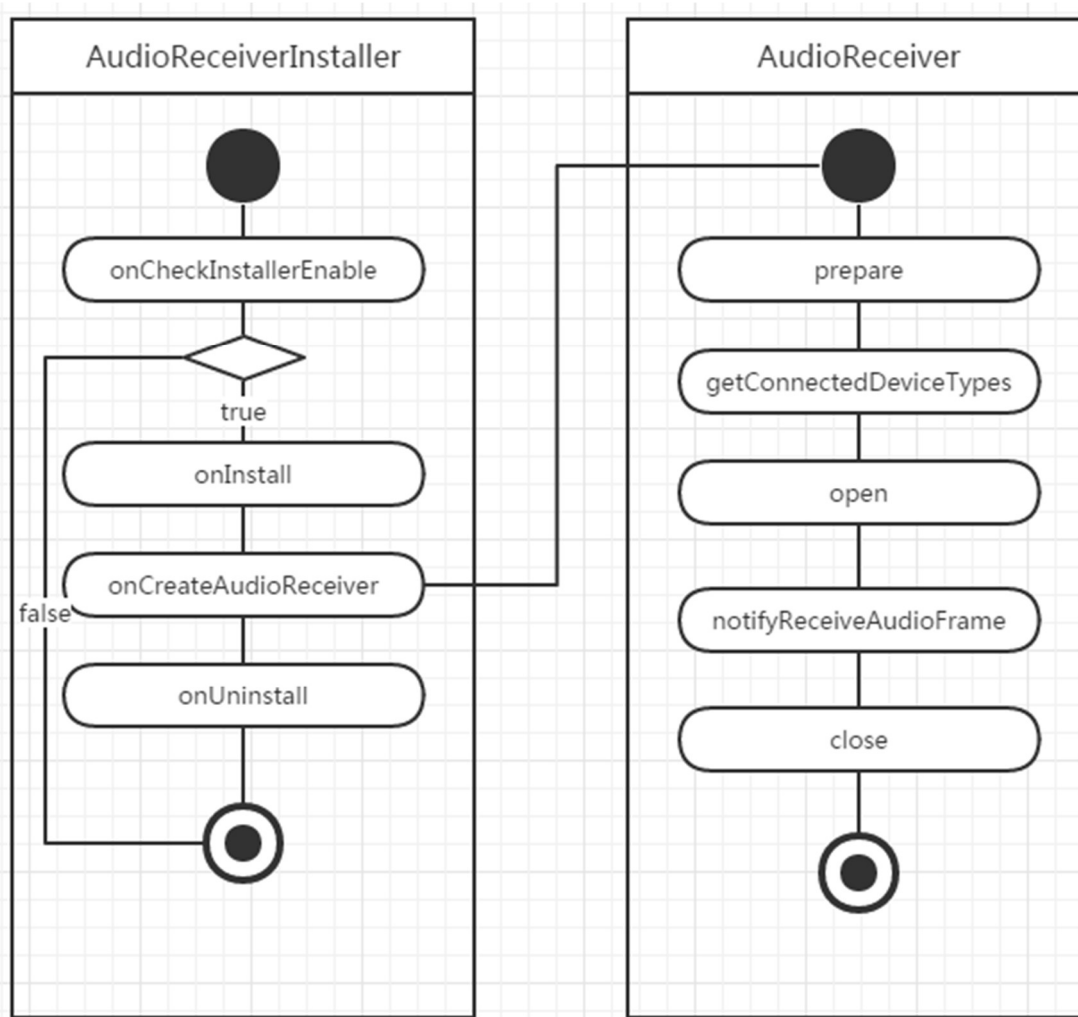
主要场景包括但不限于:

- 1) 用户通过麦克风唱歌, 这里要接收用户的人声数据, 用于之后的处理

- 2) 用于通过麦克风进行语音搜索，这里需要获取到人声数据，用于之后的语音识别

如果需要提供这个功能，请实现该模块的相关接口，K 歌会调用这些接口来进行人音频接收；如果不提供该功能，则无需实现。

接口的调用逻辑如下：



4.2.1 AudioReceiverInstaller

音频接收装载器

用于麦克风收录声音时的回调

protected 方法	
boolean	<p>onCheckInstallerEnable()</p> <p>是否可使用该安装器</p> <p>返回值:</p> <p>boolean : true 可以使用该安装器构建; false 不可以使用</p>
void	<p>onCreate()</p> <p>当 K 歌进入播放模式的时候回调,发生在 onInstall 之后</p>
void	<p>onInstall()</p> <p>当采用该安装器的时候回调,每次装载只会回调一次</p>
void	<p>onUninstall()</p> <p>该安装器被其他的安装器所替代, 需要卸载该安装器的时候调用</p>
void	<p>onEnterPlay()</p> <p>当 K 歌进入播放模式的时候回调,发生在 onInstall 之后</p>
void	<p>onExitPlay()</p> <p>当 K 歌进入播放模式的时候回调,发生在 onUnInstall 之前</p>
AudioReceiver	<p>onCreateAudioReceiver(AudioParams params)</p> <p>创建声音接收器</p> <p>入参:</p> <p>AudioParams params : 音频参数</p> <p>返回值:</p>

	AudioReceiver : 音频接收器
int	<p>getReceiverChannelType()</p> <p>获取接收声音通道类型</p> <p>返回值:</p> <p>Int: 见 4.4.6 ReceiverChannelType</p>
List<Integer>	<p>getAdaptDeviceTypeList()</p> <p>获取当前适配的可用外设类型</p> <p>返回值:</p> <p>List<Integer>: 见 4.4.7 AdaptDeviceType</p>
boolean	<p>getCanCheckReciverDevice()</p> <p>是否支持获取当前外设接收器的状态</p> <p>用于判断 isReceiverDeviceConnected 和</p> <p>notifyReceiverDeviceConnectChanged 是否有效</p> <p>返回值:</p> <p>true: 可以, false: 不可以</p>
boolean	<p>isReceiverDeviceConnected()</p> <p>外设接收器是否已连接 (如果内置接收器, 则可认为已连接)</p> <p>返回值:</p> <p>true: 已连接 (插入), false: 未连接 (拔出)</p>
boolean	<p>getCanCheckMic()</p> <p>是否支持获取麦克风的状况</p> <p>用于判断 isMicOpen 和 notifyMicOpenChanged 是否有效</p>

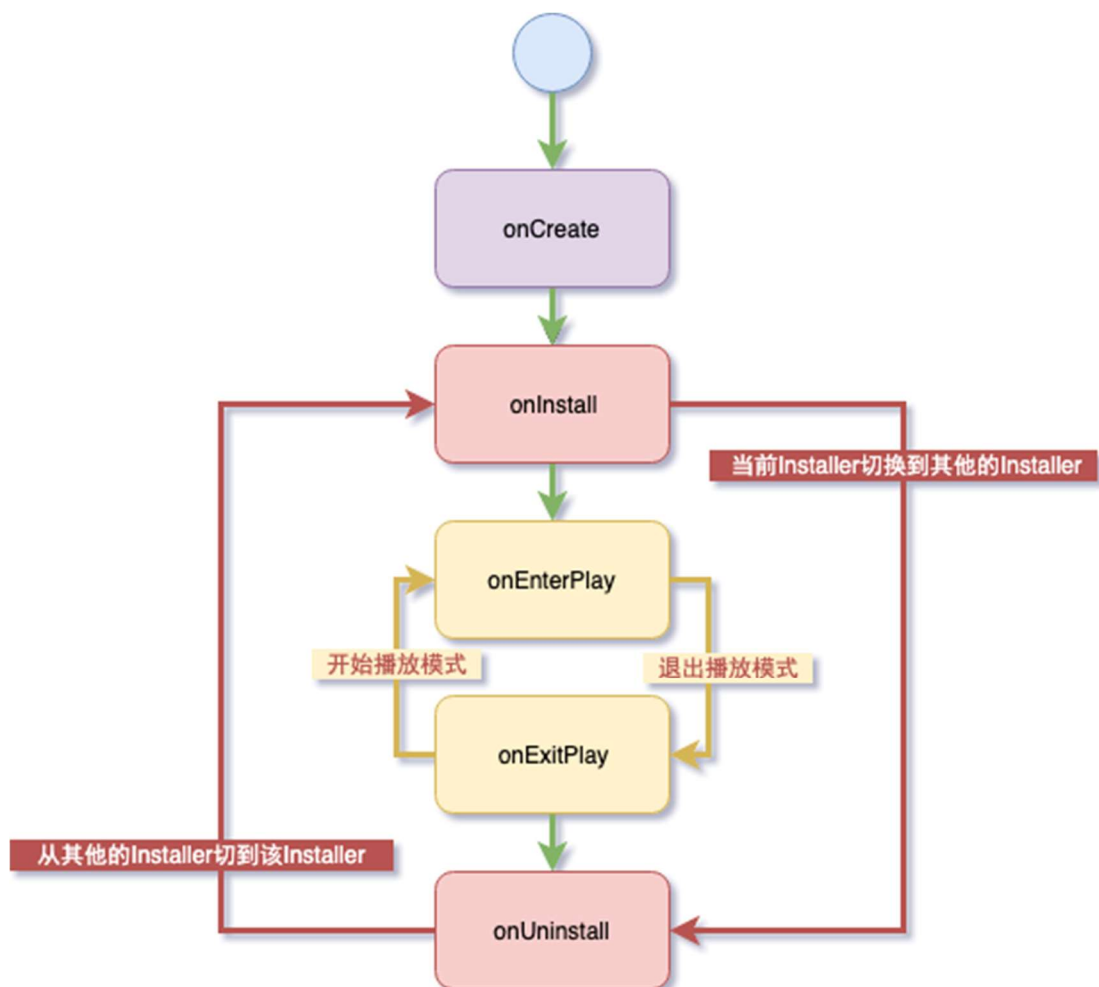
	<p>返回值:</p> <p>true: 可以, false: 不可以</p>
boolean	<p>isMicOpen()</p> <p>麦克风是否打开</p> <p>返回值:</p> <p>true: 打开, false: 关闭</p>
DeviceInfo	<p>getDeviceInfo()</p> <p>获取当前硬件设备信息</p> <p>返回值:</p> <p>详见: 4.4.8 DeviceInfo</p>

Protected 方法用于声音通道实现方在合适的时机去调用

protected 方法	
Void	<p>notifyReceiverDeviceConnectChanged(boolean)</p> <p>外设接收器连接状态改变的时候回调</p> <p>入参:</p> <p>true: 连接 (插入), false: 未连接 (拔出)</p>
Void	<p>notifyMicOpenChanged(boolean)</p> <p>麦克风状态改变的时候回调</p> <p>入参:</p> <p>true: 打开, false: 关闭</p>
void	<p>notifyErrorInfo(int errorCode, int errorMsg)</p> <p>发生错误的时候回调</p>

	<p>errorCode 详见: 4.4.9 ErrorCode</p> <p>errorMsg, 附带错误详情和想给用户提示的信息</p>
--	--

AudioReceiverInstaller 回调接口关系图:



4.2.2 AudioReceiver

音频接收器

常量	
int	DEIVCE_TYPE_UNKOWN

	麦克风设备类型未知
int	DEIVCE_TYPE_MIC 麦克风设备类型:话筒麦克风
int	DEIVCE_TYPE_CONTROL 麦克风设备类型:遥控器麦克风
int	DEIVCE_TYPE_PHONE 麦克风设备类型:手机麦克风

public 方法	
void	prepare() 准备接收音频，开始接收音频之前会调用此方法
int	getConnectedDeviceTypes() 获取当前已经连接上的麦克风外设类型有哪些 返回值： int ： 设备类型的按位或的结果，比如，如果话筒麦克风是 1，手机麦克风是 4，这两种同时连接，那么则返回 $1 4 = 5$ ，详见常量部分
void	open(int deviceType) 打开声音接收 入参： int deviceType ： 麦克风外设的类型，详见常量部分
void	close() 关闭声音输入

void	<p>setReceiveAudioToOutput(boolean flag)</p> <p>设置麦克风收到的声音数据是否需要通过声音输出设备播放出来。比如，如果是在 K 歌过程中，人声数据是需要播放出来的，此时应该设置为 true；如果是在语音搜索，则，应该设置为 false</p> <p>入参：</p> <p>boolean flag ： true or false</p>
Int[]	<p>getPreparedPhoneSocketPorts()</p> <p>获取可用的手机连接 socket 的端口</p> <p>返回值：</p> <p>Int[] 端口号数组</p>

protected 方法	
void	<p>notifyReceiveAudioFrame(AudioFrame frame, int playbackPosition, float volume)</p> <p>通知接收到麦克风收录的声音数据。这个方法是由 SDK 底层调用的，用于把声音数据传给 K 歌。</p> <p>入参：</p> <p>AudioFrame frame：音频数据帧</p> <p>int playbackPosition ： 该数据帧的第一个 sample 的数据对应 AudioOutput 中音频数据的采样点位置，用于伴奏和人声对齐之用</p> <p>float volume ： 声音音量大小， 0.0f~ 1.0f，主要用于给用户展示一个声音输入的波动效果</p>

void	<p>notifyPhoneSocketPortsPrepared(int[] ports)</p> <p>通知用于手机连接的 socket 连接端口已经准备好。这个方法是由 SDK 底层调用的，通知 K 歌连接端口的状态。</p> <p>入参：</p> <p>int[] ports ： 端口号数组，最多支持四路端口。</p>
------	--

注意，notifyReceiveAudioFrame 接口是希望能把人声数据及时的回传给 K 歌，用于打分和保存，所以希望 playbackPosition 能尽可能的接近设备收到人声时对应的伴奏播放时间的采样点位置，偏差越大，打分就会越不准，保存的作品也会有更大的偏差。

另外希望 playbackPosition 可以比较真实的反应这段人声数据的位置，并且稳定，最好保持 playbackPosition 的差值基本一致，否则说明获取的 playbackPosition 不是很准确，请检查相关逻辑。

4.3 其他一些辅助功能的模块

除了音频输出和输入之外的一些功能，包括但不限于：

- 1) 升级，可以包括主动帮助底层检测升级，或者底层升级的时候 K 歌帮忙展示其状态
- 2) 底层组件报错, K 歌可以帮其弹窗提示
- 3) 错误码, 底层统一的一套错误码
- 4) 设备信息获取，比如厂商和型号
- 5) 建立手机与 TV 连接的通路，比如 socket 通道，可以让手机实时把人声数据传到 TV 端播放出来

- 6) dlina 功能，用于全民 K 歌手机版把一些视频投放到 TV 端，支持拉起 K 歌来播放

这些辅助功能先列了一些功能点，详细接口之后逐步完善。

4.4 其他封装类

4.4.1 AudioParams

音频参数

常量	
int	BIT_DEPTH_8BIT 采样深度为 8bit，即 1 字节
int	BIT_DEPTH_16BIT 采样深度为 16bit，即 2 字节

成员变量	
long	sampleRate 采样频率, Hz
int	channelCount 声道数
int	bitDepth 采样深度

4.4.2 AudioFrame

音频数据帧的封装

成员变量	
short[]	buffer 音频数据
byte[]	byteBuffer 音频数据
int	size 其中的音频数组的大小，不需要做 short byte 格式长度转换，不可超过 buffer 或者 byteBuffer 的 length

传入人声同时支持 short 数组类型和 byte 数组类型

方法	
isByteData()	接口判断传入的数据是 byte 格式还是 short 格式

4.4.3 AudioPlayState

AudioOutput 的播放状态

常量	
int	PLAYSTATE_NEW 停止状态
int	PLAYSTATE_STOPPED

	停止状态
int	PLAYSTATE_PAUSED 暂停状态
int	PLAYSTATE_PLAYING 播放状态

4.4.4 AudioEnv

public 方法	
void	getApplicationContext() 获取当前应用上下文
AudioEnv	get() 静态 获取 AudioEnv 单例
void	setDebugPrinter(DebugPrinter) 实现 DebugPrinter 接口,可以用于监控 Audio 的管理流程,Audio 管理流程过程中会输出日志到这个接口中

4.4.5 DebugPrinter

public 方法	
void	println(String) Audio 管理流程中正常日志输出
Void	printThrowable(Throwable) Audio 管理流程中发生异常时候输出

4.4.6 ReceiverChannelType

接收声音通道的适配类型

常量	
int	TYPE_NONE 未适配
int	TYPE_AUDIO_RECORD Android 原生：AudioRecord，如全局麦克风
int	TYPE_MIDDLEWARE 中间件

4.4.7 AdaptDeviceType

已适配可支持的外设类型

常量	
int	TYPE_NONE 未适配
int	TYPE_MIC 麦克风
int	TYPE_KTV_BOX 点唱机
Int	TYPE_LOUDSPEAKER_BOX

	音箱
--	----

4.4.8 DeviceInfo

当前适配的硬件设备信息

常量	
int	TYPE_UNKNOWN 未知
int	TYPE_TV 电视
int	TYPE_BOX 电视盒子
int	TYPE_KTV_BOX 点唱机
Int	TYPE_MIC 话筒
变量	
int	mDeviceType 设备类型
String	mSerialNumber 设备 id

4.4.9 ErrorCode

常量	
int	CODE_METHOD_INNER_ERROR 调用方法内部出错
int	CODE_NOT_INIT Installer 未初始化
int	CODE_STATUS_ERROR 错误的状态下不能执行此方法
int	CODE_LENGTH_ERROR 参数错误，长度参数不满足要求
int	CODE_NOT_SUPPORTED_DONGLE 不支持的 dongle
int	CODE_NOT_SUPPORTED_INTERFACE 不支持的接口
int	CODE_MIDDLEWARE_KILLED 中间件被系统 kill 掉
int	CODE_UN_IMPLEMENT_INTERFACE 接口存在，但底层功能未实现
int	CODE_NETWORK_CHANGED 网络切换导致的连接断开
int	CODE_NOT_SUPPORT_AUDIORECORD 不支持 Android 原生的 Record

5. 示例

比如巴金提供了音频输出的功能，那么需要实现的类的关系如下图所示

