

直播 SDK

客户端 API

产品文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

文档目录

客户端 API

iOS&Mac

API 概览

V2TXLivePusher

V2TXLivePusherObserver

V2TXLivePlayer

V2TXLivePlayerObserver

V2TXLivePremier

TXAudioEffectManager

TXBeautyManager

TXDeviceManager

错误码表

类型定义

Android

API 概览

V2TXLivePusher

V2TXLivePusherObserver

V2TXLivePlayer

V2TXLivePlayerObserver

V2TXLivePremier

TXAudioEffectManager

TXBeautyManager

TXDeviceManager

错误码表

类型定义

属性定义

客户端 API

iOS&Mac

API 概览

最近更新时间：2024-06-06 15:47:57

API OVERVIEW

V2 推流器相关接口

函数列表	描述
setObserver:	设置推流器回调
setRenderView:	设置本地摄像头预览 View
setRenderMirror:	设置本地摄像头预览镜像
setEncoderMirror:	设置视频编码镜像
setRenderRotation:	设置本地摄像头预览画面的旋转角度
setRenderFillMode:	设置本地摄像头预览画面的填充模式
startCamera:	打开本地摄像头
stopCamera	关闭本地摄像头
startMicrophone	打开麦克风
stopMicrophone	关闭麦克风
startVirtualCamera:	开启图片推流
stopVirtualCamera	关闭图片推流
startScreenCapture:	开始全系统的屏幕分享（该接口支持 iOS 11.0 及以上的 iPhone 和 iPad）
stopScreenCapture	关闭屏幕采集
pauseAudio	静音本地音频

resumeAudio	取消静音本地音频
pauseVideo	暂停推流器的视频流
resumeVideo	恢复推流器的视频流
startPush:	开始音视频数据推流
stopPush	停止推送音视频数据
isPushing	当前推流器是否正在推流中
setAudioQuality:	设置推流音频质量
setVideoQuality:	设置推流视频编码参数
getAudioEffectManager	获取音效管理对象
getBeautyManager	获取美颜管理对象
getDeviceManager	获取设备管理对象
snapshot	截取推流过程中的本地画面
setWatermark:x:y:scale:	设置推流器水印。默认情况下，水印不开启
enableVolumeEvaluation:	启用采集音量大小提示
enableCustomVideoProcess:pixelFormat:bufferType:	开启/关闭自定义视频处理
enableCustomVideoCapture:	开启/关闭自定义视频采集
enableCustomAudioCapture:	开启/关闭自定义音频采集
sendCustomVideoFrame:	在自定义视频采集模式下，将采集的视频数据发送到SDK
sendCustomAudioFrame:	在自定义音频采集模式下，将采集的音频数据发送到SDK
enableAudioProcessObserver:format:	开启/关闭对经过前处理后的本地音频帧的监听回调
sendSeiMessage:data:	发送 SEI 消息
showDebugView:	显示仪表盘
setProperty:value:	调用 V2TXLivePusher 的高级 API 接口。
setMixTranscodingConfig:	设置云端的混流转码参数

startLocalRecording:	开始录制音视频流
stopLocalRecording	停止录制音视频流

直播推流器事件回调

函数列表	描述
onError:message:extraInfo:	直播推流器错误通知，推流器出现错误时，会回调该通知
onWarning:message:extraInfo:	直播推流器警告通知
onCaptureFirstAudioFrame	首帧音频采集完成的回调通知
onCaptureFirstVideoFrame	首帧视频采集完成的回调通知
onMicrophoneVolumeUpdate:	麦克风采集音量值回调
onPushStatusUpdate:message:extraInfo:	推流器连接状态回调通知
onStatisticsUpdate:	直播推流器统计数据回调
onSnapshotComplete:	截图回调
onProcessAudioFrame:	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onProcessVideoFrame:dstFrame:	自定义视频处理回调
onGLContextDestroyed	SDK 内部的 OpenGL 环境的销毁通知
onSetMixTranscodingConfig:message:	设置云端的混流转码参数的回调，对应于 setMixTranscodingConfig 接口
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCaptureStopped:	当屏幕分享停止时，SDK 会通过此回调通知
onLocalRecordBegin:storagePath:	录制任务开始的事件回调
onLocalRecording:storagePath:	录制任务正在进行的进展事件回调
onLocalRecordComplete:storagePath:	录制任务已经结束的事件回调
onVoiceActivityDetectionUpdate:	调用 enableVoiceActivityDetection 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

V2 播放器相关接口

函数列表	描述
<code>setObserver:</code>	设置播放器回调
<code>setRenderView:</code>	设置播放器的视频渲染 View ，该控件负责显示视频内容
<code>setRenderRotation:</code>	设置播放器画面的旋转角度
<code>setRenderFillMode:</code>	设置画面的填充模式
<code>startLivePlay:</code>	开始播放音视频流
<code>stopPlay</code>	停止播放音视频流
<code>isPlaying</code>	播放器是否正在播放中
<code>pauseAudio</code>	暂停播放器的音频流
<code>resumeAudio</code>	恢复播放器的音频流
<code>pauseVideo</code>	暂停播放器的视频流
<code>resumeVideo</code>	恢复播放器的视频流
<code>setPlayoutVolume:</code>	设置播放器音量
<code>setCacheParams:maxTime:</code>	设置播放器缓存自动调整的最小和最大时间（单位：秒）
<code>switchStream:</code>	直播流无缝切换，支持 FLV 和 LEB
<code>getStreamList</code>	获取码流信息
<code>enableVolumeEvaluation:</code>	启用播放音量大小提示
<code>snapshot</code>	截取播放过程中的视频画面
<code>enableObserveVideoFrame:pixelFormat:bufferType:</code>	开启/关闭对视频帧的监听回调
<code>enableObserveAudioFrame:</code>	开启/关闭对音频数据的监听回调
<code>enableReceiveSeiMessage:payloadType:</code>	开启接收 SEI 消息
<code>enablePictureInPicture:</code>	开启画中画功能，仅支持直播和快直播播放

<code>showDebugView:</code>	是否显示播放器状态信息的调试浮层
<code>setProperty:value:</code>	调用 <code>V2TXLivePlayer</code> 的高级 API 接口
<code>startLocalRecording:</code>	开始录制音视频流
<code>stopLocalRecording</code>	停止录制音视频流

直播播放器事件回调

函数列表	描述
<code>onError:code:message:extraInfo:</code>	直播播放器错误通知，播放器出现错误时，会回调该通知
<code>onWarning:code:message:extraInfo:</code>	直播播放器警告通知
<code>onVideoResolutionChanged:width:height:</code>	直播播放器分辨率变化通知
<code>onConnected:extraInfo:</code>	已经成功连接到服务器
<code>onVideoPlaying:firstPlay:extraInfo:</code>	视频播放事件
<code>onAudioPlaying:firstPlay:extraInfo:</code>	音频播放事件
<code>onVideoLoading:extraInfo:</code>	视频加载事件
<code>onAudioLoading:extraInfo:</code>	音频加载事件
<code>onPlayoutVolumeUpdate:volume:</code>	播放器音量大小回调
<code>onStatisticsUpdate:statistics:</code>	直播播放器统计数据回调
<code>onSnapshotComplete:image:</code>	截图回调
<code>onRenderVideoFrame:frame:</code>	自定义视频渲染回调
<code>onPlayoutAudioFrame:frame:</code>	音频数据回调
<code>onReceiveSeiMessage:payloadType:data:</code>	收到 SEI 消息的回调，发送端通过 <code>V2TXLivePusher</code> 中的 <code>sendSeiMessage</code> 来发送 SEI 消息
<code>onStreamSwitched:url:code:</code>	分辨率无缝切换回调
<code>onPictureInPictureStateUpdate:state:message:extraInfo:</code>	画中画状态变更回调

<code>onLocalRecordBegin:errCode:storagePath:</code>	录制任务开始的事件回调
<code>onLocalRecording:durationMs:storagePath:</code>	录制任务正在进行的进展事件回调
<code>onLocalRecordComplete:errCode:storagePath:</code>	录制任务已经结束的事件回调

V2TXLive 高级接口

函数列表	描述
<code>getSDKVersionStr</code>	获取 SDK 版本号
<code>setObserver:</code>	设置 V2TXLivePremier 回调接口
<code>setLogConfig:</code>	设置 Log 的配置信息
<code>setEnvironment:</code>	设置 SDK 接入环境
<code>setLicence:key:</code>	设置 SDK 的授权 License
<code>setSocks5Proxy:port:username:password:config:</code>	设置 SDK socks5 代理配置
<code>enableAudioCaptureObserver:format:</code>	开启/关闭对音频采集数据的监听回调（可读写）
<code>enableAudioPlayoutObserver:format:</code>	开启/关闭对最终系统要播放出的音频数据的监听回调
<code>enableVoiceEarMonitorObserver:</code>	开启/关闭耳返音频数据的监听回调
<code>setUserId:</code>	设置 userId
<code>callExperimentalAPI:</code>	调用实验性 API 接口

V2TXLive 高级回调接口

函数列表	描述
<code>onLog:log:</code>	自定义 Log 输出回调接口
<code>onLicenceLoaded:Reason:</code>	setLicence 接口回调
<code>onCaptureAudioFrame:</code>	本地麦克风采集到的音频数据回调
<code>onPlayoutAudioFrame:</code>	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

`onVoiceEarMonitorAudioFrame:`

耳返的音频数据

人声相关的特效接口

函数列表	描述
<code>enableVoiceEarMonitor:</code>	开启耳返
<code>setVoiceEarMonitorVolume:</code>	设置耳返音量
<code>setVoiceReverbType:</code>	设置人声的混响效果
<code>setVoiceChangerType:</code>	设置人声的变声特效
<code>setVoiceVolume:</code>	设置语音音量
<code>setVoicePitch:</code>	设置语音音调

背景音乐的相关接口

函数列表	描述
<code>startPlayMusic:onStart:onProgress:onComplete:</code>	开始播放背景音乐
<code>stopPlayMusic:</code>	停止播放背景音乐
<code>pausePlayMusic:</code>	暂停播放背景音乐
<code>resumePlayMusic:</code>	恢复播放背景音乐
<code>setAllMusicVolume:</code>	设置所有背景音乐的本地音量和远端音量的大小
<code>setMusicPublishVolume:volume:</code>	设置某一首背景音乐的远端音量的大小
<code>setMusicPlayoutVolume:volume:</code>	设置某一首背景音乐的本地音量的大小
<code>setMusicPitch:pitch:</code>	调整背景音乐的音调高低
<code>setMusicSpeedRate:speedRate:</code>	调整背景音乐的变速效果
<code>getMusicCurrentPosInMS:</code>	获取背景音乐的播放进度（单位：毫秒）
<code>getMusicDurationInMS:</code>	获取背景音乐的总时长（单位：毫秒）

<code>seekMusicToPosInMS:pts:</code>	设置背景音乐的播放进度（单位：毫秒）
<code>setMusicScratchSpeedRate:speedRate:</code>	调整搓碟的变速效果
<code>preloadMusic:onProgress:onError:</code>	预加载背景音乐
<code>getMusicTrackCount:</code>	获取背景音乐的音轨数量
<code>setMusicTrack:track:</code>	指定背景音乐的播放音轨

美颜相关接口

函数列表	描述
<code>setBeautyStyle:</code>	设置美颜（磨皮）算法
<code>setBeautyLevel:</code>	设置美颜级别
<code>setWhitenessLevel:</code>	设置美白级别
<code>enableSharpnessEnhancement:</code>	开启清晰度增强
<code>setRuddyLevel:</code>	设置红润级别
<code>setFilter:</code>	设置色彩滤镜效果
<code>setFilterStrength:</code>	设置色彩滤镜的强度
<code>setGreenScreenFile:</code>	设置绿幕背景视频
<code>setEyeScaleLevel:</code>	设置大眼级别
<code>setFaceSlimLevel:</code>	设置瘦脸级别
<code>setFaceVLevel:</code>	设置 V 脸级别
<code>setChinLevel:</code>	设置下巴拉伸或收缩
<code>setFaceShortLevel:</code>	设置短脸级别
<code>setFaceNarrowLevel:</code>	设置窄脸级别
<code>setNoseSlimLevel:</code>	设置瘦鼻级别
<code>setEyeLightenLevel:</code>	设置亮眼级别
<code>setToothWhitenLevel:</code>	设置牙齿美白级别

setWrinkleRemoveLevel:	设置祛皱级别
setPouchRemoveLevel:	设置祛眼袋级别
setSmileLinesRemoveLevel:	设置法令纹去除级别
setForeheadLevel:	设置发际线调整级别
setEyeDistanceLevel:	设置眼距
setEyeAngleLevel:	设置眼角调整级别
setMouthShapeLevel:	设置嘴型调整级别
setNoseWingLevel:	设置鼻翼调整级别
setNosePositionLevel:	设置鼻子位置
setLipsThicknessLevel:	设置嘴唇厚度
setFaceBeautyLevel:	设置脸型
setMotionTpl:inDir:	选择 AI 动效挂件
setMotionMute:	是否在动效素材播放时静音

音视频设备相关的类型定义

函数列表	描述
onDeviceChanged:type:state:	本地设备的通断状态发生变化（仅适用于桌面系统）

设备操作接口

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera:	切换前置或后置摄像头（仅适用于移动端）
isCameraZoomSupported	查询当前摄像头是否支持缩放（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio:	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus:	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition:	设置摄像头的对焦位置（仅适用于移动端）
isCameraTorchSupported	查询是否支持开启闪光灯（仅适用于移动端）
enableCameraTorch:	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute:	设置音频路由（仅适用于移动端）
setExposureCompensation:	设置摄像头的曝光参数，取值范围从-1到1
getDevicesList:	获取设备列表（仅适用于桌面端）
setCurrentDevice:deviceId:	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice:	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume:deviceType:	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume:	获取当前设备的音量（仅适用于桌面端）
setCurrentDeviceMute:deviceType:	设置当前设备的静音状态（仅适用于桌面端）
getCurrentDeviceMute:	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice:enable:	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest:	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest:	开始麦克风测试（仅适用于桌面端）
startMicDeviceTest:playback:	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest:	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
setObserver:	设备热插拔回调（仅适用于 Mac 系统）

setCameraCapturerParam:	设置摄像头采集偏好
-------------------------	-----------

弃用接口

函数列表	描述
setSystemVolumeType:	设置系统音量类型（仅适用于移动端）

V2TXLivePusher

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePusher @ TXLiteAVSDK

Function: 腾讯云直播推流器

功能

腾讯云直播推流器

介绍

主要负责将本地的音频和视频画面进行编码，并推送到指定的推流地址，支持任意的推流服务端。

推流器包含如下能力：

自定义的视频采集，让您可以根据项目需要定制自己的音视频数据源。

美颜、滤镜、贴纸，包含多套美颜磨皮算法（自然&光滑）和多款色彩空间滤镜（支持自定义滤镜）。

Qos 流量控制技术，具备上行网络自适应能力，可以根据主播端网络的具体情况实时调节音视频数据量。

脸型调整、动效挂件，支持基于优图 AI 人脸识别技术的大眼、瘦脸、隆鼻等脸型微调以及动效挂件效果，只需要购买 优图 License 就可以轻松实现丰富的直播效果。

V2TXLivePusher

V2TXLivePusher

函数列表	描述
setObserver:	设置推流器回调
setRenderView:	设置本地摄像头预览 View
setRenderMirror:	设置本地摄像头预览镜像

setEncoderMirror:	设置视频编码镜像
setRenderRotation:	设置本地摄像头预览画面的旋转角度
setRenderFillMode:	设置本地摄像头预览画面的填充模式
startCamera:	打开本地摄像头
stopCamera	关闭本地摄像头
startMicrophone	打开麦克风
stopMicrophone	关闭麦克风
startVirtualCamera:	开启图片推流
stopVirtualCamera	关闭图片推流
startScreenCapture:	开始全系统的屏幕分享（该接口支持 iOS 11.0 及以上的 iPhone 和 iPad）
stopScreenCapture	关闭屏幕采集
pauseAudio	静音本地音频
resumeAudio	取消静音本地音频
pauseVideo	暂停推流器的视频流
resumeVideo	恢复推流器的视频流
startPush:	开始音视频数据推流
stopPush	停止推送音视频数据
isPushing	当前推流器是否正在推流中
setAudioQuality:	设置推流音频质量
setVideoQuality:	设置推流视频编码参数
getAudioEffectManager	获取音效管理对象
getBeautyManager	获取美颜管理对象
getDeviceManager	获取设备管理对象
snapshot	截取推流过程中的本地画面
setWatermark:x:y:scale:	设置推流器水印。默认情况下，水印不开启

enableVolumeEvaluation:	启用采集音量大小提示
enableCustomVideoProcess:pixelFormat:bufferType:	开启/关闭自定义视频处理
enableCustomVideoCapture:	开启/关闭自定义视频采集
enableCustomAudioCapture:	开启/关闭自定义音频采集
sendCustomVideoFrame:	在自定义视频采集模式下，将采集的视频数据发送到SDK
sendCustomAudioFrame:	在自定义音频采集模式下，将采集的音频数据发送到SDK
enableAudioProcessObserver:format:	开启/关闭对经过前处理后的本地音频帧的监听回调
sendSeiMessage:data:	发送 SEI 消息
showDebugView:	显示仪表盘
setProperty:value:	调用 V2TXLivePusher 的高级 API 接口。
setMixTranscodingConfig:	设置云端的混流转码参数
startLocalRecording:	开始录制音视频流
stopLocalRecording	停止录制音视频流

setObserver:

setObserver:

- (void)setObserver:	(id< V2TXLivePusherObserver >)observer
----------------------	--

设置推流器回调

通过设置回调，可以监听 [V2TXLivePusher](#) 推流器的一些回调事件，包括推流器状态、音量回调、统计数据、警告和错误信息等。

参数	描述
observer	推流器的回调目标对象，更多信息请查看 V2TXLivePusherObserver 。

setRenderView:

setRenderView:

- (V2TXLiveCode)setRenderView:	(TXView *)view
--------------------------------	----------------

设置本地摄像头预览 View

本地摄像头采集到的画面，经过美颜、脸型调整、滤镜等多种效果叠加之后，最终会显示到传入的 View 上。

参数	描述
view	本地摄像头预览 View。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK：成功。

setRenderMirror:

setRenderMirror:

- (V2TXLiveCode)setRenderMirror:	(V2TXLiveMirrorType)mirrorType
----------------------------------	--

设置本地摄像头预览镜像

本地摄像头分为前置摄像头和后置摄像头，系统默认情况下，是前置摄像头镜像，后置摄像头不镜像，这里可以修改前置后置摄像头的默认镜像类型。

参数	描述
mirrorType	摄像头镜像类型 V2TXLiveMirrorType 。 V2TXLiveMirrorTypeAuto 【默认值】：默认镜像类型. 在这种情况下，前置摄像头的画面是镜像的，后置摄像头的画面不是镜像的。 V2TXLiveMirrorTypeEnable: 前置摄像头 和 后置摄像头，都切换为镜像模式。 V2TXLiveMirrorTypeDisable: 前置摄像头 和 后置摄像头，都切换为非镜像模式。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setEncoderMirror:

setEncoderMirror:

- (V2TXLiveCode)setEncoderMirror:	(BOOL)mirror
-----------------------------------	--------------

设置视频编码镜像

参数	描述
mirror	是否镜像。 NO【默认值】: 播放端看到的是非镜像画面。 YES: 播放端看到的是镜像画面。

注意

编码镜像只影响观众端看到的视频效果。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setRenderRotation:

setRenderRotation:

- (V2TXLiveCode)setRenderRotation:	(V2TXLiveRotation)rotation
------------------------------------	--

设置本地摄像头预览画面的旋转角度

参数	描述
rotation	预览画面的旋转角度 V2TXLiveRotation 。 V2TXLiveRotation0【默认值】: 0度, 不旋转。 V2TXLiveRotation90: 顺时针旋转90度。 V2TXLiveRotation180: 顺时针旋转180度。 V2TXLiveRotation270: 顺时针旋转270度。

注意

只旋转本地预览画面，不影响推流出去的画面。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setRenderFillMode:

setRenderFillMode:

- (V2TXLiveCode)setRenderFillMode:	(V2TXLiveFillMode)mode
------------------------------------	--

设置本地摄像头预览画面的填充模式

参数	描述
mode	画面填充模式 V2TXLiveFillMode 。 V2TXLiveFillModeFill 【默认值】: 图像铺满屏幕，不留黑边，如果图像宽高比不同于屏幕宽高比，部分画面内容会被裁剪掉。 V2TXLiveFillModeFit: 图像适应屏幕，保持画面完整，但如果图像宽高比不同于屏幕宽高比，会有黑边的存在。 V2TXLiveFillModeScaleFill: 图像拉伸铺满，因此长度和宽度可能不会按比例变化。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startCamera:

startCamera:

- (V2TXLiveCode)startCamera:	(BOOL)frontCamera
------------------------------	-------------------

打开本地摄像头

参数	描述
frontCamera	是否为前置摄像头。 YES 【默认值】: 切换到前置摄像头。 NO: 切换到后置摄像头。

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下, 仅有一个采集源可以上行, 不同采集源之间切换, 请先关闭前一采集源, 再开启后一采集源, 保证同一采集源的开启和关闭是成对调用的。比如: 采集源从Camera切换到VirtualCamera, 调用顺序是 startCamera -> stopCamera -> startVirtualCamera。

返回值说明:

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopCamera

stopCamera

关闭本地摄像头

返回值说明:

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startMicrophone

startMicrophone

打开麦克风

返回值说明:

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopMicrophone

stopMicrophone

关闭麦克风

返回值说明:

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startVirtualCamera:

startVirtualCamera:

- (V2TXLiveCode)startVirtualCamera:	(TXImage *)image
-------------------------------------	------------------

开启图片推流

参数	描述
image	图片。

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下, 仅有一个采集源可以上行, 不同采集源之间切换, 请先关闭前一采集源, 再开启后一采集源, 保证同一采集源的开启和关闭是成对调用的。比如: 采集源从Camera切换到VirtualCamera, 调用顺序是 startCamera -> stopCamera -> startVirtualCamera。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopVirtualCamera

stopVirtualCamera

关闭图片推流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startScreenCapture:

startScreenCapture:

- (V2TXLiveCode)startScreenCapture:	(NSString *)appGroup
-------------------------------------	----------------------

开始全系统的屏幕分享（该接口支持 iOS 11.0 及以上的 iPhone 和 iPad）

--	--

参数	描述
appGroup	主 App 与 Broadcast 共享的 Application Group Identifier，可以指定为 nil，但按照文档设置会使功能更加可靠。

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下，仅有一个采集源可以上行，不同采集源之间切换，请先关闭前一采集源，再开启后一采集源，保证同一采集源的开启和关闭是成对调用的。比如：采集源从Camera切换到ScreenCapture，调用顺序是 startCamera -> stopCamera -> startScreenCapture。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_ERROR_NOT_SUPPORTED: 功能不支持。

stopScreenCapture

stopScreenCapture

关闭屏幕采集

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseAudio

pauseAudio

静音本地音频

静音本地音频后，SDK 不会继续采集麦克风的声音。

与 stopMicrophone 不同之处在于 pauseAudio 并不会停止发送音频数据，而是继续发送码率极低的静音包。

由于 MP4 等视频文件格式，对于音频的连续性是要求很高的，使用 stopMicrophone 会导致录制出的 MP4 不易播放，

因此在对录制质量要求很高的场景中，建议选择 pauseAudio，从而录制出兼容性更好的 MP4 文件。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeAudio

resumeAudio

取消静音本地音频

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseVideo

pauseVideo

暂停推流器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeVideo

resumeVideo

恢复推流器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startPush:

startPush:

- (V2TXLiveCode)startPush:	(NSString *)url
----------------------------	-----------------

开始音视频数据推流

参数	描述
url	推流的目标地址，支持任意推流服务端。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 操作成功，开始连接推流目标地址。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，url 不合法。

V2TXLIVE_ERROR_INVALID_LICENSE: 操作失败，license 不合法，鉴权失败。

V2TXLIVE_ERROR_REFUSED: 操作失败，RTC 不支持同一设备上同时推拉同一个 StreamId。

stopPush

stopPush

停止推送音视频数据

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

isPushing

isPushing

当前推流器是否正在推流中

返回值说明：

是否正在推流。

1: 正在推流中。

0: 已经停止推流。

setAudioQuality:

setAudioQuality:

--	--

- (V2TXLiveCode)setAudioQuality:	(V2TXLiveAudioQuality)quality
----------------------------------	-------------------------------

设置推流音频质量

参数	描述
quality	音频质量 V2TXLiveAudioQuality 。 V2TXLiveAudioQualityDefault 【默认值】：通用。 V2TXLiveAudioQualitySpeech: 语音。 V2TXLiveAudioQualityMusic: 音乐。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 推流过程中，不允许调整音质。

setVideoQuality:

setVideoQuality:

- (V2TXLiveCode)setVideoQuality:	(V2TXLiveVideoEncoderParam *)param
----------------------------------	------------------------------------

设置推流视频编码参数

参数	描述
param	视频编码参数 V2TXLiveVideoEncoderParam 。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

getAudioEffectManager

getAudioEffectManager

获取音效管理对象

通过音效管理，您可以使用以下功能：

调整麦克风收集的人声音量。

设置混响和变声效果。

开启耳返，设置耳返音量。

添加背景音乐，调整背景音乐的播放效果。

参考 [TXAudioEffectManager](#)

getBeautyManager

getBeautyManager

获取美颜管理对象

通过美颜管理，您可以使用以下功能：

设置“美颜风格”、“美白”、“红润”、“大眼”、“瘦脸”、“V脸”、“下巴”、“短脸”、“小鼻”、“亮眼”、“白牙”、“祛眼袋”、“祛皱纹”、“祛法令纹”等美容效果。

调整“发际线”、“眼间距”、“眼角”、“嘴形”、“鼻翼”、“鼻子位置”、“嘴唇厚度”、“脸型”。

设置人脸挂件（素材）等动态效果。

添加美妆。

进行手势识别。

参考 [TXBeautyManager](#)

getDeviceManager

getDeviceManager

获取设备管理对象

通过设备管理，您可以使用以下功能：

切换前后摄像头。

设置自动聚焦。

设置摄像头缩放倍数。

打开或关闭闪光灯。

切换耳机或者扬声器。

修改音量类型(媒体音量或者通话音量)。

参考 [TXDeviceManager](#)

snapshot

snapshot

截取推流过程中的本地画面

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 已经停止推流，不允许调用截图操作。

setWatermark:x:y:scale:

setWatermark:x:y:scale:

- (V2TXLiveCode)setWatermark:	(TXImage *)image
x:	(float)x
y:	(float)y
scale:	(float)scale

设置推流器水印。默认情况下，水印不开启

参数	描述
image	水印图片。如果该值为 nil，则等效于禁用水印。
scale	水印图片的缩放比例，取值范围为0 - 1的浮点数。
x	水印的横坐标，取值范围为0 - 1的浮点数。
y	水印的纵坐标，取值范围为0 - 1的浮点数。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enableVolumeEvaluation:

enableVolumeEvaluation:

- (V2TXLiveCode)enableVolumeEvaluation:	(NSInteger)intervalMs
---	-----------------------

启用采集音量大小提示

开启后可以在 [onMicrophoneVolumeUpdate](#) 回调中获取到 SDK 对音量大小值的评估。

参数	描述
intervalMs	决定了音量大小回调的触发间隔，单位为 ms，最小间隔为 100ms，如果小于等于0则会关闭回调，建议设置为 300ms。【默认值】：0，不开启。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enableCustomVideoProcess:pixelFormat:bufferType:

enableCustomVideoProcess:pixelFormat:bufferType:

- (V2TXLiveCode)enableCustomVideoProcess:	(BOOL)enable
pixelFormat:	(V2TXLivePixelFormat)pixelFormat
bufferType:	(V2TXLiveBufferType)bufferType

开启/关闭自定义视频处理

参数	描述
bufferType	指定回调的数据格式。
enable	YES: 开启; NO: 关闭。【默认值】：NO。
pixelFormat	指定回调的像素格式。

注意

支持的格式组合：

V2TXLivePixelFormatTexture2D+V2TXLiveBufferTypeTexture

V2TXLivePixelFormatNV12+V2TXLiveBufferTypePixelFormatPixelBuffer

V2TXLivePixelFormatBGRA32+V2TXLiveBufferTypePixelFormatPixelBuffer

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_NOT_SUPPORTED: 不支持的格式。

enableCustomVideoCapture:

enableCustomVideoCapture:

- (V2TXLiveCode)enableCustomVideoCapture:	(BOOL)enable
---	--------------

开启/关闭自定义视频采集

在自定义视频采集模式下，SDK 不再从摄像头采集图像，只保留编码和发送能力。

参数	描述
enable	YES：开启自定义采集；NO：关闭自定义采集。【默认值】：NO。

注意

需要在 [startPush](#) 之前调用，才会生效。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enableCustomAudioCapture:

enableCustomAudioCapture:

- (V2TXLiveCode)enableCustomAudioCapture:	(BOOL)enable
---	--------------

开启/关闭自定义音频采集

@brief 开启/关闭自定义音频采集。

在自定义音频采集模式下，SDK 不再从麦克风采集声音，只保留编码和发送能力。

参数	描述
enable	YES: 开启自定义采集; NO: 关闭自定义采集。【默认值】: NO。

注意

需要在 [startPush](#) 前调用才会生效。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

sendCustomVideoFrame:

sendCustomVideoFrame:

- (V2TXLiveCode)sendCustomVideoFrame:	(V2TXLiveVideoFrame *)videoFrame
---------------------------------------	--

在自定义视频采集模式下，将采集的视频数据发送到SDK

在自定义视频采集模式下，SDK不再采集摄像头数据，仅保留编码和发送功能。

您可以把采集到的 [SampleBuffer](#) 打包到 [V2TXLiveVideoFrame](#) 中，然后通过该API定期的发送。

参数	描述
videoFrame	向 SDK 发送的 视频帧数据 V2TXLiveVideoFrame 。

注意

需要在 [startPush](#) 之前调用 [enableCustomVideoCapture](#) 开启自定义采集。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 发送失败，视频帧数据不合法。

sendCustomAudioFrame:

sendCustomAudioFrame:

--	--

- (V2TXLiveCode)sendCustomAudioFrame:	(V2TXLiveAudioFrame *)audioFrame
---------------------------------------	----------------------------------

在自定义音频采集模式下，将采集的音频数据发送到SDK

@info 在自定义音频采集模式下，将采集的音频数据发送到SDK，SDK不再采集麦克风数据，仅保留编码和发送功能。

@param audioFrame 向 SDK 发送的 音频帧数据 [V2TXLiveAudioFrame](#)。

@return 返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 发送失败，音频帧数据不合法。

@note 需要在 [startPush](#) 之前调用 [enableCustomAudioCapture](#) 开启自定义采集。

enableAudioProcessObserver:format:

enableAudioProcessObserver:format:

- (V2TXLiveCode)enableAudioProcessObserver:	(BOOL)enable
format:	(V2TXLiveAudioFrameObserverFormat *)format

开启/关闭对经过前处理后的本地音频帧的监听回调

参数	描述
enable	是否开启。【默认值】：false。
format	设置回调出的 AudioFrame 的格式。

注意

需要在 [startPush](#) 之前调用，才会生效。

sendSeiMessage:data:

sendSeiMessage:data:

- (V2TXLiveCode)sendSeiMessage:	(int)payloadType
data:	(NSData *)data

发送 SEI 消息

播放端 [V2TXLivePlayer](#) 通过 [V2TXLivePlayerObserver](#) 中的 `onReceiveSeiMessage` 回调来接收该消息。

参数	描述
data	待发送的数据。
payloadType	数据类型，支持 5、242。推荐填：242。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

showDebugView:

showDebugView:

- (void)showDebugView:	(BOOL)isShow
------------------------	--------------

显示仪表盘

参数	描述
isShow	是否显示。【默认值】：NO。

setProperty:value:

setProperty:value:

- (V2TXLiveCode)setProperty:	(NSString *)key
value:	(NSObject *)value

调用 [V2TXLivePusher](#) 的高级 API 接口。

参数	描述
key	高级 API 对应的 key, 详情请参考 V2TXLiveProperty 定义。
value	调用 key 所对应的高级 API 时，需要的参数。

注意

该接口用于调用一些高级功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，key 不允许为空。

setMixTranscodingConfig:

setMixTranscodingConfig:

- (V2TXLiveCode)setMixTranscodingConfig:	(V2TXLiveTranscodingConfig *)config
--	-------------------------------------

设置云端的混流转码参数

如果您在实时音视频 [控制台](#) 中的功能配置页开启了“启用旁路推流”功能，房间里的每一路画面都会有一个默认的直播 **CDN 地址**。

一个直播间中可能有不止一位主播，而且每个主播都有自己的画面和声音，但对于 **CDN** 观众来说，他们只需要一路直播流，所以您需要将多路音视频流混成一路标准的直播流，这就需要混流转码

当您调用 `setMixTranscodingConfig` 接口时，SDK 会向腾讯云的转码服务器发送一条指令，目的是将房间里的多路音视频流混合为一，您可以通过 `mixUsers` 参数来调整每一路画面的位置，以及是否只混合声音，也可以通过 `videoWidth`、`videoHeight`、`videoBitrate` 等参数控制混合音视频流的编码参数。

```
【画面1】=> 解码====> \\\n\n【画面2】=> 解码=> 画面混合=> 编码=> 【混合后的画面】\n\n【画面3】=> 解码====> /
```

```
【声音1】=> 解码====> \\  
\\  
【声音2】=> 解码=> 声音混合=> 编码=> 【混合后的声音】  
/  
【声音3】=> 解码====> /
```

参考文档：[云端混流转码](#)。

参数	描述
config	请参考 V2TXLiveDef.h 中关于 V2TXLiveTranscodingConfig 的介绍。如果传入 nil 则取消云端混流转码。

注意

关于云端混流的注意事项：

云端转码会引入一定的 CDN 观看延时，大概会增加1 - 2秒。

调用该函数的用户，会将连麦中的多路画面混合到自己当前这路画面或者 config 中指定的 streamId 上。

请注意，若您还在房间中且不再需要混流，请务必传入 nil 进行取消，因为当您发起混流后，云端混流模块就会开始工作，不及时取消混流可能会引起不必要的计费损失。

请放心，您退房时会自动取消混流状态。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 未开启推流时，不允许设置混流转码参数。

startLocalRecording:

startLocalRecording:

- (V2TXLiveCode)startLocalRecording:	(V2TXLiveLocalRecordingParams *)params
--------------------------------------	---

开始录制音视频流

注意

推流开启后才能开始录制，非推流状态下开启录制无效。

录制过程中不要动态切换分辨率和软/硬剪辑，生成的视频极有可能出现异常。

返回值说明：

返回值 `V2TXLiveCode`。

`V2TXLIVE_OK` : 成功。

`V2TXLIVE_ERROR_INVALID_PARAMETER` : 参数不合法，比如filePath 为空。

`V2TXLIVE_ERROR_REFUSED` : API被拒绝，推流尚未开始。

stopLocalRecording

stopLocalRecording

停止录制音视频流

注意

当停止推流后，如果视频还在录制中，SDK 内部会自动结束录制。

V2TXLivePusherObserver

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePusherObserver @ TXLiteAVSDK

Function: 腾讯云直播推流的回调通知

功能

腾讯云直播的推流回调通知。

介绍

可以接收 [V2TXLivePusher](#) 推流器的一些推流通知，包括推流器连接状态、音视频首帧回调、统计数据、警告和错误信息等。

V2TXLivePusherObserver

V2TXLivePusherObserver

函数列表	描述
onError:message:extraInfo:	直播推流器错误通知，推流器出现错误时，会回调该通知
onWarning:message:extraInfo:	直播推流器警告通知
onCaptureFirstAudioFrame	首帧音频采集完成的回调通知
onCaptureFirstVideoFrame	首帧视频采集完成的回调通知
onMicrophoneVolumeUpdate:	麦克风采集音量值回调
onPushStatusUpdate:message:extraInfo:	推流器连接状态回调通知
onStatisticsUpdate:	直播推流器统计数据回调

<code>onSnapshotComplete:</code>	截图回调
<code>onProcessAudioFrame:</code>	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
<code>onProcessVideoFrame:dstFrame:</code>	自定义视频处理回调
<code>onGLContextDestroyed</code>	SDK 内部的 OpenGL 环境的销毁通知
<code>onSetMixTranscodingConfig:message:</code>	设置云端的混流转码参数的回调，对应于 <code>setMixTranscodingConfig</code> 接口
<code>onScreenCaptureStarted</code>	当屏幕分享开始时，SDK 会通过此回调通知
<code>onScreenCaptureStopped:</code>	当屏幕分享停止时，SDK 会通过此回调通知
<code>onLocalRecordBegin:storagePath:</code>	录制任务开始的事件回调
<code>onLocalRecording:storagePath:</code>	录制任务正在进行的进展事件回调
<code>onLocalRecordComplete:storagePath:</code>	录制任务已经结束的事件回调
<code>onVoiceActivityDetectionUpdate:</code>	调用 <code>enableVoiceActivityDetection</code> 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

onError:message:extraInfo:

onError:message:extraInfo:

- (void)onError:	(V2TXLiveCode)code
message:	(NSString *)msg
extraInfo:	(NSDictionary *)extraInfo

直播推流器错误通知，推流器出现错误时，会回调该通知

参数	描述
code	错误码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	错误信息。

onWarning:message:extraInfo:

onWarning:message:extraInfo:

- (void)onWarning:	(V2TXLiveCode)code
message:	(NSString *)msg
extraInfo:	(NSDictionary *)extraInfo

直播推流器警告通知

参数	描述
code	警告码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	警告信息。

onCaptureFirstAudioFrame

onCaptureFirstAudioFrame

首帧音频采集完成的回调通知

onCaptureFirstVideoFrame

onCaptureFirstVideoFrame

首帧视频采集完成的回调通知

onMicrophoneVolumeUpdate:

onMicrophoneVolumeUpdate:

- (void)onMicrophoneVolumeUpdate:	(NSInteger)volume
-----------------------------------	-------------------

麦克风采集音量值回调

参数	描述
----	----

volume	音量大小。
--------	-------

注意

调用 [enableVolumeEvaluation](#) 开启采集音量大小提示之后，会收到这个回调通知。

onPushStatusUpdate:message:extraInfo:

onPushStatusUpdate:message:extraInfo:

- (void)onPushStatusUpdate:	(V2TXLivePushStatus)status
message:	(NSString *)msg
extraInfo:	(NSDictionary *)extraInfo

推流器连接状态回调通知

参数	描述
extraInfo	扩展信息。
msg	连接状态信息。
status	推流器连接状态 V2TXLivePushStatus 。

onStatisticsUpdate:

onStatisticsUpdate:

- (void)onStatisticsUpdate:	(V2TXLivePusherStatistics *)statistics
-----------------------------	---

直播推流器统计数据回调

参数	描述
statistics	推流器统计数据 V2TXLivePusherStatistics

onSnapshotComplete:

onSnapshotComplete:

- (void)onSnapshotComplete:	(nullable TXImage *)image
-----------------------------	---------------------------

截图回调

参数	描述
image	已截取的视频画面。

注意

调用 [snapshot](#) 截图之后，会收到这个回调通知。

onProcessAudioFrame:**onProcessAudioFrame:**

- (void) onProcessAudioFrame:	(V2TXLiveAudioFrame *)frame
-------------------------------	-----------------------------

本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理、音效处理和混 BGM 之后的数据，在最终进行网络编码之前，以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长固定为 0.02s，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 $\text{采样率} \times \text{时间帧长} \times \text{声道数} \times \text{采样点位宽}$ 。

以 SDK 默认的音频录制格式 48000 采样率、单声道、16 采样点位宽为例，字节帧长为 $48000 \times 0.02s \times 1 \times 16\text{bit} = 15360\text{bit} = 1920\text{字节}$ 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

onProcessVideoFrame:dstFrame:**onProcessVideoFrame:dstFrame:**

- (void)onProcessVideoFrame:	(V2TXLiveVideoFrame * _Nonnull)srcFrame
dstFrame:	(V2TXLiveVideoFrame * _Nonnull)dstFrame

自定义视频处理回调

参数	描述
dstFrame	用于承载处理过的视频画面。
srcFrame	用于承载未处理的视频画面。

注意

需要调用 [enableCustomVideoProcess](#) 开启自定义视频处理，才会收到这个回调通知。

【情况一】美颜组件会产生新的纹理

如果您使用的美颜组件会在处理图像的过程中产生一帧全新的纹理（用于承载处理后的图像），那请您在回调函数中将 `dstFrame.textureId` 设置为新纹理的 ID。

```

// ObjectiveC
(void) onProcessVideoFrame:(V2TXLiveVideoFrame * _Nonnull)srcFrame dstFrame:(V2TXLiveVideoFrame *
_Nonnull)dstFrame
{
    GLuint dstTextureId = renderItemWithTexture(srcFrame.textureId, srcFrame.width, srcFrame.height);
    dstFrame.textureId = dstTextureId;
    return 0;
}
// Swift
```

【情况二】美颜组件并不自身产生新纹理

如果您使用的第三方美颜模块并不生成新的纹理，而是需要您设置给该模块一个输入纹理和一个输出纹理，则可以考虑如下方案：

```

// ObjectiveC
(void) onProcessVideoFrame:(V2TXLiveVideoFrame * _Nonnull)srcFrame dstFrame:(V2TXLiveVideoFrame *
_Nonnull)dstFrame
{
    thirdparty_process(srcFrame.textureId, srcFrame.width, srcFrame.height, dstFrame.textureId);
    return 0;
}
// Swift
```

onGLContextDestroyed

onGLContextDestroyed

SDK 内部的 OpenGL 环境的销毁通知

onSetMixTranscodingConfig:message:

onSetMixTranscodingConfig:message:

- (void)onSetMixTranscodingConfig:	(V2TXLiveCode)code
message:	(NSString *)msg

设置云端的混流转码参数的回调，对应于 {@link setMixTranscodingConfig} 接口

参数	描述
code	0表示成功，其余值表示失败。
msg	具体错误原因。

onScreenCaptureStarted

onScreenCaptureStarted

当屏幕分享开始时，SDK 会通过此回调通知

onScreenCaptureStopped:

onScreenCaptureStopped:

- (void)onScreenCaptureStopped:	(int)reason
---------------------------------	-------------

当屏幕分享停止时，SDK 会通过此回调通知

参数	描述
reason	停止原因 0：表示用户主动停止。 1：iOS 表示录屏被系统中断；Mac、Windows 表示屏幕分享窗口被关闭。

2：Windows 表示屏幕分享的显示屏状态变更（如接口被拔出、投影模式变更等）；其他平台不抛出。

onLocalRecordBegin:storagePath:

onLocalRecordBegin:storagePath:

- (void)onLocalRecordBegin:	(NSInteger)errCode
storagePath:	(NSString *)storagePath

录制任务开始的事件回调

参数	描述
code	状态码。 0：录制任务启动成功。 -1：内部错误导致录制任务启动失败。 -2：文件后缀名有误（比如不支持的录制格式）。 -6：录制已经启动，需要先停止录制。 -7：录制文件已存在，需要先删除文件。 -8：录制目录无写入权限，请检查目录权限问题。
storagePath	录制的文件地址。

onLocalRecording:storagePath:

onLocalRecording:storagePath:

- (void)onLocalRecording:	(NSInteger)durationMs
storagePath:	(NSString *)storagePath

录制任务正在进行的进展事件回调

参数	描述
durationMs	录制时长。
storagePath	录制的文件地址。

onLocalRecordComplete:storagePath:

onLocalRecordComplete:storagePath:

- (void)onLocalRecordComplete:	(NSInteger)errCode
storagePath:	(NSString *)storagePath

录制任务已经结束的事件回调

参数	描述
code	状态码。 0：结束录制任务成功。 -1：录制失败。 -2：切换分辨率或横竖屏导致录制结束。 -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
storagePath	录制的文件地址。

onVoiceActivityDetectionUpdate:

onVoiceActivityDetectionUpdate:

- (void)onVoiceActivityDetectionUpdate:	(BOOL)active
---	--------------

调用 {@link enableVoiceActivityDetection} 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

参数	描述
active	人声开始或停止。

V2TXLivePlayer

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePlayer @ TXLiteAVSDK

Function: 腾讯云直播播放器

功能

腾讯云直播播放器。
主要负责从指定的直播流地址拉取音视频数据，并进行解码和本地渲染播放。

介绍

播放器包含如下能力：

- 支持 RTMP、HTTP-FLV、HLS、TRTC、WebRTC 协议。
- 屏幕截图，可以截取当前直播流的视频画面。
- 延时调节，可以设置播放器缓存自动调整的最小和最大时间。
- 自定义的视频数据处理，您可以根据项目需要处理直播流中的视频数据后，再进行渲染以及播放。

V2TXLivePlayer

V2TXLivePlayer

函数列表	描述
setObserver:	设置播放器回调
setRenderView:	设置播放器的视频渲染 View，该控件负责显示视频内容
setRenderRotation:	设置播放器画面的旋转角度
setRenderFillMode:	设置画面的填充模式

startLivePlay:	开始播放音视频流
stopPlay	停止播放音视频流
isPlaying	播放器是否正在播放中
pauseAudio	暂停播放器的音频流
resumeAudio	恢复播放器的音频流
pauseVideo	暂停播放器的视频流
resumeVideo	恢复播放器的视频流
setPlayoutVolume:	设置播放器音量
setCacheParams:maxTime:	设置播放器缓存自动调整的最小和最大时间 (单位：秒)
switchStream:	直播流无缝切换，支持 FLV 和 LEB
getStreamList	获取码流信息
enableVolumeEvaluation:	启用播放音量大小提示
snapshot	截取播放过程中的视频画面
enableObserveVideoFrame:pixelFormat:bufferType:	开启/关闭对视频帧的监听回调
enableObserveAudioFrame:	开启/关闭对音频数据的监听回调
enableReceiveSeiMessage:payloadType:	开启接收 SEI 消息
enablePictureInPicture:	开启画中画功能，仅支持直播和快直播播放
showDebugView:	是否显示播放器状态信息的调试浮层
setProperty:value:	调用 V2TXLivePlayer 的高级 API 接口
startLocalRecording:	开始录制音视频流
stopLocalRecording	停止录制音视频流

setObserver:

setObserver:

--	--

- (void)setObserver:	(id< V2TXLivePlayerObserver >)observer
----------------------	--

设置播放器回调

通过设置回调，可以监听 [V2TXLivePlayer](#) 播放器的一些回调事件，包括播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等。

参数	描述
observer	播放器的回调目标对象，更多信息请查看 V2TXLivePlayerObserver

setRenderView:

setRenderView:

- (V2TXLiveCode)setRenderView:	(TXView *)view
--------------------------------	----------------

设置播放器的视频渲染 **View**，该控件负责显示视频内容

参数	描述
view	播放器渲染 View 。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK：成功。

setRenderRotation:

setRenderRotation:

- (V2TXLiveCode)setRenderRotation:	(V2TXLiveRotation)rotation
------------------------------------	--

设置播放器画面的旋转角度

参数	描述
rotation	旋转角度 V2TXLiveRotation 。 V2TXLiveRotation0 【默认值】：0度，不旋转。 V2TXLiveRotation90：顺时针旋转90度。 V2TXLiveRotation180：顺时针旋转180度。

V2TXLiveRotation270: 顺时针旋转270度。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setRenderFillMode:

setRenderFillMode:

- (V2TXLiveCode)setRenderFillMode:	(V2TXLiveFillMode)mode
------------------------------------	--

设置画面的填充模式

参数	描述
mode	画面填充模式 V2TXLiveFillMode 。 V2TXLiveFillModeFill 【默认值】: 图像铺满屏幕，不留黑边，如果图像宽高比不同于屏幕宽高比，部分画面内容会被裁剪掉。 V2TXLiveFillModeFit: 图像适应屏幕，保持画面完整，但如果图像宽高比不同于屏幕宽高比，会有黑边的存在。 V2TXLiveFillModeScaleFill: 图像拉伸铺满，因此长度和宽度可能不会按比例变化。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startLivePlay:

startLivePlay:

- (V2TXLiveCode)startLivePlay:	(NSString *)url
--------------------------------	-----------------

开始播放音视频流

参数	描述
url	音视频流的播放地址，支持 RTMP，HTTP-FLV，TRTC。

注意

10.7 版本开始，需要通过 [setLicence](#) 或者 [setLicence](#) 设置 Licence 后方可成功播放，否则将播放失败（黑屏），全局仅设置一次即可。直播 Licence、短视频 Licence 和视频播放 Licence 均可使用，若您暂未获取上述 Licence，可[快速免费申请测试版 Licence](#) 以正常播放，正式版 Licence 需[购买](#)。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 操作成功，开始连接并播放。

V2TXLIVE_ERROR_INVALID_PARAMETER：操作失败，url 不合法。

V2TXLIVE_ERROR_REFUSED：RTC 不支持同一设备上同时推拉同一个 StreamId。

V2TXLIVE_ERROR_INVALID_LICENSE：licence 不合法，播放失败。

stopPlay

stopPlay

停止播放音视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

isPlaying

isPlaying

播放器是否正在播放中

返回值说明：

是否正在播放。

1: 正在播放中。

0: 已经停止播放。

pauseAudio

pauseAudio

暂停播放器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeAudio

resumeAudio

恢复播放器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseVideo

pauseVideo

暂停播放器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeVideo

resumeVideo

恢复播放器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setPlayoutVolume:

setPlayoutVolume:

- (V2TXLiveCode)setPlayoutVolume:	(NSInteger)volume
-----------------------------------	-------------------

设置播放器音量

参数	描述
volume	音量大小，取值范围0 - 100。【默认值】：100。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setCacheParams:maxTime:**setCacheParams:maxTime:**

- (V2TXLiveCode)setCacheParams:	(CGFloat)minTime
maxTime:	(CGFloat)maxTime

设置播放器缓存自动调整的最小和最大时间（单位：秒）

参数	描述
maxTime	缓存自动调整的最大时间，取值需要大于0。【默认值】：5。
minTime	缓存自动调整的最小时间，取值需要大于0。【默认值】：1。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，minTime 和 maxTime 需要大于0。

V2TXLIVE_ERROR_REFUSED: 播放器处于播放状态，不支持修改缓存策略。

switchStream:**switchStream:**

--	--

- (V2TXLiveCode)switchStream:	(NSString*)newUrl
-------------------------------	-------------------

直播流无缝切换，支持 FLV 和 LEB

参数	描述
newUrl	新的拉流地址。

getStreamList

getStreamList

获取码流信息

enableVolumeEvaluation:

enableVolumeEvaluation:

- (V2TXLiveCode)enableVolumeEvaluation:	(NSInteger)intervalMs
---	-----------------------

启用播放音量大小提示

开启后可以在 [onPlayoutVolumeUpdate](#) 回调中获取到 SDK 对音量大小值的评估。

参数	描述
intervalMs	决定了 onPlayoutVolumeUpdate 回调的触发间隔，单位为ms，最小间隔为100ms，如果小于等于0则会关闭回调，建议设置为300ms；【默认值】：0，不开启。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

snapshot

snapshot

截取播放过程中的视频画面

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 播放器处于停止状态，不允许调用截图操作。

enableObserveVideoFrame:pixelFormat:bufferType:

enableObserveVideoFrame:pixelFormat:bufferType:

- (V2TXLiveCode)enableObserveVideoFrame:	(BOOL)enable
pixelFormat:	(V2TXLivePixelFormat)pixelFormat
bufferType:	(V2TXLiveBufferType)bufferType

开启/关闭对视频帧的监听回调

SDK 在您开启此开关后将不再渲染视频画面，您可以通过 [V2TXLivePlayerObserver](#) 获得视频帧，并执行自定义的渲染逻辑。

参数	描述
bufferType	自定义渲染回调的视频数据格式 V2TXLiveBufferType 。
enable	是否开启自定义渲染。【默认值】：NO。
pixelFormat	自定义渲染回调的视频像素格式 V2TXLivePixelFormat 。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_NOT_SUPPORTED: 像素格式或者数据格式不支持。

enableObserveAudioFrame:

enableObserveAudioFrame:

- (V2TXLiveCode)enableObserveAudioFrame:	(BOOL)enable
--	--------------

开启/关闭对音频数据的监听回调

如果您开启此开关，您可以通过 [V2TXLivePlayerObserver](#) 获得音频数据，并执行自定义的逻辑。

参数	描述
----	----

enable	是否开启音频数据回调。【默认值】：NO。
--------	----------------------

返回值说明：

返回值 [V2TXLiveCode](#)

V2TXLIVE_OK: 成功

enableReceiveSeiMessage:payloadType:

enableReceiveSeiMessage:payloadType:

- (V2TXLiveCode)enableReceiveSeiMessage:	(BOOL)enable
payloadType:	(int)payloadType

开启接收 SEI 消息

参数	描述
enable	YES: 开启接收 SEI 消息; NO: 关闭接收 SEI 消息。【默认值】：NO。
payloadType	指定接收 SEI 消息的 payloadType，支持 5、242、243，请与发送端的 payloadType 保持一致。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enablePictureInPicture:

enablePictureInPicture:

- (V2TXLiveCode)enablePictureInPicture:	(BOOL)enable
---	--------------

开启画中画功能，仅支持直播和快直播播放

参数	描述
enable	YES: 开启画中画功能; NO: 关闭画中画功能。【默认值】：NO。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

showDebugView:

showDebugView:

- (void)showDebugView:	(BOOL)isShow
------------------------	--------------

是否显示播放器状态信息的调试浮层

参数	描述
isShow	是否显示。【默认值】：NO。

setProperty:value:

setProperty:value:

- (V2TXLiveCode)setProperty:	(NSString *)key
value:	(NSObject *)value

调用 V2TXLivePlayer 的高级 API 接口

参数	描述
key	高级 API 对应的 key, 详情请参考 V2TXLiveProperty 定义。
value	调用 key 所对应的高级 API 时，需要的参数。

注意

该接口用于调用一些高级功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，key 不允许为 nil。

startLocalRecording:

startLocalRecording:

- (V2TXLiveCode)startLocalRecording:	(V2TXLiveLocalRecordingParams *)params
--------------------------------------	--

开始录制音视频流

注意

拉流开启后才能开始录制，非拉流状态下开启录制无效。

录制过程中不要动态切换软/硬解，生成的视频极有可能出现异常。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK : 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER : 参数不合法，比如filePath 为空。

V2TXLIVE_ERROR_REFUSED : API被拒绝，拉流尚未开始。

stopLocalRecording

stopLocalRecording

停止录制音视频流

注意

当停止拉流后，如果视频还在录制中，SDK 内部会自动结束录制。

V2TXLivePlayerObserver

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePlayerObserver @ TXLiteAVSDK

Function: 腾讯云直播的播放器回调通知

功能

腾讯云直播的播放器回调通知。

介绍

可以接收 [V2TXLivePlayer](#) 播放器的一些回调通知，包括播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等。

V2TXLivePlayerObserver

V2TXLivePlayerObserver

函数列表	描述
onError:code:message:extraInfo:	直播播放器错误通知，播放器出现错误时，会回调该通知
onWarning:code:message:extraInfo:	直播播放器警告通知
onVideoResolutionChanged:width:height:	直播播放器分辨率变化通知
onConnected:extraInfo:	已经成功连接到服务器
onVideoPlaying:firstPlay:extraInfo:	视频播放事件
onAudioPlaying:firstPlay:extraInfo:	音频播放事件

<code>onVideoLoading:extraInfo:</code>	视频加载事件
<code>onAudioLoading:extraInfo:</code>	音频加载事件
<code>onPlayoutVolumeUpdate:volume:</code>	播放器音量大小回调
<code>onStatisticsUpdate:statistics:</code>	直播播放器统计数据回调
<code>onSnapshotComplete:image:</code>	截图回调
<code>onRenderVideoFrame:frame:</code>	自定义视频渲染回调
<code>onPlayoutAudioFrame:frame:</code>	音频数据回调
<code>onReceiveSeiMessage:payloadType:data:</code>	收到 SEI 消息的回调，发送端通过 <code>V2TXLivePusher</code> 中的 <code>sendSeiMessage</code> 来发送 SEI 消息
<code>onStreamSwitched:url:code:</code>	分辨率无缝切换回调
<code>onPictureInPictureStateUpdate:state:message:extraInfo:</code>	画中画状态变更回调
<code>onLocalRecordBegin:errCode:storagePath:</code>	录制任务开始的事件回调
<code>onLocalRecording:durationMs:storagePath:</code>	录制任务正在进行的进展事件回调
<code>onLocalRecordComplete:errCode:storagePath:</code>	录制任务已经结束的事件回调

`onError:code:message:extraInfo:`

`onError:code:message:extraInfo:`

<code>-(void)onError:</code>	(id< <code>V2TXLivePlayer</code> >)player
<code>code:</code>	(<code>V2TXLiveCode</code>)code
<code>message:</code>	(NSString *)msg
<code>extraInfo:</code>	(NSDictionary *)extraInfo

直播播放器错误通知，播放器出现错误时，会回调该通知

参数	描述
<code>code</code>	错误码 <code>V2TXLiveCode</code> 。
<code>extraInfo</code>	扩展信息。

msg	错误信息。
player	回调该通知的播放器对象。

onWarning:code:message:extraInfo:

onWarning:code:message:extraInfo:

- (void)onWarning:	(id<V2TXLivePlayer>)player
code:	(V2TXLiveCode)code
message:	(NSString *)msg
extraInfo:	(NSDictionary *)extraInfo

直播播放器警告通知

参数	描述
code	警告码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	警告信息。
player	回调该通知的播放器对象。

onVideoResolutionChanged:width:height:

onVideoResolutionChanged:width:height:

- (void)onVideoResolutionChanged:	(id<V2TXLivePlayer>)player
width:	(NSInteger)width
height:	(NSInteger)height

直播播放器分辨率变化通知

参数	描述
height	视频高。

player	回调该通知的播放器对象。
width	视频宽。

onConnected:extraInfo:

onConnected:extraInfo:

- (void)onConnected:	(id< V2TXLivePlayer >)player
extraInfo:	(NSDictionary *)extraInfo

已经成功连接到服务器

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。

onVideoPlaying:firstPlay:extraInfo:

onVideoPlaying:firstPlay:extraInfo:

- (void)onVideoPlaying:	(id< V2TXLivePlayer >)player
firstPlay:	(BOOL)firstPlay
extraInfo:	(NSDictionary *)extraInfo

视频播放事件

参数	描述
extraInfo	扩展信息。
firstPlay	第一次播放标志。
player	回调该通知的播放器对象。

onAudioPlaying:firstPlay:extraInfo:

onAudioPlaying:firstPlay:extraInfo:

- (void)onAudioPlaying:	(id< V2TXLivePlayer >)player
firstPlay:	(BOOL)firstPlay
extraInfo:	(NSDictionary *)extraInfo

音频播放事件

参数	描述
extraInfo	扩展信息。
firstPlay	第一次播放标志。
player	回调该通知的播放器对象。

onVideoLoading:extraInfo:**onVideoLoading:extraInfo:**

- (void)onVideoLoading:	(id< V2TXLivePlayer >)player
extraInfo:	(NSDictionary *)extraInfo

视频加载事件

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。

onAudioLoading:extraInfo:**onAudioLoading:extraInfo:**

- (void)onAudioLoading:	(id< V2TXLivePlayer >)player
extraInfo:	(NSDictionary *)extraInfo

音频加载事件

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。

onPlayoutVolumeUpdate:volume:

onPlayoutVolumeUpdate:volume:

- (void)onPlayoutVolumeUpdate:	(id<V2TXLivePlayer>)player
volume:	(NSInteger)volume

播放器音量大小回调

参数	描述
player	回调该通知的播放器对象。
volume	音量大小。

注意

调用 [enableVolumeEvaluation](#) 开启播放音量大小提示之后，会收到这个回调通知。

onStatisticsUpdate:statistics:

onStatisticsUpdate:statistics:

- (void)onStatisticsUpdate:	(id<V2TXLivePlayer>)player
statistics:	(V2TXLivePlayerStatistics *)statistics

直播播放器统计数据回调

参数	描述
player	回调该通知的播放器对象。
statistics	播放器统计数据 V2TXLivePlayerStatistics 。

onSnapshotComplete:image:

onSnapshotComplete:image:

- (void)onSnapshotComplete:	(id< V2TXLivePlayer >)player
image:	(nullable TXImage *)image

截图回调

参数	描述
image	已截取的视频画面。
player	回调该通知的播放器对象。

注意

调用 [snapshot](#) 截图之后，会收到这个回调通知。

onRenderVideoFrame:frame:

onRenderVideoFrame:frame:

- (void)onRenderVideoFrame:	(id< V2TXLivePlayer >)player
frame:	(V2TXLiveVideoFrame *)videoFrame

自定义视频渲染回调

参数	描述
player	回调该通知的播放器对象。
videoFrame	视频帧数据 V2TXLiveVideoFrame 。

注意

需要您调用 [enableObserveVideoFrame](#) 开启回调开关。

onPlayoutAudioFrame:frame:

onPlayoutAudioFrame:frame:

- (void)onPlayoutAudioFrame:	(id<V2TXLivePlayer>)player
frame:	(V2TXLiveAudioFrame *)audioFrame

音频数据回调

参数	描述
audioFrame	音频帧数据 V2TXLiveAudioFrame 。
player	回调该通知的播放器对象。

注意

需要您调用 [enableObserveAudioFrame](#) 开启回调开关。请在当前回调中使用 audioFrame 的 data。

onReceiveSeiMessage:payloadType:data:

onReceiveSeiMessage:payloadType:data:

- (void)onReceiveSeiMessage:	(id<V2TXLivePlayer>)player
payloadType:	(int)payloadType
data:	(NSData *)data

收到 SEI 消息的回调，发送端通过 {@link V2TXLivePusher} 中的 `sendSeiMessage` 来发送 SEI 消息

参数	描述
data	数据。
payloadType	回调数据的SEI payloadType。
player	回调该通知的播放器对象。

注意

调用 [V2TXLivePlayer](#) 中的 `enableReceiveSeiMessage` 开启接收 SEI 消息之后，会收到这个回调通知。

onStreamSwitched:url:code:

onStreamSwitched:url:code:

--	--

- (void)onStreamSwitched:	(id<V2TXLivePlayer>)player
url:	(NSString *)url
code:	(NSInteger)code

分辨率无缝切换回调

参数	描述
code	状态码，0：成功，-1：切换超时，-2：切换失败，服务端错误，-3：切换失败，客户端错误。
player	回调该通知的播放器对象。
url	切换的播放地址。

注意

调用 [V2TXLivePlayer](#) 中的 `switchStream` 切换分辨率，会收到这个回调通知。

onPictureInPictureStateUpdate:state:message:extraInfo:

onPictureInPictureStateUpdate:state:message:extraInfo:

- (void)onPictureInPictureStateUpdate:	(id<V2TXLivePlayer>)player
state:	(V2TXLivePictureInPictureState)state
message:	(NSString *)msg
extraInfo:	(NSDictionary *)extraInfo

画中画状态变更回调

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。
state	画中画的状态。

注意

调用 [V2TXLivePlayer](#) 中的 `enablePictureInPicture` 开启画中画之后，会收到这个回调通知。

onLocalRecordBegin:errCode:storagePath:

onLocalRecordBegin:errCode:storagePath:

- (void)onLocalRecordBegin:	(id< V2TXLivePlayer >)player
errCode:	(NSInteger)errCode
storagePath:	(NSString *)storagePath

录制任务开始的事件回调

参数	描述
code	状态码。 0：录制任务启动成功。 -1：内部错误导致录制任务启动失败。 -2：文件后缀名有误（比如不支持的录制格式）。 -6：录制已经启动，需要先停止录制。 -7：录制文件已存在，需要先删除文件。 -8：录制目录无写入权限，请检查目录权限问题。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

onLocalRecording:durationMs:storagePath:

onLocalRecording:durationMs:storagePath:

- (void)onLocalRecording:	(id< V2TXLivePlayer >)player
durationMs:	(NSInteger)durationMs
storagePath:	(NSString *)storagePath

录制任务正在进行的进展事件回调

参数	描述
durationMs	录制时长。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

onLocalRecordComplete:errCode:storagePath:

onLocalRecordComplete:errCode:storagePath:

- (void)onLocalRecordComplete:	(id<V2TXLivePlayer>)player
errCode:	(NSInteger)errCode
storagePath:	(NSString *)storagePath

录制任务已经结束的事件回调

参数	描述
code	状态码。 0：结束录制任务成功。 -1：录制失败。 -2：切换分辨率或横竖屏导致录制结束。 -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

V2TXLivePremier

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePremier @ TXLiteAVSDK

Function: V2TXLive 高级接口

V2TXLivePremier

V2TXLivePremier

函数列表	描述
getSDKVersionStr	获取 SDK 版本号
setObserver:	设置 V2TXLivePremier 回调接口
setLogConfig:	设置 Log 的配置信息
setEnvironment:	设置 SDK 接入环境
setLicence:key:	设置 SDK 的授权 License
setSocks5Proxy:port:username:password:config:	设置 SDK socks5 代理配置
enableAudioCaptureObserver:format:	开启/关闭对音频采集数据的监听回调（可读写）
enableAudioPlayoutObserver:format:	开启/关闭对最终系统要播放出的音频数据的监听回调
enableVoiceEarMonitorObserver:	开启/关闭耳返音频数据的监听回调
setUserId:	设置 userId
callExperimentalAPI:	调用实验性 API 接口

V2TXLivePremierObserver

--	--

函数列表	描述
onLog:log:	自定义 Log 输出回调接口
onLicenceLoaded:Reason:	setLicence 接口回调
onCaptureAudioFrame:	本地麦克风采集到的音频数据回调
onPlayoutAudioFrame:	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onVoiceEarMonitorAudioFrame:	耳返的音频数据

getSDKVersionStr

getSDKVersionStr

获取 SDK 版本号

setObserver:

setObserver:

+ (void)setObserver:	(id< V2TXLivePremierObserver >)observer
----------------------	---

设置 **V2TXLivePremier** 回调接口

setLogConfig:

setLogConfig:

+ (V2TXLiveCode)setLogConfig:	(V2TXLiveLogConfig *)config
-------------------------------	--

设置 Log 的配置信息

setEnvironment:

setEnvironment:

+ (V2TXLiveCode)setEnvironment:	(const char *)env
---------------------------------	-------------------

设置 SDK 接入环境

参数	描述
env	目前支持“default”和“GDPR”两个参数。 default：默认环境，SDK 会在全球寻找最佳接入点进行接入。 GDPR：所有音视频数据和质量统计数据都不会经过中国大陆地区的服务器。

注意

如您的应用无特殊需求，请不要调用此接口进行设置。

setLicence:key:

setLicence:key:

+ (void)setLicence:	(NSString *)url
key:	(NSString *)key

设置 SDK 的授权 License

文档地址：<https://cloud.tencent.com/document/product/454/34750>。

参数	描述
key	licence的密钥。
url	licence的地址。

setSocks5Proxy:port:username:password:config:

setSocks5Proxy:port:username:password:config:

+ (V2TXLiveCode)setSocks5Proxy:	(NSString *)host
port:	(NSInteger)port
username:	(NSString *)username
password:	(NSString *)password
config:	(V2TXLiveSocks5ProxyConfig *)config

设置 SDK socks5 代理配置

参数	描述
config	配置使用 socks5 代理服务器的协议。
host	socks5 代理服务器的地址。
password	socks5 代理服务器的验证的密码。
port	socks5 代理服务器的端口。
username	socks5 代理服务器的验证的用户名。

enableAudioCaptureObserver:format:

enableAudioCaptureObserver:format:

+ (V2TXLiveCode)enableAudioCaptureObserver:	(BOOL)enable
format:	(V2TXLiveAudioFrameObserverFormat *)format

开启/关闭对音频采集数据的监听回调（可读写）

参数	描述
enable	是否开启。【默认值】：false。
format	设置回调出的 AudioFrame 的格式。

注意

需要在 [startPush](#) 之前调用，才会生效。

enableAudioPlayoutObserver:format:

enableAudioPlayoutObserver:format:

+ (V2TXLiveCode)enableAudioPlayoutObserver:	(BOOL)enable
format:	(V2TXLiveAudioFrameObserverFormat *)format

开启/关闭对最终系统要播放出的音频数据的监听回调

--	--

参数	描述
enable	是否开启。【默认值】：false。
format	设置回调出的 AudioFrame 的格式。

enableVoiceEarMonitorObserver:

enableVoiceEarMonitorObserver:

+ (V2TXLiveCode)enableVoiceEarMonitorObserver:	(BOOL)enable
--	--------------

开启/关闭耳返音频数据的监听回调

参数	描述
enable	是否开启。【默认值】：false。

setUserId:

setUserId:

+ (void)setUserId:	(NSString *)userId
--------------------	--------------------

设置 userId

参数	描述
userId	业务侧自身维护的用户/设备id。

callExperimentalAPI:

callExperimentalAPI:

+ (V2TXLiveCode)callExperimentalAPI:	(NSString *)jsonStr
--------------------------------------	---------------------

调用实验性 API 接口

参数	描述
jsonStr	接口及参数描述的 JSON 字符串。

注意

该接口用于调用一些实验性功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，参数非法。

onLog:log:

onLog:log:

- (void)onLog:	(V2TXLiveLogLevel)level
log:	(NSString *)log

自定义 **Log** 输出回调接口

onLicenceLoaded:Reason:

onLicenceLoaded:Reason:

- (void)onLicenceLoaded:	(int)result
Reason:	(NSString *)reason

setLicence 接口回调

参数	描述
reason	设置 licence 失败原因。
result	设置 licence 结果 0 成功，负数失败。

onCaptureAudioFrame:

onCaptureAudioFrame:

- (void) onCaptureAudioFrame:	(V2TXLiveAudioFrame *)frame
-------------------------------	--

本地麦克风采集到的音频数据回调

参数	描述
frame	音频数据。

注意

请不要在此回调函数中做任何耗时操作，建议直接拷贝到另一线程进行处理，否则会导致各种声音问题。

此接口回调出的音频数据支持修改。

此接口回调出的音频时间帧长固定为0.02s。

由时间帧长转化为字节帧长的公式为【采样率 × 时间帧长 × 声道数 × 采样点位宽】。

以SDK默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为【 $48000 \times 0.02s \times 1 \times 16bit = 15360bit = 1920$ 字节】。

此接口回调出的音频数据**不包含**背景音、音效、混响等前处理效果，延迟极低。

需要您调用 [enableAudioCaptureObserver](#) 开启回调开关。

onPlayoutAudioFrame:

onPlayoutAudioFrame:

- (void) onPlayoutAudioFrame:	(V2TXLiveAudioFrame *)frame
-------------------------------	-----------------------------

将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把各路待播放的音频混合之后的音频数据，在提交系统播放之前，以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长固定为 0.02s，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 采样率 × 时间帧长 × 声道数 × 采样点位宽 。

以 SDK 默认的音频录制格式 48000 采样率、单声道、16 采样点位宽为例，字节帧长为 $48000 \times 0.02s \times 1 \times 16bit = 15360bit = 1920$ 字节 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

3. 此接口回调出的是对各路待播放音频数据的混合，但其中并不包含耳返的音频数据。

onVoiceEarMonitorAudioFrame:

onVoiceEarMonitorAudioFrame:

- (void) onVoiceEarMonitorAudioFrame:	(V2TXLiveAudioFrame *)frame
---------------------------------------	-----------------------------

耳返的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把耳返的音频数据在播放之前以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长不固定，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 采样率 × 时间帧长 × 声道数 × 采样点位宽 。

以 TRTC 默认的音频录制格式 48000 采样率、单声道、16采样点位宽为例，0.02s 的音频数据字节帧长为 $48000 \times 0.02s \times 1 \times 16bit = 15360bit = 1920字节$ 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，否则会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

TXAudioEffectManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 背景音乐、短音效和人声特效的管理类

Function: 用于对背景音乐、短音效和人声特效进行设置的管理类

TXAudioEffectManager

TXAudioEffectManager

函数列表	描述
enableVoiceEarMonitor:	开启耳返
setVoiceEarMonitorVolume:	设置耳返音量
setVoiceReverbType:	设置人声的混响效果
setVoiceChangerType:	设置人声的变声特效
setVoiceVolume:	设置语音音量
setVoicePitch:	设置语音音调
startPlayMusic:onStart:onProgress:onComplete:	开始播放背景音乐
stopPlayMusic:	停止播放背景音乐
pausePlayMusic:	暂停播放背景音乐
resumePlayMusic:	恢复播放背景音乐
setAllMusicVolume:	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume:volume:	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume:volume:	设置某一首背景音乐的本地音量的大小
setMusicPitch:pitch:	调整背景音乐的音调高低

<code>setMusicSpeedRate:speedRate:</code>	调整背景音乐的变速效果
<code>getMusicCurrentPosInMS:</code>	获取背景音乐的播放进度（单位：毫秒）
<code>getMusicDurationInMS:</code>	获取背景音乐的总时长（单位：毫秒）
<code>seekMusicToPosInMS:pts:</code>	设置背景音乐的播放进度（单位：毫秒）
<code>setMusicScratchSpeedRate:speedRate:</code>	调整搓碟的变速效果
<code>preloadMusic:onProgress:onError:</code>	预加载背景音乐
<code>getMusicTrackCount:</code>	获取背景音乐的音轨数量
<code>setMusicTrack:track:</code>	指定背景音乐的播放音轨

结构体类型

函数列表	描述
<code>TXAudioMusicParam</code>	背景音乐的播放控制信息

枚举类型

枚举类型	描述
<code>TXVoiceReverbType</code>	混响特效
<code>TXVoiceChangeType</code>	变声特效

enableVoiceEarMonitor:

enableVoiceEarMonitor:

- (void)enableVoiceEarMonitor:	(BOOL)enable
--------------------------------	--------------

开启耳返

主播开启耳返后，可以在耳机里听到麦克风采集到的自己发出的声音，该特效适用于主播唱歌的应用场景中。

需要您注意的是，由于蓝牙耳机的硬件延迟非常高，所以在主播佩戴蓝牙耳机时无法开启此特效，请尽量在用户界面上提示主播佩戴有线耳机。

同时也需要注意，并非所有的手机开启此特效后都能达到优秀的耳返效果，我们已经对部分耳返效果不佳的手机屏蔽了该特效。

参数	描述
enable	YES：开启；NO：关闭。

注意

仅在主播佩戴耳机时才能开启此特效，同时请您提示主播佩戴有线耳机。

setVoiceEarMonitorVolume:

setVoiceEarMonitorVolume:

- (void)setVoiceEarMonitorVolume:	(NSInteger)volume
-----------------------------------	-------------------

设置耳返音量

通过该接口您可以设置耳返特效中声音的音量大小。

参数	描述
volume	音量大小，取值范围为 0 - 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoiceReverbType:

setVoiceReverbType:

- (void)setVoiceReverbType:	(TXVoiceReverbType)reverbType
-----------------------------	-------------------------------

设置人声的混响效果

通过该接口您可以设置人声的混响效果，具体特效请参见枚举定义 [TXVoiceReverbType](#)。

注意

设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceChangerType:

setVoiceChangerType:

- (void)setVoiceChangerType:	(TXVoiceChangeType)changerType
------------------------------	--------------------------------

设置人声的变声特效

通过该接口您可以设置人声的变声特效，具体特效请参见枚举定义 [TXVoiceChangeType](#)。

注意

设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceVolume:

setVoiceVolume:

- (void)setVoiceVolume:	(NSInteger)volume
-------------------------	-------------------

设置语音音量

该接口可以设置语音音量的大小，一般配合音乐音量的设置接口 [setAllMusicVolume](#) 协同使用，用于调谐语音和音乐在混音前各自的音量占比。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoicePitch:

setVoicePitch:

-(void)setVoicePitch:	(double)pitch
-----------------------	---------------

设置语音音调

该接口可以设置语音音调，用于实现变调不变速的目的。

参数	描述
----	----

pitch

音调，取值范围为-1.0f~1.0f，默认值：0.0f。

startPlayMusic:onStart:onProgress:onComplete:

startPlayMusic:onStart:onProgress:onComplete:

- (void)startPlayMusic:	(TXAudioMusicParam *)musicParam
onStart:	(TXAudioMusicStartBlock _Nullable)startBlock
onProgress:	(TXAudioMusicProgressBlock _Nullable)progressBlock
onComplete:	(TXAudioMusicCompleteBlock _Nullable)completeBlock

开始播放背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
completeBlock	播放结束回调。
musicParam	音乐参数。
progressBlock	播放进度回调。
startBlock	播放开始回调。

注意

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

stopPlayMusic:

stopPlayMusic:

- (void)stopPlayMusic:	(int32_t)id
------------------------	-------------

停止播放背景音乐

参数	描述
----	----

id	音乐 ID。
----	--------

pausePlayMusic:

pausePlayMusic:

- (void)pausePlayMusic:	(int32_t)id
-------------------------	-------------

暂停播放背景音乐

参数	描述
id	音乐 ID。

resumePlayMusic:

resumePlayMusic:

- (void)resumePlayMusic:	(int32_t)id
--------------------------	-------------

恢复播放背景音乐

参数	描述
id	音乐 ID。

setAllMusicVolume:

setAllMusicVolume:

- (void)setAllMusicVolume:	(NSInteger)volume
----------------------------	-------------------

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

本地音量：即主播本地可以听到的背景音乐的音量大小。

远端音量：即观众端可以听到的背景音乐的音量大小。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：60。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPublishVolume:volume:

setMusicPublishVolume:volume:

- (void)setMusicPublishVolume:	(int32_t)id
volume:	(NSInteger)volume

设置某一首背景音乐的远端音量的大小

该接口可以细粒度地控制每一首背景音乐的远端音量，也就是观众端可听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100；默认值：60。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPlayoutVolume:volume:

setMusicPlayoutVolume:volume:

- (void)setMusicPlayoutVolume:	(int32_t)id
volume:	(NSInteger)volume

设置某一首背景音乐的本地音量的大小

该接口可以细粒度地控制每一首背景音乐的本地音量，也就是主播本地可以听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。

volume	音量大小，取值范围为0 - 100，默认值：60。
--------	---------------------------

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPitch:pitch:

setMusicPitch:pitch:

- (void)setMusicPitch:	(int32_t)id
pitch:	(double)pitch

调整背景音乐的音调高低

参数	描述
id	音乐 ID。
pitch	音调，默认值是0.0f，范围是：[-1 ~ 1] 之间的浮点数。

setMusicSpeedRate:speedRate:

setMusicSpeedRate:speedRate:

- (void)setMusicSpeedRate:	(int32_t)id
speedRate:	(double)speedRate

调整背景音乐的变速效果

参数	描述
id	音乐 ID。
speedRate	速度，默认值是1.0f，范围是：[0.5 ~ 2] 之间的浮点数。

getMusicCurrentPosInMS:

getMusicCurrentPosInMS:

- (NSInteger)getMusicCurrentPosInMS:	(int32_t)id
--------------------------------------	-------------

获取背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。

返回值说明：

成功返回当前播放时间，单位：毫秒，失败返回 -1。

getMusicDurationInMS:

getMusicDurationInMS:

- (NSInteger)getMusicDurationInMS:	(NSString *)path
------------------------------------	------------------

获取背景音乐的总时长（单位：毫秒）

参数	描述
path	音乐文件路径。

返回值说明：

成功返回时长，失败返回 -1。

seekMusicToPosInMS:pts:

seekMusicToPosInMS:pts:

- (void)seekMusicToPosInMS:	(int32_t)id
pts:	(NSInteger)pts

设置背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。

pts	单位: 毫秒。
-----	---------

注意

请尽量避免过度频繁地调用该接口，因为该接口可能会再次读写音乐文件，耗时稍高。

因此，当用户拖拽音乐的播放进度条时，请在用户完成拖拽操作后再调用本接口。

因为 UI 上的进度条控件往往会以很高的频率反馈用户的拖拽进度，如不做频率限制，会导致较差的用户体验。

setMusicScratchSpeedRate:speedRate:

setMusicScratchSpeedRate:speedRate:

- (void)setMusicScratchSpeedRate:	(int32_t)id
speedRate:	(double)scratchSpeedRate

调整搓碟的变速效果

参数	描述
id	音乐 ID。
scratchSpeedRate	搓碟速度，默认值是1.0f，范围是：[-12.0 ~ 12.0] 之间的浮点数, 速度值正/负表示方向正/反，绝对值大小表示速度快慢。

注意

前置条件 preloadMusic 成功。

preloadMusic:onProgress:onError:

preloadMusic:onProgress:onError:

- (void)preloadMusic:	(TXAudioMusicParam *)preloadParam
onProgress:	(TXMusicPreloadProgressBlock _Nullable)progressBlock
onError:	(TXMusicPreloadErrorBlock _Nullable)errorBlock

预加载背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

--	--

参数	描述
preloadParam	预加载音乐参数。

注意

1. 预先加载最多同时支持2个不同 ID 的预加载，且预加载时长不超过10分钟，使用完需 `stopPlayMusic`，否则内存不释放。
2. 若该ID对应的音乐正在播放中，预加载会失败，需先调用 `stopPlayMusic`。
3. 当 `musicParam` 和传入 `startPlayMusic` 的 `musicParam` 完全相同时，预加载有效。

getMusicTrackCount:

getMusicTrackCount:

- (NSInteger)getMusicTrackCount:	(int32_t)id
----------------------------------	-------------

获取背景音乐的音轨数量

参数	描述
id	音乐 ID。

setMusicTrack:track:

setMusicTrack:track:

- (void)setMusicTrack:	(int32_t)id
track:	(NSInteger)track

指定背景音乐的播放音轨

参数	描述
id	音乐 ID。
index	默认播放第一个音轨。取值范围[0, 音轨总数)。

注意

音轨总数量可通过 `getMusicTrackCount` 接口获取。

TXVoiceReverbType

TXVoiceReverbType

混响特效

混响特效可以作用于人声之上，通过声学算法对声音进行叠加处理，模拟出各种不同环境下的临场感受，目前支持如下几种混响效果：

0：关闭；1：KTV；2：小房间；3：大会堂；4：低沉；5：洪亮；6：金属声；7：磁性；8：空灵；9：录音棚；10：悠扬 11：录音棚2。

枚举	取值	描述
TXVoiceReverbType_0	0	关闭特效
TXVoiceReverbType_1	1	KTV
TXVoiceReverbType_2	2	小房间
TXVoiceReverbType_3	3	大会堂
TXVoiceReverbType_4	4	低沉
TXVoiceReverbType_5	5	洪亮
TXVoiceReverbType_6	6	金属声
TXVoiceReverbType_7	7	磁性
TXVoiceReverbType_8	8	空灵
TXVoiceReverbType_9	9	录音棚
TXVoiceReverbType_10	10	悠扬
TXVoiceReverbType_11	11	录音棚2

TXVoiceChangeType

TXVoiceChangeType

变声特效

变声特效可以作用于人声之上，通过声学算法对人声进行二次处理，以获得与原始声音所不同的音色，目前支持如下几种变声特效：

0：关闭；1：熊孩子；2：萝莉；3：大叔；4：重金属；5：感冒；6：外语腔；7：困兽；8：肥宅；9：强电流；10：重机械；11：空灵。

枚举	取值	描述
TXVoiceChangeType_0	0	关闭
TXVoiceChangeType_1	1	熊孩子
TXVoiceChangeType_2	2	萝莉
TXVoiceChangeType_3	3	大叔
TXVoiceChangeType_4	4	重金属
TXVoiceChangeType_5	5	感冒
TXVoiceChangeType_6	6	外语腔
TXVoiceChangeType_7	7	困兽
TXVoiceChangeType_8	8	肥宅
TXVoiceChangeType_9	9	强电流
TXVoiceChangeType_10	10	重机械
TXVoiceChangeType_11	11	空灵

TXAudioMusicParam

TXAudioMusicParam

背景音乐的播放控制信息

该信息用于在接口 [startPlayMusic](#) 中指定背景音乐的相关信息，包括播放 ID、文件路径和循环次数等：

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

枚举类型	描述
ID	【字段含义】 音乐 ID。 【特殊说明】 SDK 允许播放多路音乐，因此需要使用 ID 进行标记，用于控制音乐的开始、停止、音量等。
endTimeMS	【字段含义】 音乐结束播放时间点，单位毫秒，0表示播放至文件结尾。

isShortFile	【字段含义】播放的是否为短音乐文件。 【推荐取值】YES：需要重复播放的短音乐文件；NO：正常的音乐文件。默认值：NO。
loopCount	【字段含义】音乐循环播放的次数。 【推荐取值】取值范围为0 - 任意正整数，默认值：0。0 表示播放音乐一次；1 表示播放音乐两次；以此类推。
path	【字段含义】音效文件的完整路径或 URL 地址。支持的音频格式包括 MP3、AAC、M4A、WAV。
publish	【字段含义】是否将音乐传到远端。 【推荐取值】YES：音乐在本地播放的同时，远端用户也能听到该音乐；NO：主播只能在本地听到该音乐，远端观众听不到。默认值：NO。
startTimeMS	【字段含义】音乐开始播放时间点，单位：毫秒。

TXBeautyManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 美颜与图像处理参数设置类

Function: 修改美颜、滤镜、绿幕等参数

TXBeautyManager

TXBeautyManager

函数列表	描述
setBeautyStyle:	设置美颜（磨皮）算法
setBeautyLevel:	设置美颜级别
setWhitenessLevel:	设置美白级别
enableSharpnessEnhancement:	开启清晰度增强
setRuddyLevel:	设置红润级别
setFilter:	设置色彩滤镜效果
setFilterStrength:	设置色彩滤镜的强度
setGreenScreenFile:	设置绿幕背景视频
setEyeScaleLevel:	设置大眼级别
setFaceSlimLevel:	设置瘦脸级别
setFaceVLevel:	设置 V 脸级别
setChinLevel:	设置下巴拉伸或收缩
setFaceShortLevel:	设置短脸级别
setFaceNarrowLevel:	设置窄脸级别

<code>setNoseSlimLevel:</code>	设置瘦鼻级别
<code>setEyeLightenLevel:</code>	设置亮眼级别
<code>setToothWhitenLevel:</code>	设置牙齿美白级别
<code>setWrinkleRemoveLevel:</code>	设置祛皱级别
<code>setPouchRemoveLevel:</code>	设置祛眼袋级别
<code>setSmileLinesRemoveLevel:</code>	设置法令纹去除级别
<code>setForeheadLevel:</code>	设置发际线调整级别
<code>setEyeDistanceLevel:</code>	设置眼距
<code>setEyeAngleLevel:</code>	设置眼角调整级别
<code>setMouthShapeLevel:</code>	设置嘴型调整级别
<code>setNoseWingLevel:</code>	设置鼻翼调整级别
<code>setNosePositionLevel:</code>	设置鼻子位置
<code>setLipsThicknessLevel:</code>	设置嘴唇厚度
<code>setFaceBeautyLevel:</code>	设置脸型
<code>setMotionTpl:inDir:</code>	选择 AI 动效挂件
<code>setMotionMute:</code>	是否在动效素材播放时静音

枚举类型

枚举类型	描述
<code>TXBeautyStyle</code>	美颜（磨皮）算法

setBeautyStyle:

setBeautyStyle:

- (void)setBeautyStyle:	(<code>TXBeautyStyle</code>)beautyStyle
-------------------------	---

设置美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案：

参数	描述
beautyStyle	美颜风格，TXBeautyStyleSmooth：光滑；TXBeautyStyleNature：自然；TXBeautyStylePitu：优图。

setBeautyLevel:

setBeautyLevel:

- (void)setBeautyLevel:	(float)beautyLevel
-------------------------	--------------------

设置美颜级别

参数	描述
beautyLevel	美颜级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

setWhitenessLevel:

setWhitenessLevel:

- (void)setWhitenessLevel:	(float)whitenessLevel
----------------------------	-----------------------

设置美白级别

参数	描述
whitenessLevel	美白级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

enableSharpnessEnhancement:

enableSharpnessEnhancement:

- (void)enableSharpnessEnhancement:	(BOOL)enable
-------------------------------------	--------------

开启清晰度增强

setRuddyLevel:

setRuddyLevel:

- (void)setRuddyLevel:	(float)ruddyLevel
------------------------	-------------------

设置红润级别

参数	描述
ruddyLevel	红润级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

setFilter:

setFilter:

- (void)setFilter:	(nullable UIImage *)image
--------------------	---------------------------

设置色彩滤镜效果

色彩滤镜，是一副包含色彩映射关系的颜色查找表图片，您可以在我们提供的官方 **Demo** 中找到预先准备好的几张滤镜图片。

SDK 会根据该查找表中的映射关系，对摄像头采集出的原始视频画面进行二次处理，以达到预期的滤镜效果。

参数	描述
image	包含色彩映射关系的颜色查找表图片，必须是 png 格式。

setFilterStrength:

setFilterStrength:

- (void)setFilterStrength:	(float)strength
----------------------------	-----------------

设置色彩滤镜的强度

该数值越高，色彩滤镜的作用强度越明显，经过滤镜处理后的视频画面跟原画面的颜色差异越大。

我默认的滤镜浓度是 0.5，如果您觉得默认的滤镜效果不明显，可以设置为 0.5 以上的数字，最大值为 1。

参数	描述
strength	从 0 到 1，数值越大滤镜效果越明显，默认值为 0.5。

setGreenScreenFile:

setGreenScreenFile:

- (int)setGreenScreenFile:	(nullable NSString *)path
----------------------------	---------------------------

设置绿幕背景视频

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。此接口所开启的绿幕功能不具备智能去除背景的能力，需要被拍摄者的背后有一块绿色的幕布来辅助产生特效。

参数	描述
path	MP4格式的视频文件路径; 设置空值表示关闭特效。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeScaleLevel:

setEyeScaleLevel:

- (int)setEyeScaleLevel:	(float)eyeScaleLevel
--------------------------	----------------------

设置大眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeScaleLevel	大眼级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceSlimLevel:

setFaceSlimLevel:

- (int)setFaceSlimLevel:	(float)faceSlimLevel
--------------------------	----------------------

设置瘦脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceSlimLevel	瘦脸级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceVLevel:

setFaceVLevel:

- (int)setFaceVLevel:	(float)faceVLevel
-----------------------	-------------------

设置 V 脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceVLevel	V 脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setChinLevel:

setChinLevel:

- (int)setChinLevel:	(float)chinLevel
----------------------	------------------

设置下巴拉伸或收缩

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
chinLevel	下巴拉伸或收缩级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示收缩，大于 0 表示拉伸。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceShortLevel:

setFaceShortLevel:

- (int)setFaceShortLevel:	(float)faceShortLevel
---------------------------	-----------------------

设置短脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceShortLevel	短脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceNarrowLevel:

setFaceNarrowLevel:

- (int)setFaceNarrowLevel:	(float)faceNarrowLevel
----------------------------	------------------------

设置窄脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
level	窄脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseSlimLevel:

setNoseSlimLevel:

- (int)setNoseSlimLevel:	(float)noseSlimLevel
--------------------------	----------------------

设置瘦鼻级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
noseSlimLevel	瘦鼻级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeLightenLevel:

setEyeLightenLevel:

- (int)setEyeLightenLevel:	(float)eyeLightenLevel
----------------------------	------------------------

设置亮眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeLightenLevel	亮眼级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setToothWhitenLevel:

setToothWhitenLevel:

- (int)setToothWhitenLevel:	(float)toothWhitenLevel
-----------------------------	-------------------------

设置牙齿美白级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
toothWhitenLevel	白牙级别，取值范围 0 - 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setWrinkleRemoveLevel:

setWrinkleRemoveLevel:

- (int)setWrinkleRemoveLevel:	(float)wrinkleRemoveLevel
-------------------------------	---------------------------

设置祛皱级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
wrinkleRemoveLevel	祛皱级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setPouchRemoveLevel:

setPouchRemoveLevel:

- (int)setPouchRemoveLevel:	(float)pouchRemoveLevel
-----------------------------	-------------------------

设置祛眼袋级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
pouchRemoveLevel	祛眼袋级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setSmileLinesRemoveLevel:

setSmileLinesRemoveLevel:

--

- (int)setSmileLinesRemoveLevel:	(float)smileLinesRemoveLevel
----------------------------------	------------------------------

设置法令纹去除级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
smileLinesRemoveLevel	法令纹级别，取值范围 0 - 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setForeheadLevel:

setForeheadLevel:

- (int)setForeheadLevel:	(float)foreheadLevel
--------------------------	----------------------

设置发际线调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
foreheadLevel	发际线级别，取值范围-9 - 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeDistanceLevel:

setEyeDistanceLevel:

- (int)setEyeDistanceLevel:	(float)eyeDistanceLevel
-----------------------------	-------------------------

设置眼距

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述

eyeDistanceLevel

眼距级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeAngleLevel:

setEyeAngleLevel:

- (int)setEyeAngleLevel:

(float)eyeAngleLevel

设置眼角调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeAngleLevel	眼角调整级别，取值范围-9 - 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMouthShapeLevel:

setMouthShapeLevel:

- (int)setMouthShapeLevel:

(float)mouthShapeLevel

设置嘴型调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
mouthShapeLevel	嘴型级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseWingLevel:

setNoseWingLevel:

- (int)setNoseWingLevel:	(float)noseWingLevel
--------------------------	----------------------

设置鼻翼调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
noseWingLevel	鼻翼调整级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNosePositionLevel:

setNosePositionLevel:

- (int)setNosePositionLevel:	(float)nosePositionLevel
------------------------------	--------------------------

设置鼻子位置

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
nosePositionLevel	鼻子位置级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示抬高，大于 0 表示降低。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setLipsThicknessLevel:

setLipsThicknessLevel:

- (int)setLipsThicknessLevel:	(float)lipsThicknessLevel
-------------------------------	---------------------------

设置嘴唇厚度

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
lipsThicknessLevel	嘴唇厚度级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceBeautyLevel:

setFaceBeautyLevel:

- (int)setFaceBeautyLevel:	(float)faceBeautyLevel
----------------------------	------------------------

设置脸型

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceBeautyLevel	美型级别，取值范围 0 - 9；0 表示关闭，1 - 9 值越大，效果越明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMotionTmpl:inDir:

setMotionTmpl:inDir:

- (void)setMotionTmpl:	(nullable NSString *)tmplName
inDir:	(nullable NSString *)tmplDir

选择 AI 动效挂件

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述

tmplDir	动效素材文件所在目录。
tmplName	动效挂件名称。

setMotionMute:

setMotionMute:

- (void)setMotionMute:	(BOOL)motionMute
------------------------	------------------

是否在动效素材播放时静音

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。
有些挂件本身会有声音特效，通过此 API 可以关闭这些特效播放时所带的声音效果。

参数	描述
motionMute	YES：静音；NO：不静音。

TXBeautyStyle

TXBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TXBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。
TXBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TXBeautyStylePitu	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TXDeviceManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 音视频设备管理模块

Function: 用于管理摄像头、麦克风和扬声器等音视频相关的硬件设备

TXDeviceManager

TXDeviceObserver

函数列表	描述
onDeviceChanged:type:state:	本地设备的通断状态发生变化（仅适用于桌面系统）

TXDeviceManager

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera:	切换前置或后置摄像头（仅适用于移动端）
isCameraZoomSupported	查询当前摄像头是否支持缩放（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio:	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus:	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition:	设置摄像头的对焦位置（仅适用于移动端）
isCameraTorchSupported	查询是否支持开启闪光灯（仅适用于移动端）
enableCameraTorch:	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动

	端)
setAudioRoute:	设置音频路由（仅适用于移动端）
setExposureCompensation:	设置摄像头的曝光参数，取值范围从-1到1
getDevicesList:	获取设备列表（仅适用于桌面端）
setCurrentDevice:deviceId:	设置当前要使用的设备（仅适用于桌面端）
getCurrentDevice:	获取当前正在使用的设备（仅适用于桌面端）
setCurrentDeviceVolume:deviceType:	设置当前设备的音量（仅适用于桌面端）
getCurrentDeviceVolume:	获取当前设备的音量（仅适用于桌面端）
setCurrentDeviceMute:deviceType:	设置当前设备的静音状态（仅适用于桌面端）
getCurrentDeviceMute:	获取当前设备的静音状态（仅适用于桌面端）
enableFollowingDefaultAudioDevice:enable:	设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）
startCameraDeviceTest:	开始摄像头测试（仅适用于桌面端）
stopCameraDeviceTest	结束摄像头测试（仅适用于桌面端）
startMicDeviceTest:	开始麦克风测试（仅适用于桌面端）
startMicDeviceTest:playback:	开始麦克风测试（仅适用于桌面端）
stopMicDeviceTest	结束麦克风测试（仅适用于桌面端）
startSpeakerDeviceTest:	开始扬声器测试（仅适用于桌面端）
stopSpeakerDeviceTest	结束扬声器测试（仅适用于桌面端）
setObserver:	设备热插拔回调（仅适用于 Mac 系统）
setCameraCapturerParam:	设置摄像头采集偏好
setSystemVolumeType:	设置系统音量类型（仅适用于移动端）

结构体类型

函数列表	描述

TXCameraCaptureParam	摄像头采集参数
TXMediaDeviceInfo	音视频设备的相关信息（仅适用于桌面平台）

枚举类型

枚举类型	描述
TXSystemVolumeType	系统音量类型（仅适用于移动设备）
TXAudioRoute	音频路由（即声音的播放模式）
TXMediaDeviceType	设备类型（仅适用于桌面平台）
TXMediaDeviceState	设备操作
TXCameraCaptureMode	摄像头采集偏好

onDeviceChanged:type:state:

onDeviceChanged:type:state:

- (void)onDeviceChanged:	(NSString*)deviceId
type:	(TXMediaDeviceType)mediaType
state:	(TXMediaDeviceState)mediaState

本地设备的通断状态发生变化（仅适用于桌面系统）

当本地设备（包括摄像头、麦克风以及扬声器）被插入或者拔出时，SDK 便会抛出此事件回调。

参数	描述
deviceId	设备 ID
state	通断状态，0：设备已添加；1：设备已被移除；2：设备已启用。
type	设备类型

isFrontCamera

isFrontCamera

判断当前是否为前置摄像头（仅适用于移动端）

switchCamera:

switchCamera:

- (NSInteger)switchCamera:	(BOOL)frontCamera
----------------------------	-------------------

切换前置或后置摄像头（仅适用于移动端）

isCameraZoomSupported

isCameraZoomSupported

查询当前摄像头是否支持缩放（仅适用于移动端）

getCameraZoomMaxRatio

getCameraZoomMaxRatio

获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio:

setCameraZoomRatio:

- (NSInteger)setCameraZoomRatio:	(CGFloat)zoomRatio
----------------------------------	--------------------

设置摄像头的缩放倍数（仅适用于移动端）

参数	描述
zoomRatio	取值范围1 - 5，取值为1表示最远视角（正常镜头），取值为5表示最近视角（放大镜头）。最大值推荐为5，若超过5，视频数据会变得模糊不清。

isAutoFocusEnabled

isAutoFocusEnabled

查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus:

enableCameraAutoFocus:

- (NSInteger)enableCameraAutoFocus:	(BOOL)enabled
-------------------------------------	---------------

开启自动对焦功能（仅适用于移动端）

开启后，SDK 会自动检测画面中的人脸位置，并将摄像头的焦点始终对焦在人脸位置上。

setCameraFocusPosition:

setCameraFocusPosition:

- (NSInteger)setCameraFocusPosition:	(CGPoint)position
--------------------------------------	-------------------

设置摄像头的对焦位置（仅适用于移动端）

您可以通过该接口实现如下交互：

1. 在本地摄像头的预览画面上，允许用户单击操作。
2. 在用户的单击位置显示一个矩形方框，以示摄像头会在此处对焦。
3. 随后将用户点击位置的坐标通过本接口传递给 SDK，之后 SDK 会操控摄像头按照用户期望的位置进行对焦。

参数	描述
position	对焦位置，请传入期望对焦点的坐标值

注意

使用该接口的前提是先通过 [enableCameraAutoFocus](#) 关闭自动对焦功能。

返回值说明：

0：操作成功；负数：操作失败。

isCameraTorchSupported

isCameraTorchSupported

查询是否支持开启闪光灯（仅适用于移动端）

enableCameraTorch:

enableCameraTorch:

- (NSInteger)enableCameraTorch:	(BOOL)enabled
---------------------------------	---------------

开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute:

setAudioRoute:

- (NSInteger)setAudioRoute:	(TXAudioRoute)route
-----------------------------	---------------------

设置音频路由（仅适用于移动端）

手机有两个音频播放设备：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。
设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。
设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

setExposureCompensation:

setExposureCompensation:

- (NSInteger)setExposureCompensation:	(CGFloat)value
---------------------------------------	----------------

设置摄像头的曝光参数，取值范围从-1到1

getDevicesList:

getDevicesList:

- (NSArray<TXMediaDeviceInfo *> * _Nullable)getDevicesList:	(TXMediaDeviceType)type
---	-------------------------

获取设备列表（仅适用于桌面端）

参数	描述
type	设备类型，指定需要获取哪种设备的列表。详见 TXMediaDeviceType 定义。

注意

使用完毕后请调用 `release` 方法释放资源，这样可以让 SDK 维护 ITXDeviceCollection 对象的生命周期。不要使用 `delete` 释放返回的 Collection 对象，`delete ITXDeviceCollection*` 指针会导致异常崩溃。

type 只支持 TXMediaDeviceTypeMic、TXMediaDeviceTypeSpeaker、TXMediaDeviceTypeCamera。

此接口只支持 Mac 和 Windows 平台。

setCurrentDevice:deviceId:

setCurrentDevice:deviceId:

- (NSInteger)setCurrentDevice:	(TXMediaDeviceType)type
deviceId:	(NSString *)deviceId

设置当前要使用的设备（仅适用于桌面端）

参数	描述
deviceId	设备ID，您可以通过接口 getDevicesList 获得设备 ID。
type	设备类型，详见 TXMediaDeviceType 定义。

返回值说明：

0：操作成功；负数：操作失败。

getCurrentDevice:

getCurrentDevice:

- (TXMediaDeviceInfo * _Nullable)getCurrentDevice:	(TXMediaDeviceType)type
--	-------------------------

获取当前正在使用的设备（仅适用于桌面端）

setCurrentDeviceVolume:deviceType:

setCurrentDeviceVolume:deviceType:

- (NSInteger)setCurrentDeviceVolume:	(NSInteger)volume
deviceType:	(TXMediaDeviceType)type

设置当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持设置音量的。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

getCurrentDeviceVolume:

getCurrentDeviceVolume:

- (NSInteger)getCurrentDeviceVolume:	(TXMediaDeviceType)type
--------------------------------------	-------------------------

获取当前设备的音量（仅适用于桌面端）

这里的音量指的是麦克风的采集音量或者扬声器的播放音量，摄像头是不支持获取音量的。

setCurrentDeviceMute:deviceType:

setCurrentDeviceMute:deviceType:

- (NSInteger)setCurrentDeviceMute:	(BOOL)mute
deviceType:	(TXMediaDeviceType)type

设置当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

getCurrentDeviceMute:

getCurrentDeviceMute:

--	--

- (BOOL)getCurrentDeviceMute:	(TXMediaDeviceType)type
-------------------------------	-------------------------

获取当前设备的静音状态（仅适用于桌面端）

这里的音量指的是麦克风和扬声器，摄像头是不支持静音操作的。

enableFollowingDefaultAudioDevice:enable:

enableFollowingDefaultAudioDevice:enable:

- (NSInteger)enableFollowingDefaultAudioDevice:	(TXMediaDeviceType)type
enable:	(BOOL)enable

设置 SDK 使用的音频设备根据跟随系统默认设备（仅适用于桌面端）

仅支持设置麦克风和扬声器类型，摄像头暂不支持跟随系统默认设备

参数	描述
enable	是否跟随系统默认的音频设备。 true：跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 立即切换音频设备。 false：不跟随。当系统默认音频设备发生改变或者有新音频设备插入时，SDK 不会切换音频设备。
type	设备类型，详见 TXMediaDeviceType 定义。

startCameraDeviceTest:

startCameraDeviceTest:

- (NSInteger)startCameraDeviceTest:	(NSView *)view
-------------------------------------	----------------

开始摄像头测试（仅适用于桌面端）

注意

在测试过程中可以使用 [setCurrentDevice](#) 接口切换摄像头。

stopCameraDeviceTest

stopCameraDeviceTest

结束摄像头测试（仅适用于桌面端）

startMicDeviceTest:

startMicDeviceTest:

- (NSInteger)startMicDeviceTest:	(NSInteger)interval
----------------------------------	---------------------

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 **volume** 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。

注意

该接口调用后默认会回播麦克风录制到的声音到扬声器中。

startMicDeviceTest:playback:

startMicDeviceTest:playback:

- (NSInteger)startMicDeviceTest:	(NSInteger)interval
playback:	(BOOL)playback

开始麦克风测试（仅适用于桌面端）

该接口可以测试麦克风是否能正常工作，测试到的麦克风采集音量的大小，会以回调的形式通知给您，其中 **volume** 的取值范围为0 - 100。

参数	描述
interval	麦克风音量的回调间隔。
playback	是否开启回播麦克风声音，开启后用户测试麦克风时会听到自己的声音。

stopMicDeviceTest

stopMicDeviceTest

结束麦克风测试（仅适用于桌面端）

startSpeakerDeviceTest:

startSpeakerDeviceTest:

- (NSInteger)startSpeakerDeviceTest:	(NSString *)audioFilePath
--------------------------------------	---------------------------

开始扬声器测试（仅适用于桌面端）

该接口通过播放指定的音频文件，用于测试播放设备是否能正常工作。如果用户在测试时能听到声音，说明播放设备能正常工作。

参数	描述
filePath	声音文件的路径

stopSpeakerDeviceTest

stopSpeakerDeviceTest

结束扬声器测试（仅适用于桌面端）

setObserver:

setObserver:

- (void)setObserver:	(nullable id<TXDeviceObserver>) observer
----------------------	--

设备热插拔回调（仅适用于 Mac 系统）

setCameraCapturerParam:

setCameraCapturerParam:

- (void)setCameraCapturerParam:	(TXCameraCaptureParam *)params
---------------------------------	--------------------------------

设置摄像头采集偏好

setSystemVolumeType:

setSystemVolumeType:

- (NSInteger)setSystemVolumeType:	(TXSystemVolumeType)type
-----------------------------------	--------------------------

设置系统音量类型（仅适用于移动端）

@deprecated v9.5 版本开始不推荐使用，建议使用 `TRTCCloud` 中的 `startLocalAudio(quality)` 接口替代之，通过 `quality` 参数来决策音质。

TXSystemVolumeType(Deprecated)

TXSystemVolumeType(Deprecated)

系统音量类型（仅适用于移动设备）

@deprecated v9.5 版本开始不推荐使用。

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。

当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。

当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TXSystemVolumeTypeAuto	0	自动切换模式
TXSystemVolumeTypeMedia	1	全程媒体音量
TXSystemVolumeTypeVOIP	2	全程通话音量

TXAudioRoute

TXAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。

手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。

设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

枚举	取值	描述
TXAudioRouteSpeakerphone	0	Speakerphone ：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TXAudioRouteEarpiece	1	Earpiece ：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。

TXMediaDeviceType

TXMediaDeviceType

设备类型（仅适用于桌面平台）

该枚举值用于定义三种类型的音视频设备，即摄像头、麦克风和扬声器，以便让一套设备管理接口可以操控三种不同类型的设备。

枚举	取值	描述
TXMediaDeviceTypeUnknown	-1	未定义的设备类型
TXMediaDeviceTypeAudioInput	0	麦克风类型设备
TXMediaDeviceTypeAudioOutput	1	扬声器类型设备
TXMediaDeviceTypeVideoCamera	2	摄像头类型设备

TXMediaDeviceState

TXMediaDeviceState

设备操作

该枚举值用于本地设备的状态变化通知 [onDeviceChanged](#)。

枚举	取值	描述
TXMediaDeviceStateAdd	0	设备已被插入
TXMediaDeviceStateRemove	1	设备已被移除
TXMediaDeviceStateActive	2	设备已启用
TXMediaDefaultDeviceChanged	3	系统默认设备变更

TXCameraCaptureMode

TXCameraCaptureMode

摄像头采集偏好

该枚举类型用于摄像头采集参数设置。

枚举	取值	描述
TXCameraResolutionStrategyAuto	0	自动调整采集参数。 SDK 根据实际的采集设备性能及网络情况，选择合适的摄像头输出参数，在设备性能及视频预览质量之间，维持平衡。
TXCameraResolutionStrategyPerformance	Not Defined	优先保证设备性能。 SDK 根据用户设置编码器的分辨率和帧率，选择最接近的摄像头输出参数，从而保证设备性能。
TXCameraResolutionStrategyHighQuality	Not Defined	优先保证视频预览质量。 SDK 选择较高的摄像头输出参数，从而提高预览视频的质量。在这种情况下，会消耗更多的 CPU 及内存做视频前处理。
TXCameraCaptureManual	Not Defined	允许用户设置本地摄像头采集的视频宽高。

TXCameraCaptureParam

TXCameraCaptureParam

摄像头采集参数

该设置能决定本地预览图像画质。

枚举类型	描述
height	字段含义： 采集图像宽度
mode	字段含义： 摄像头采集偏好，请参见 TXCameraCaptureMode
width	字段含义： 采集图像长度

TXMediaDeviceInfo

TXMediaDeviceInfo

音视频设备的相关信息（仅适用于桌面平台）

该结构体用于描述一个音视频设备的关键信息，比如设备 ID、设备名称等等，以便用户能够在用户界面上选择自己期望使用的音视频设备。

枚举类型	描述
deviceId	设备 ID（UTF-8）
deviceName	设备名称（UTF-8）
deviceProperties	设备属性
type	设备类型

错误码表

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLiveCode @ TXLiteAVSDK

Function: 腾讯云直播服务(LVB)错误码和警告码的定义。

错误码表

枚举类型

枚举类型	描述
V2TXLiveCode	V2 错误码和警告码

V2TXLiveCode

V2TXLiveCode

V2 错误码和警告码

枚举	取值	描述
V2TXLIVE_OK	0	没有错误。
V2TXLIVE_ERROR_FAILED	-1	暂未归类的通用错误。
V2TXLIVE_ERROR_INVALID_PARAMETER	-2	调用 API 时，传入的参数不合法。
V2TXLIVE_ERROR_REFUSED	-3	API 调用被拒绝。
V2TXLIVE_ERROR_NOT_SUPPORTED	-4	当前 API 不支持调用。
V2TXLIVE_ERROR_INVALID_LICENSE	-5	license 不合法，调用失败。
V2TXLIVE_ERROR_REQUEST_TIMEOUT	-6	请求服务器超时。

V2TXLIVE_ERROR_SERVER_PROCESS_FAILED	-7	服务器无法处理您的请求。
V2TXLIVE_ERROR_DISCONNECTED	-8	连接断开。
V2TXLIVE_ERROR_NO_AVAILABLE_HEVC_DECODERS	-2304	找不到可用的 HEVC 解码器。
V2TXLIVE_WARNING_NETWORK_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻。
V2TXLIVE_WARNING_VIDEO_BLOCK	2105	当前视频播放出现卡顿。
V2TXLIVE_WARNING_CAMERA_START_FAILED	-1301	摄像头打开失败。
V2TXLIVE_WARNING_CAMERA_OCCUPIED	-1316	摄像头正在被占用中，可尝试打开其他摄像头。
V2TXLIVE_WARNING_CAMERA_NO_PERMISSION	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_MICROPHONE_START_FAILED	-1302	麦克风打开失败。
V2TXLIVE_WARNING_MICROPHONE_OCCUPIED	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败。
V2TXLIVE_WARNING_MICROPHONE_NO_PERMISSION	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_SCREEN_CAPTURE_NOT_SUPPORTED	-1309	当前系统不支持屏幕分享。
V2TXLIVE_WARNING_SCREEN_CAPTURE_START_FAILED	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_SCREEN_CAPTURE_INTERRUPTED	-7001	录屏被系统中断。
V2TXLIVE_WARNING_CURRENT_ENCODE_TYPE_CHANGED	1104	表示编码器发生改变，可以通过 <code>onWarning</code> 函数的扩展信息中的 <code>codec_type</code> 字段来获取当前的编码格式。其中 1 代表 265 编码，0 代表 264 编码。注意 Windows 端不支持此错误码的扩展信息。

V2TXLIVE_WARNING_CURRENT_DECODE_TYPE_CHANGED	2008	表示解码器发生改变，可以通过 <code>onWarning</code> 函数的扩展信息中的 <code>codec_type</code> 字段来获取当前的解码格式。其中 1 代表 265 解码，0 代表 264 解码。注意 Windows 端不支持此错误码的扩展信息。
--	------	---

类型定义

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLiveDef @ TXLiteAVSDK

Function: 腾讯云直播服务(LVB) 关键类型定义

类型定义

结构体类型

函数列表	描述
V2TXLiveVideoEncoderParam	视频编码参数
V2TXLiveVideoFrame	视频帧信息
V2TXLiveAudioFrame	音频帧数据
V2TXLiveAudioFrameObserverFormat	音频帧回调格式
V2TXLivePusherStatistics	推流器的统计数据
V2TXLivePlayerStatistics	播放器的统计数据
V2TXLiveMixStream	云端混流中每一路子画面的位置信息
V2TXLiveTranscodingConfig	云端混流（转码）配置
V2TXLiveLocalRecordingParams	本地录制音视频配置
V2TXLiveSocks5ProxyConfig	socks5 代理的协议配置
V2TXLiveLogConfig	Log配置
V2TXLiveStreamInfo	支持自适应切换的码流信息

枚举类型

枚举类型	描述
V2TXLiveMode	支持协议
V2TXLiveVideoResolution	视频分辨率
V2TXLiveVideoResolutionMode	视频宽高比模式
V2TXLiveMirrorType	本地摄像头镜像类型
V2TXLiveFillMode	视频画面填充模式
V2TXLiveRotation	视频画面顺时针旋转角度
V2TXLivePixelFormat	视频帧的像素格式
V2TXLiveBufferType	视频数据包装格式
V2TXLivePictureInPictureState	画中画的状态
V2TXLiveAudioQuality	声音音质
V2TXLiveAudioFrameOperationMode	音频回调数据读写模式
V2TXLivePushStatus	直播流的连接状态
V2TXAudioRoute	声音播放模式（音频路由）
V2TXLiveMixInputType	混流输入类型配置
V2TXLiveRecordMode	本地音视频录制模式
V2TXLiveLogLevel	日志级别枚举值

V2TXLiveMode

V2TXLiveMode

支持协议

枚举	取值	描述
V2TXLiveMode_RTMP	Not Defined	支持协议: RTMP。
V2TXLiveMode_RTC	Not Defined	支持协议: TRTC。

V2TXLiveVideoResolution

V2TXLiveVideoResolution

视频分辨率

枚举	取值	描述
V2TXLiveVideoResolution160x160	Not Defined	分辨率 160*160，码率范围：100Kbps ~ 150Kbps，帧率：15fps。
V2TXLiveVideoResolution270x270	Not Defined	分辨率 270*270，码率范围：200Kbps ~ 300Kbps，帧率：15fps。
V2TXLiveVideoResolution480x480	Not Defined	分辨率 480*480，码率范围：350Kbps ~ 525Kbps，帧率：15fps。
V2TXLiveVideoResolution320x240	Not Defined	分辨率 320*240，码率范围：250Kbps ~ 375Kbps，帧率：15fps。
V2TXLiveVideoResolution480x360	Not Defined	分辨率 480*360，码率范围：400Kbps ~ 600Kbps，帧率：15fps。
V2TXLiveVideoResolution640x480	Not Defined	分辨率 640*480，码率范围：600Kbps ~ 900Kbps，帧率：15fps。
V2TXLiveVideoResolution320x180	Not Defined	分辨率 320*180，码率范围：250Kbps ~ 400Kbps，帧率：15fps。
V2TXLiveVideoResolution480x270	Not Defined	分辨率 480*270，码率范围：350Kbps ~ 550Kbps，帧率：15fps。
V2TXLiveVideoResolution640x360	Not Defined	分辨率 640*360，码率范围：500Kbps ~ 900Kbps，帧率：15fps。
V2TXLiveVideoResolution960x540	Not Defined	分辨率 960*540，码率范围：800Kbps ~ 1500Kbps，帧率：15fps。
V2TXLiveVideoResolution1280x720	Not Defined	分辨率 1280*720，码率范围：1000Kbps ~ 1800Kbps，帧率：15fps。
V2TXLiveVideoResolution1920x1080	Not Defined	分辨率 1920*1080，码率范围：2500Kbps ~ 3000Kbps，帧率：15fps。

V2TXLiveVideoResolutionMode

V2TXLiveVideoResolutionMode

视频宽高比模式

注意

横屏模式下的分辨率: $V2TXLiveVideoResolution640 \times 360 + V2TXLiveVideoResolutionModeLandscape = 640 \times 360$ 。

竖屏模式下的分辨率: $V2TXLiveVideoResolution640 \times 360 + V2TXLiveVideoResolutionModePortrait = 360 \times 640$ 。

枚举	取值	描述
V2TXLiveVideoResolutionModeLandscape	0	横屏模式。
V2TXLiveVideoResolutionModePortrait	1	竖屏模式。

V2TXLiveMirrorType

V2TXLiveMirrorType

本地摄像头镜像类型

枚举	取值	描述
V2TXLiveMirrorTypeAuto	Not Defined	系统默认镜像类型，前置摄像头镜像，后置摄像头不镜像。
V2TXLiveMirrorTypeEnable	Not Defined	前置摄像头和后置摄像头，都切换为镜像模式。
V2TXLiveMirrorTypeDisable	Not Defined	前置摄像头和后置摄像头，都切换为非镜像模式。

V2TXLiveFillMode

V2TXLiveFillMode

视频画面填充模式

枚举	取值	描述
V2TXLiveFillModeFill	Not	图像铺满屏幕，超出显示视窗的视频部分将被裁剪，画

	Defined	面显示可能不完整。
V2TXLiveFillModeFit	Not Defined	图像长边填满屏幕，短边区域会被填充黑色，画面的内容完整。
V2TXLiveFillModeScaleFill	Not Defined	图像拉伸铺满，因此长度和宽度可能不会按比例变化。

V2TXLiveRotation

V2TXLiveRotation

视频画面顺时针旋转角度

枚举	取值	描述
V2TXLiveRotation0	Not Defined	不旋转。
V2TXLiveRotation90	Not Defined	顺时针旋转90度。
V2TXLiveRotation180	Not Defined	顺时针旋转180度。
V2TXLiveRotation270	Not Defined	顺时针旋转270度。

V2TXLivePixelFormat

V2TXLivePixelFormat

视频帧的像素格式

枚举	取值	描述
V2TXLivePixelFormatUnknown	Not Defined	未知。
V2TXLivePixelFormatI420	Not Defined	YUV420P I420。
V2TXLivePixelFormatNV12	Not Defined	YUV420SP NV12。

V2TXLivePixelFormatBGRA32	Not Defined	BGRA8888。
V2TXLivePixelFormatTexture2D	Not Defined	OpenGL 2D 纹理。

V2TXLiveBufferType

V2TXLiveBufferType

视频数据包装格式

注意

在自定义采集和自定义渲染功能，您需要用到下列枚举值来指定您希望以什么样的格式来包装视频数据。

PixelBuffer：直接使用效率最高，iOS 系统提供了众多 API 获取或处理 **PixelBuffer**。

NSData：当使用自定义渲染时，**PixelBuffer**拷贝一次到**NSData**。当使用自定义采集时，**NSData**拷贝一次到**PixelBuffer**。因此，性能会受到一定程度的影响。

枚举	取值	描述
V2TXLiveBufferTypeUnknown	Not Defined	未知。
V2TXLiveBufferTypePixelBuffer	Not Defined	直接使用效率最高，iOS 系统提供了众多 API 获取或处理 PixelBuffer 。
V2TXLiveBufferTypeNSData	Not Defined	会有一定的性能消耗，SDK 内部是直接处理 PixelBuffer 的，所以会存在 NSData 和 PixelBuffer 之间类型转换所产生的内存拷贝开销。
V2TXLiveBufferTypeTexture	Not Defined	直接操作纹理 ID，性能最好。

V2TXLivePictureInPictureState

V2TXLivePictureInPictureState

画中画的状态

枚举	取值	描述
V2TXLivePictureInPictureStateUndefined	Not	未定义。

	Defined	
V2TXLivePictureInPictureStateOccurError	Not Defined	画中画发生错误。
V2TXLivePictureInPictureStateWillStart	Not Defined	画中画将要开始。
V2TXLivePictureInPictureStateDidStart	Not Defined	画中画已经开始。
V2TXLivePictureInPictureStateWillStop	Not Defined	画中画将要停止。
V2TXLivePictureInPictureStateDidStop	Not Defined	画中画已经停止。

V2TXLiveAudioQuality

V2TXLiveAudioQuality

声音音质

枚举	取值	描述
V2TXLiveAudioQualitySpeech	Not Defined	语音音质：采样率：16k；单声道；音频码率：16kbps；适合语音通话为主的场景，比如在线会议，语音通话。
V2TXLiveAudioQualityDefault	Not Defined	默认音质：采样率：48k；单声道；音频码率：50kbps；SDK 默认的音频质量，如无特殊需求推荐选择之。
V2TXLiveAudioQualityMusic	Not Defined	音乐音质：采样率：48k；双声道 + 全频带；音频码率：128kbps；适合需要高保真传输音乐的场景，比如 K 歌、音乐直播等。

V2TXLiveAudioFrameOperationMode

V2TXLiveAudioFrameOperationMode

音频回调数据读写模式

SDK 提供了两种音频回调数据的操作模式。

读写模式（ReadWrite）：可以获取并修改回调的音频数据，默认模式。

只读模式（ReadOnly）：仅从回调中获取音频数据。

枚举	取值	描述
V2TXLiveAudioFrameOperationModeReadWrite	0	读写模式：可以获取并修改回调的音频数据。
V2TXLiveAudioFrameOperationModeReadOnly	1	只读模式：仅从回调中获取音频数据。

V2TXLivePushStatus

V2TXLivePushStatus

直播流的连接状态

枚举	取值	描述
V2TXLivePushStatusDisconnected	Not Defined	与服务器断开连接。
V2TXLivePushStatusConnecting	Not Defined	正在连接服务器。
V2TXLivePushStatusConnectSuccess	Not Defined	连接服务器成功。
V2TXLivePushStatusReconnecting	Not Defined	重连服务器中。

V2TXAudioRoute

V2TXAudioRoute

声音播放模式（音频路由）

枚举	取值	描述
V2TXAudioModeSpeakerphone	Not Defined	扬声器。
V2TXAudioModeEarpiece	Not Defined	听筒。

V2TXLiveMixInputType

V2TXLiveMixInputType

混流输入类型配置

枚举	取值	描述
V2TXLiveMixInputTypeAudioVideo	Not Defined	混入音视频。
V2TXLiveMixInputTypePureVideo	Not Defined	只混入视频。
V2TXLiveMixInputTypePureAudio	Not Defined	只混入音频。

V2TXLiveRecordMode

V2TXLiveRecordMode

本地音视频录制模式

枚举	取值	描述
V2TXLiveRecordModeBoth	Not Defined	Both mode: 录制音频和视频

V2TXLiveLogLevel

V2TXLiveLogLevel

日志级别枚举值

枚举	取值	描述
V2TXLiveLogLevelAll	0	输出所有级别的 log。
V2TXLiveLogLevelDebug	1	输出 DEBUG, INFO, WARNING, ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelInfo	2	输出 INFO, WARNING, ERROR 和 FATAL 级别的 log。

V2TXLiveLogLevelWarning	3	只输出 WARNING, ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelError	4	只输出 ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelFatal	5	只输出 FATAL 级别的 log。
V2TXLiveLogLevelNULL	6	不输出任何 sdk log。

V2TXLiveVideoEncoderParam

V2TXLiveVideoEncoderParam

视频编码参数

该设置决定远端用户看到的画面质量。

枚举类型	描述
minVideoBitrate	【字段含义】最低视频码率，SDK 会在网络不佳的情况下主动降低视频码率以保持流畅度，最低会降至 minVideoBitrate 所设定的数值。 【推荐取值】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoBitrate	【字段含义】目标视频码率，SDK 会按照目标码率进行编码，只有在弱网络环境下才会主动降低视频码率。 【推荐取值】请参考 V2TXLiveVideoResolution 在各档位注释的最佳码率，也可以在此基础上适当调高。 比如：V2TXLiveVideoResolution1280x720 对应 1200kbps 的目标码率，您也可以设置为 1500kbps 用来获得更好的观感清晰度。 【特别说明】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoFps	【字段含义】视频采集帧率。 【推荐取值】15fps 或 20fps。5fps 以下，卡顿感明显。10fps 以下，会有轻微卡顿感。20fps 以上，会浪费带宽（电影的帧率为 24fps）。
videoResolution	【字段含义】视频分辨率。 【特别说明】如需使用竖屏分辨率，请指定 videoResolutionMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。 【推荐取值】 桌面平台（Win + Mac）：建议选择 640 × 360 及以上分辨率，videoResolutionMode 选择 Landscape，即横屏分辨率。

videoResolutionMode	【字段含义】分辨率模式（横屏分辨率 or 竖屏分辨率）。 【推荐取值】桌面平台（Windows、Mac）建议选择 Landscape。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。
---------------------	--

V2TXLiveVideoFrame

V2TXLiveVideoFrame

视频帧信息

注意

自定义采集和自定义渲染时使用。自定义采集时，需要使用 V2TXLiveVideoFrame 来包装待发送的视频帧；自定义渲染时，会返回经过 V2TXLiveVideoFrame 包装的视频帧。

枚举类型	描述
bufferType	【字段含义】视频数据包装格式。 【推荐取值】V2TXLiveBufferTypePixelFormat。
data	【字段含义】bufferType 为 V2TXLiveBufferTypeNSData 时的视频数据。
height	【字段含义】视频高度。
pixelBuffer	【字段含义】bufferType 为 V2TXLiveBufferTypePixelFormat 时的视频数据。
pixelFormat	【字段含义】视频帧像素格式。 【推荐取值】V2TXLivePixelFormatNV12。
rotation	【字段含义】视频帧的顺时针旋转角度。
textureId	【字段含义】视频纹理ID。
width	【字段含义】视频宽度。

V2TXLiveAudioFrame

V2TXLiveAudioFrame

音频帧数据

枚举类型	描述

channel	【字段含义】声道数。
data	【字段含义】音频数据。
sampleRate	【字段含义】采样率。
timestamp	【字段含义】时间戳，单位ms。

V2TXLiveAudioFrameObserverFormat

V2TXLiveAudioFrameObserverFormat

音频帧回调格式

枚举类型	描述
channel	【字段含义】声道数。 【推荐取值】默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
mode	【字段含义】回调数据读写模式。 【推荐取值】V2TXLiveAudioFrameOperationModeReadOnly：仅从回调中获取音频数据。可设定的模式有 V2TXLiveAudioFrameOperationModeReadOnly, V2TXLiveAudioFrameOperationModeReadWrite。
sampleRate	【字段含义】采样率。 【推荐取值】默认值：48000Hz。支持 16000, 32000, 44100, 48000。
samplesPerCall	【字段含义】采样点数。 【推荐取值】取值必须是 sampleRate/100 的整数倍。

V2TXLivePusherStatistics

V2TXLivePusherStatistics

推流器的统计数据

枚举类型	描述
appCpu	【字段含义】当前 App 的 CPU 使用率（%）。
audioBitrate	【字段含义】音频码率（Kbps）。
fps	【字段含义】帧率（fps）。

height	【字段含义】 视频高度。
netSpeed	【字段含义】 上行速度（kbps）
rtt	【字段含义】 从 SDK 到云端的往返延时（ms）
systemCpu	【字段含义】 当前系统的 CPU 使用率（%）。
videoBitrate	【字段含义】 视频码率（Kbps）。
width	【字段含义】 视频宽度。

V2TXLivePlayerStatistics

V2TXLivePlayerStatistics

播放器的统计数据

枚举类型	描述
appCpu	【字段含义】 当前 App 的 CPU 使用率（%）。
audioBitrate	【字段含义】 音频码率（Kbps）。
audioBlockRate	【字段含义】 音频播放卡顿率，单位（%）。 音频播放卡顿率（audioBlockRate）= 音频播放的累计卡顿时长（audioTotalBlockTime）/ 音频播放的区间时长（2000ms）。
audioPacketLoss	【字段含义】 网络音频丢包率（%），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
audioTotalBlockTime	【字段含义】 音频播放的累计卡顿时长（ms）。 该时长为区间（2s）内的卡顿时长。
fps	【字段含义】 帧率（fps）。
height	【字段含义】 视频高度。
jitterBufferDelay	【字段含义】 播放延迟（ms）。
netSpeed	【字段含义】 下载速度（kbps）
rtt	【字段含义】 从 SDK 到云端的往返延时（ms），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
systemCpu	【字段含义】 当前系统的 CPU 使用率（%）。

videoBitrate	【字段含义】视频码率（Kbps）。
videoBlockRate	【字段含义】视频播放卡顿率，单位（%）。 视频播放卡顿率（videoBlockRate）= 视频播放的累计卡顿时长（videoTotalBlockTime）/ 视频播放的区间时长（2000ms）。
videoPacketLoss	【字段含义】网络视频丢包率（%），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
videoTotalBlockTime	【字段含义】视频播放的累计卡顿时长（ms）。 该时长为区间（2s）内的卡顿时长。
width	【字段含义】视频宽度。

V2TXLiveMixStream

V2TXLiveMixStream

云端混流中每一路子画面的位置信息

枚举类型	描述
height	【字段含义】图层位置高度（绝对像素值）。
inputType	【字段含义】该直播流的输入类型。
streamId	【字段含义】参与混流的 userId 所在对应的推流 streamId，nil 表示当前推流 streamId。
userId	【字段含义】参与混流的 userId。
width	【字段含义】图层位置宽度（绝对像素值）。
x	【字段含义】图层位置 x 坐标（绝对像素值）。
y	【字段含义】图层位置 y 坐标（绝对像素值）。
zOrder	【字段含义】图层层级（1 - 15）不可重复。

V2TXLiveTranscodingConfig

V2TXLiveTranscodingConfig

云端混流（转码）配置

--	--

枚举类型	描述
audioBitrate	【字段含义】最终转码后的音频码率。 【推荐取值】默认值：64kbps，取值范围是 [32, 192]，单位：kbps。
audioChannels	【字段含义】最终转码后的音频声道数。 【推荐取值】默认值：1。取值范围为 [1,2] 中的整型。
audioSampleRate	【字段含义】最终转码后的音频采样率。 【推荐取值】默认值：48000Hz。支持12000HZ、16000HZ、22050HZ、24000HZ、32000HZ、44100HZ、48000HZ。
backgroundColor	【字段含义】混合后画面的底色颜色，默认为黑色，格式为十六进制数字，比如：“0x61B9F1”代表 RGB 分别为(97,158,241)。 【推荐取值】默认值：0x000000，黑色。
backgroundImage	【字段含义】混合后画面的背景图。 【推荐取值】默认值：nil，即不设置背景图。 【特别说明】背景图需要您事先在“控制台 => 应用管理 => 功能配置 => 素材管理”中上传，上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置到 backgroundImage 里即可。 例如：假设“图片ID”为 63，可以设置 backgroundImage = "63"。
mixStreams	【字段含义】每一路子画面的位置信息。
outputStreamId	【字段含义】输出到 CDN 上的直播流 ID。 如不设置该参数，SDK 会执行默认逻辑，即房间里的多路流会混合到该接口调用者的视频流上，也就是 $A + B \Rightarrow A$。 如果设置该参数，SDK 会将房间里的多路流混合到您指定的直播流 ID 上，也就是 $A + B \Rightarrow C$。 【推荐取值】默认值：nil，即房间里的多路流会混合到该接口调用者的视频流上。
videoBitrate	【字段含义】最终转码后的视频分辨率的码率（kbps）。 【推荐取值】如果填0，后台会根据 videoWidth 和 videoHeight 来估算码率，您也可以参考枚举定义 V2TXLiveVideoResolution 的注释。
videoFramerate	【字段含义】最终转码后的视频分辨率的帧率（FPS）。 【推荐取值】默认值：15fps，取值范围是 (0,30]。
videoGOP	【字段含义】最终转码后的视频分辨率的关键帧间隔（又称为 GOP）。 【推荐取值】默认值：2，单位为秒，取值范围是 [1,8]。
videoHeight	【字段含义】最终转码后的视频分辨率的高度。 【推荐取值】推荐值：640px，如果你是纯音频推流，请将 width × height 设为 0px × 0px，否则混流后会携带一条画布背景的视频流。
videoWidth	【字段含义】最终转码后的视频分辨率的宽度。

【推荐取值】推荐值：360px，如果你是纯音频推流，请将 width × height 设为 0px × 0px，否则混流后会携带一条画布背景的视频流。

V2TXLiveLocalRecordingParams

V2TXLiveLocalRecordingParams

本地录制音视频配置

枚举类型	描述
filePath	【字段含义】录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。 【推荐取值】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。
interval	【字段含义】interval 录制信息更新频率，单位毫秒，有效范围：1000-10000。 【推荐取值】-1，表示不回调。
recordMode	【字段含义】媒体录制模式。 【推荐取值】V2TXLiveRecordModeBoth，即同时录制音频和视频。

V2TXLiveSocks5ProxyConfig

V2TXLiveSocks5ProxyConfig

socks5 代理的协议配置

枚举类型	描述
supportHttps	【字段含义】是否支持 https。 【推荐取值】默认值：true。
supportTcp	【字段含义】是否支持 tcp。 【推荐取值】默认值：true。
supportUdp	【字段含义】是否支持 udp。 【推荐取值】默认值：true。

V2TXLiveLogConfig

V2TXLiveLogConfig

Log配置

枚举类型	描述
enableConsole	【字段含义】是否允许 SDK 在编辑器（XCoder、Android Studio、Visual Studio 等）的控制台上打印 Log。 【推荐取值】默认值：NO。
enableLogFile	【字段含义】是否启用本地 Log 文件。 【特殊说明】如非特殊需要，请不要关闭本地 Log 文件，否则腾讯云技术团队将无法在出现问题时进行跟踪和定位。 【推荐取值】默认值：YES。
enableObserver	【字段含义】是否通过 V2TXLivePremierObserver 接收要打印的 Log 信息。 【特殊说明】如果您希望自己实现 Log 写入，可以打开此开关，Log 信息会通过 V2TXLivePremierObserver#onLog 回调给您。 【推荐取值】默认值：NO。
logLevel	【字段含义】设置 Log 级别。 【推荐取值】默认值：V2TXLiveLogLevelAll。
logPath	【字段含义】设置本地 Log 的存储目录，默认 Log 存储位置： iOS & Mac: sandbox Documents/log。

V2TXLiveStreamInfo

V2TXLiveStreamInfo

支持自适应切换的码流信息

枚举类型	描述
bitrate	【字段含义】 码率, 单位bps, 默认值：0, 表示未知。
framerate	【字段含义】 帧率, 默认值：0, 表示未知。
height	【字段含义】 视频高, 默认值：0, 表示未知。
url	【字段含义】 流地址, 通过 SwitchStream 接口调用实现多码率质量切换。
width	【字段含义】 视频宽, 默认值：0, 表示未知。

Android

API 概览

最近更新时间：2024-06-06 15:47:57

API OVERVIEW

V2 推流器相关接口

函数列表	描述
release	释放 V2TXLivePusher 资源
setObserver	设置推流器回调
setRenderView	设置本地摄像头预览 View
setRenderMirror	设置本地摄像头预览镜像
setEncoderMirror	设置视频编码镜像
setRenderRotation	设置本地摄像头预览画面的旋转角度
setRenderFillMode	设置本地摄像头预览画面的填充模式
startCamera	打开本地摄像头
stopCamera	关闭本地摄像头
startMicrophone	打开麦克风
stopMicrophone	关闭麦克风
startVirtualCamera	开启图片推流
stopVirtualCamera	关闭图片推流
startScreenCapture	开启屏幕采集
stopScreenCapture	关闭屏幕采集
pauseAudio	暂停推流器的音频流
resumeAudio	恢复推流器的音频流

pauseVideo	暂停推流器的视频流
resumeVideo	恢复推流器的视频流
startPush	开始音视频数据推流
stopPush	停止推送音视频数据
isPushing	当前推流器是否正在推流中
setAudioQuality	设置推流音频质量
setVideoQuality	设置推流视频编码参数
getAudioEffectManager	获取音效管理对象
getBeautyManager	获取美颜管理对象
getDeviceManager	获取设备管理对象
snapshot	截取推流过程中的本地画面
setWatermark	设置推流器水印。默认情况下，水印不开启
enableVolumeEvaluation	启用采集音量大小提示
enableCustomVideoProcess	开启/关闭自定义视频处理
enableCustomVideoCapture	开启/关闭自定义视频采集
enableCustomAudioCapture	开启/关闭自定义音频采集
sendCustomVideoFrame	在自定义视频采集模式下，将采集的视频数据发送到SDK
sendCustomAudioFrame	在自定义音频采集模式下，将采集的音频数据发送到SDK
enableAudioProcessObserver	开启/关闭对经过前处理后的本地音频帧的监听回调
sendSeiMessage	发送 SEI 消息
startSystemAudioLoopback	打开系统声音采集
stopSystemAudioLoopback	关闭系统声音采集
showDebugView	显示仪表盘。
setProperty	调用 V2TXLivePusher 的高级 API 接口。
setMixTranscodingConfig	设置云端的混流转码参数

startLocalRecording	开始录制音视频流
stopLocalRecording	停止录制音视频流
enableVoiceActivityDetection	启用人声检测

直播推流器事件回调

函数列表	描述
onError	直播推流器错误通知，推流器出现错误时，会回调该通知
onWarning	直播推流器警告通知
onCaptureFirstAudioFrame	首帧音频采集完成的回调通知
onCaptureFirstVideoFrame	首帧视频采集完成的回调通知
onMicrophoneVolumeUpdate	麦克风采集音量值回调
onPushStatusUpdate	推流器连接状态回调通知
onStatisticsUpdate	直播推流器统计数据回调
onSnapshotComplete	截图回调
onGLContextCreated	SDK 内部的 OpenGL 环境的创建通知
onProcessAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onProcessVideoFrame	自定义视频处理回调
onGLContextDestroyed	SDK 内部的 OpenGL 环境的销毁通知
onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 setMixTranscodingConfig 接口
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCaptureStopped	当屏幕分享停止时，SDK 会通过此回调通知
onLocalRecordBegin	录制任务开始的事件回调
onLocalRecording	录制任务正在进行中的进展事件回调
onLocalRecordComplete	录制任务已经结束的事件回调

`onVoiceActivityDetectionUpdate`

调用 `enableVoiceActivityDetection` 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

V2 播放器相关接口

函数列表	描述
<code>setObserver</code>	设置播放器回调
<code>setRenderView</code>	设置播放器的视频渲染 <code>View</code> ，该控件负责显示视频内容
<code>setRenderRotation</code>	设置播放器画面的旋转角度
<code>setRenderFillMode</code>	设置画面的填充模式
<code>startLivePlay</code>	开始播放音视频流
<code>stopPlay</code>	停止播放音视频流
<code>isPlaying</code>	播放器是否正在播放中
<code>pauseAudio</code>	暂停播放器的音频流
<code>resumeAudio</code>	恢复播放器的音频流
<code>pauseVideo</code>	暂停播放器的视频流
<code>resumeVideo</code>	恢复播放器的视频流
<code>setPlayoutVolume</code>	设置播放器音量
<code>setCacheParams</code>	设置播放器缓存自动调整的最小和最大时间（单位：秒）
<code>switchStream</code>	直播流无缝切换，支持 FLV 和 LEB
<code>getStreamList</code>	获取码流信息
<code>enableVolumeEvaluation</code>	启用播放音量大小提示
<code>snapshot</code>	截取播放过程中的视频画面
<code>enableObserveVideoFrame</code>	开启/关闭对视频帧的监听回调
<code>enableObserveAudioFrame</code>	开启/关闭对音频数据的监听回调
<code>enableReceiveSeiMessage</code>	开启接收 SEI 消息

<code>showDebugView</code>	是否显示播放器状态信息的调试浮层
<code>setProperty</code>	调用 <code>V2TXLivePlayer</code> 的高级 API 接口
<code>startLocalRecording</code>	开始录制音视频流
<code>stopLocalRecording</code>	停止录制音视频流

直播播放器事件回调

函数列表	描述
<code>onError</code>	直播播放器错误通知，播放器出现错误时，会回调该通知
<code>onWarning</code>	直播播放器警告通知
<code>onVideoResolutionChanged</code>	直播播放器分辨率变化通知
<code>onConnected</code>	已经成功连接到服务器
<code>onVideoPlaying</code>	视频播放事件
<code>onAudioPlaying</code>	音频播放事件
<code>onVideoLoading</code>	视频加载事件
<code>onAudioLoading</code>	音频加载事件
<code>onPlayoutVolumeUpdate</code>	播放器音量大小回调
<code>onStatisticsUpdate</code>	直播播放器统计数据回调
<code>onSnapshotComplete</code>	截图回调
<code>onRenderVideoFrame</code>	自定义视频渲染回调
<code>onPlayoutAudioFrame</code>	音频数据回调
<code>onReceiveSeiMessage</code>	收到 SEI 消息的回调，发送端通过 <code>V2TXLivePusher</code> 中的 <code>sendSeiMessage</code> 来发送 SEI 消息
<code>onStreamSwitched</code>	分辨率无缝切换回调
<code>onLocalRecordBegin</code>	录制任务开始的事件回调
<code>onLocalRecording</code>	录制任务正在进行的进展事件回调

`onLocalRecordComplete`

录制任务已经结束的事件回调

V2TXLive 高级接口

函数列表	描述
<code>getSDKVersionStr</code>	获取 SDK 版本号
<code>setObserver</code>	设置 V2TXLivePremier 回调接口
<code>setLogConfig</code>	设置 Log 的配置信息
<code>setEnvironment</code>	设置 SDK 接入环境
<code>setLicence</code>	设置 SDK 的授权 License
<code>setSocks5Proxy</code>	设置 SDK socks5 代理配置
<code>enableAudioCaptureObserver</code>	开启/关闭对音频采集数据的监听回调（可读写）
<code>enableAudioPlayoutObserver</code>	开启/关闭对最终系统要播放出的音频数据的监听回调
<code>enableVoiceEarMonitorObserver</code>	开启/关闭耳返音频数据的监听回调
<code>setUserId</code>	设置 userId
<code>callExperimentalAPI</code>	调用实验性 API 接口

V2TXLive 高级回调接口

函数列表	描述
<code>onLog</code>	自定义 Log 输出回调接口
<code>onLicenceLoaded</code>	<code>setLicence</code> 接口回调
<code>onCaptureAudioFrame</code>	本地麦克风采集到的音频数据回调
<code>onPlayoutAudioFrame</code>	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
<code>onVoiceEarMonitorAudioFrame</code>	耳返的音频数据

背景音乐预加载事件回调

函数列表	描述
onLoadProgress	背景音乐预加载进度
onLoadError	背景音乐预加载出错

背景音乐的播放事件回调

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

人声相关的特效接口

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitorVolume	设置耳返音量
setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量
setVoicePitch	设置语音音调

背景音乐的相关接口

函数列表	描述
setMusicObserver	设置背景音乐的事件回调接口

startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调
preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

美颜相关接口

函数列表	描述
setBeautyStyle	设置美颜（磨皮）算法
setBeautyLevel	设置美颜级别
setWhitenessLevel	设置美白级别
enableSharpnessEnhancement	开启清晰度增强
setRuddyLevel	设置红润级别

setFilter	设置色彩滤镜效果
setFilterStrength	设置色彩滤镜的强度
setGreenScreenFile	设置绿幕背景视频
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩
setFaceShortLevel	设置短脸级别
setFaceNarrowLevel	设置窄脸级别
setNoseSlimLevel	设置瘦鼻级别
setEyeLightenLevel	设置亮眼级别
setToothWhitenLevel	设置牙齿美白级别
setWrinkleRemoveLevel	设置祛皱级别
setPouchRemoveLevel	设置祛眼袋级别
setSmileLinesRemoveLevel	设置法令纹去除级别
setForeheadLevel	设置发际线调整级别
setEyeDistanceLevel	设置眼距
setEyeAngleLevel	设置眼角调整级别
setMouthShapeLevel	设置嘴型调整级别
setNoseWingLevel	设置鼻翼调整级别
setNosePositionLevel	设置鼻子位置
setLipsThicknessLevel	设置嘴唇厚度
setFaceBeautyLevel	设置脸型
setMotionTpl	选择 AI 动效挂件
setMotionMute	是否在动效素材播放时静音

设备操作接口

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorch	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute	设置音频路由（仅适用于移动端）
setExposureCompensation	设置摄像头的曝光参数，取值范围从-1到1
setCameraCapturerParam	设置摄像头采集偏好

弃用接口

函数列表	描述
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

V2TXLivePusher

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePusher @ TXLiteAVSDK

Function: 腾讯云直播推流器

功能

腾讯云直播推流器

介绍

主要负责将本地的音频和视频画面进行编码，并推送到指定的推流地址，支持任意的推流服务端。

推流器包含如下能力：

自定义的视频采集，让您可以根据项目需要定制自己的音视频数据源。

美颜、滤镜、贴纸，包含多套美颜磨皮算法（自然&光滑）和多款色彩空间滤镜（支持自定义滤镜）。

Qos 流量控制技术，具备上行网络自适应能力，可以根据主播端网络的具体情况实时调节音视频数据量。

脸型调整、动效挂件，支持基于优图 AI 人脸识别技术的大眼、瘦脸、隆鼻等脸型微调以及动效挂件效果，只需要购买 优图 License 就可以轻松实现丰富的直播效果。

V2TXLivePusher

V2TXLivePusher

函数列表	描述
release	释放 V2TXLivePusher 资源
setObserver	设置推流器回调
setRenderView	设置本地摄像头预览 View

setRenderView	设置本地摄像头预览 View
setRenderView	设置本地摄像头预览 View
setRenderMirror	设置本地摄像头预览镜像
setEncoderMirror	设置视频编码镜像
setRenderRotation	设置本地摄像头预览画面的旋转角度
setRenderFillMode	设置本地摄像头预览画面的填充模式
startCamera	打开本地摄像头
stopCamera	关闭本地摄像头
startMicrophone	打开麦克风
stopMicrophone	关闭麦克风
startVirtualCamera	开启图片推流
stopVirtualCamera	关闭图片推流
startScreenCapture	开启屏幕采集
stopScreenCapture	关闭屏幕采集
pauseAudio	暂停推流器的音频流
resumeAudio	恢复推流器的音频流
pauseVideo	暂停推流器的视频流
resumeVideo	恢复推流器的视频流
startPush	开始音视频数据推流
stopPush	停止推送音视频数据
isPushing	当前推流器是否正在推流中
setAudioQuality	设置推流音频质量
setVideoQuality	设置推流视频编码参数
getAudioEffectManager	获取音效管理对象
getBeautyManager	获取美颜管理对象

<code>getDeviceManager</code>	获取设备管理对象
<code>snapshot</code>	截取推流过程中的本地画面
<code>setWatermark</code>	设置推流器水印。默认情况下，水印不开启
<code>enableVolumeEvaluation</code>	启用采集音量大小提示
<code>enableCustomVideoProcess</code>	开启/关闭自定义视频处理
<code>enableCustomVideoCapture</code>	开启/关闭自定义视频采集
<code>enableCustomAudioCapture</code>	开启/关闭自定义音频采集
<code>sendCustomVideoFrame</code>	在自定义视频采集模式下，将采集的视频数据发送到SDK
<code>sendCustomAudioFrame</code>	在自定义音频采集模式下，将采集的音频数据发送到SDK
<code>enableAudioProcessObserver</code>	开启/关闭对经过前处理后的本地音频帧的监听回调
<code>sendSeiMessage</code>	发送 SEI 消息
<code>startSystemAudioLoopback</code>	打开系统声音采集
<code>stopSystemAudioLoopback</code>	关闭系统声音采集
<code>showDebugView</code>	显示仪表盘。
<code>setProperty</code>	调用 V2TXLivePusher 的高级 API 接口。
<code>setMixTranscodingConfig</code>	设置云端的混流转码参数
<code>startLocalRecording</code>	开始录制音视频流
<code>stopLocalRecording</code>	停止录制音视频流
<code>enableVoiceActivityDetection</code>	启用人声检测

release

release

释放 V2TXLivePusher 资源

setObserver

setObserver

void setObserver	(V2TXLivePusherObserver observer)
------------------	--

设置推流器回调

通过设置回调，可以监听 [V2TXLivePusher](#) 推流器的一些回调事件，包括推流器状态、音量回调、统计数据、警告和错误信息等。

参数	描述
observer	推流器的回调目标对象，更多信息请查看 V2TXLivePusherObserver 。

setRenderView

setRenderView

int setRenderView	(TXCloudVideoView view)
-------------------	--

设置本地摄像头预览 View

本地摄像头采集到的画面，经过美颜、脸型调整、滤镜等多种效果叠加之后，最终会显示到传入的 **View** 上。

参数	描述
view	本地摄像头预览 View 。

返回值说明：

返回值 [V2TXLiveCode](#)。
V2TXLIVE_OK：成功。

setRenderView

setRenderView

int setRenderView	(TextureView view)
-------------------	-------------------------------------

设置本地摄像头预览 View

本地摄像头采集到的画面，经过美颜、脸型调整、滤镜等多种效果叠加之后，最终会显示到传入的 **View** 上。

参数	描述

view	本地摄像头预览 View。
------	---------------

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK：成功。

setRenderView

setRenderView

int setRenderView	(SurfaceView view)
-------------------	--------------------

设置本地摄像头预览 View

本地摄像头采集到的画面，经过美颜、脸型调整、滤镜等多种效果叠加之后，最终会显示到传入的 View 上。

参数	描述
view	本地摄像头预览 View。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK：成功。

setRenderMirror

setRenderMirror

int setRenderMirror	(V2TXLiveMirrorType mirrorType)
---------------------	--

设置本地摄像头预览镜像

本地摄像头分为前置摄像头和后置摄像头，系统默认情况下，是前置摄像头镜像，后置摄像头不镜像，这里可以修改前置后置摄像头的默认镜像类型。

参数	描述
mirrorType	摄像头镜像类型 V2TXLiveMirrorType 。 V2TXLiveMirrorTypeAuto 【默认值】：默认镜像类型. 在这种情况下，前置摄像头的画面是镜像的，后置摄像头的画面不是镜像的。 V2TXLiveMirrorTypeEnable: 前置摄像头 和 后置摄像头，都切换为镜像模式。

V2TXLiveMirrorTypeDisable: 前置摄像头 和 后置摄像头，都切换为非镜像模式。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setEncoderMirror

setEncoderMirror

int setEncoderMirror	(boolean mirror)
----------------------	------------------

设置视频编码镜像

参数	描述
mirror	是否镜像。 false 【默认值】：播放端看到的是非镜像画面。 true: 播放端看到的是镜像画面。

注意

编码镜像只影响观众端看到的视频效果。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setRenderRotation

setRenderRotation

int setRenderRotation	(V2TXLiveRotation rotation)
-----------------------	--

设置本地摄像头预览画面的旋转角度

参数	描述
rotation	预览画面的旋转角度 V2TXLiveRotation 。 V2TXLiveRotation0 【默认值】：0度，不旋转。 V2TXLiveRotation90: 顺时针旋转90度。

	V2TXLiveRotation180: 顺时针旋转180度。 V2TXLiveRotation270: 顺时针旋转270度。
--	--

注意

只旋转本地预览画面，不影响推流出去的画面。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setRenderFillMode

setRenderFillMode

int setRenderFillMode	(V2TXLiveFillMode mode)
-----------------------	--

设置本地摄像头预览画面的填充模式

参数	描述
mode	画面填充模式 V2TXLiveFillMode 。 V2TXLiveFillModeFill 【默认值】：图像铺满屏幕，不留黑边，如果图像宽高比不同于屏幕宽高比，部分画面内容会被裁剪掉。 V2TXLiveFillModeFit: 图像适应屏幕，保持画面完整，但如果图像宽高比不同于屏幕宽高比，会有黑边的存在。 V2TXLiveFillModeScaleFill: 图像拉伸铺满，因此长度和宽度可能不会按比例变化。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startCamera

startCamera

int startCamera	(boolean frontCamera)
-----------------	-----------------------

打开本地摄像头

参数	描述
----	----

frontCamera	指定摄像头方向是否为前置。 true 【默认值】: 切换到前置摄像头。 false: 切换到后置摄像头。
-------------	--

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下, 仅有一个采集源可以上行, 不同采集源之间切换, 请先关闭前一采集源, 再开启后一采集源, 保证同一采集源的开启和关闭是成对调用的。比如: 采集源从Camera切换到VirtualCamera, 调用顺序是 startCamera -> stopCamera -> startVirtualCamera。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopCamera

stopCamera

关闭本地摄像头

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startMicrophone

startMicrophone

打开麦克风

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopMicrophone

stopMicrophone

关闭麦克风

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startVirtualCamera

startVirtualCamera

int startVirtualCamera	(Bitmap image)
------------------------	----------------

开启图片推流

参数	描述
image	图片。

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下, 仅有一个采集源可以上行, 不同采集源之间切换, 请先关闭前一采集源, 再开启后一采集源, 保证同一采集源的开启和关闭是成对调用的。比如: 采集源从Camera切换到VirtualCamera, 调用顺序是 startCamera -> stopCamera -> startVirtualCamera。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopVirtualCamera

stopVirtualCamera

关闭图片推流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startScreenCapture

startScreenCapture

开启屏幕采集

注意

startVirtualCamera, startCamera, startScreenCapture, 同一 Pusher 实例下, 仅有一个采集源可以上行, 不同采集源之间切换, 请先关闭前一采集源, 再开启后一采集源, 保证同一采集源的开启和关闭是成对调用的。比如: 采集源从Camera切换到ScreenCapture, 调用顺序是 startCamera -> stopCamera -> startScreenCapture。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

stopScreenCapture

stopScreenCapture

关闭屏幕采集

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseAudio

pauseAudio

暂停推流器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeAudio

resumeAudio

恢复推流器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseVideo

pauseVideo

暂停推流器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeVideo

resumeVideo

恢复推流器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startPush

startPush

int startPush	(String url)
---------------	--------------

开始音视频数据推流

参数	描述
url	推流的目标地址，支持任意推流服务端。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 操作成功，开始连接推流目标地址。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，url 不合法。

V2TXLIVE_ERROR_INVALID_LICENSE: 操作失败，license 不合法，鉴权失败。

V2TXLIVE_ERROR_REFUSED: 操作失败，RTC 不支持同一设备上同时推拉同一个 StreamId。

stopPush

stopPush

停止推送音视频数据

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

isPushing

isPushing

当前推流器是否正在推流中

返回值说明：

是否正在推流。

1: 正在推流中。

0: 已经停止推流。

setAudioQuality

setAudioQuality

int setAudioQuality	(V2TXLiveAudioQuality quality)
---------------------	---

设置推流音频质量

参数	描述
quality	音频质量 V2TXLiveAudioQuality 。 V2TXLiveAudioQualityDefault 【默认值】：通用。 V2TXLiveAudioQualitySpeech: 语音。

V2TXLiveAudioQualityMusic: 音乐。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 推流过程中，不允许调整音质。

setVideoQuality

setVideoQuality

int setVideoQuality	(V2TXLiveVideoEncoderParam param)
---------------------	--

设置推流视频编码参数

参数	描述
param	视频编码参数 V2TXLiveVideoEncoderParam 。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

getAudioEffectManager

getAudioEffectManager

获取音效管理对象

通过音效管理，您可以使用以下功能：

调整麦克风收集的人声音量。

设置混响和变声效果。

开启耳返，设置耳返音量。

添加背景音乐，调整背景音乐的播放效果。

参考 [TXAudioEffectManager](#)

getBeautyManager

getBeautyManager

获取美颜管理对象

通过美颜管理，您可以使用以下功能：

设置“美颜风格”、“美白”、“红润”、“大眼”、“瘦脸”、“V脸”、“下巴”、“短脸”、“小鼻”、“亮眼”、“白牙”、“祛眼袋”、“祛皱纹”、“祛法令纹”等美容效果。

调整“发际线”、“眼间距”、“眼角”、“嘴形”、“鼻翼”、“鼻子位置”、“嘴唇厚度”、“脸型”。

设置人脸挂件（素材）等动态效果。

添加美妆。

进行手势识别。

参考 [TXBeautyManager](#)

getDeviceManager

getDeviceManager

获取设备管理对象

通过设备管理，您可以使用以下功能：

切换前后摄像头。

设置自动聚焦。

设置摄像头缩放倍数。

打开或关闭闪光灯。

切换耳机或者扬声器。

修改音量类型(媒体音量或者通话音量)。

参考 [TXDeviceManager](#)

snapshot

snapshot

截取推流过程中的本地画面

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 已经停止推流，不允许调用截图操作。

setWatermark

setWatermark

int setWatermark	(Bitmap image
	float x
	float y
	float scale)

设置推流器水印。默认情况下，水印不开启

参数	描述
image	水印图片。如果该值为 null，则等效于禁用水印。
scale	水印图片的缩放比例，取值范围为0 - 1的浮点数。
x	水印的横坐标，取值范围为0 - 1的浮点数。
y	水印的纵坐标，取值范围为0 - 1的浮点数。

返回值说明：

返回值 [V2TXLiveCode](#)

V2TXLIVE_OK: 成功

enableVolumeEvaluation

enableVolumeEvaluation

int enableVolumeEvaluation	(int intervalMs)
----------------------------	------------------

启用采集音量大小提示

开启后可以在 [onMicrophoneVolumeUpdate](#) 回调中获取到 SDK 对音量大小值的评估。

参数	描述
intervalMs	决定了 onMicrophoneVolumeUpdate 回调的触发间隔，单位为ms，最小间隔为100ms，如果小于等于0则会关闭回调，建议设置为300ms；【默认值】：0，不开启。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enableCustomVideoProcess

enableCustomVideoProcess

int enableCustomVideoProcess	(boolean enable
	V2TXLivePixelFormat pixelFormat
	V2TXLiveBufferType bufferType)

开启/关闭自定义视频处理

参数	描述
enable	true: 开启; false: 关闭。【默认值】：false。

注意

支持的格式组合：

V2TXLivePixelFormatTexture2D+V2TXLiveBufferTypeTexture

V2TXLivePixelFormatI420+V2TXLiveBufferTypeByteBuffer

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_NOT_SUPPORTED: 不支持的格式。

enableCustomVideoCapture

enableCustomVideoCapture

int enableCustomVideoCapture	(boolean enable)
------------------------------	------------------

开启/关闭自定义视频采集

在自定义视频采集模式下，SDK 不再从摄像头采集图像，只保留编码和发送能力。

参数	描述
enable	true：开启自定义采集；false：关闭自定义采集。【默认值】：false。

注意

需要在 [startPush](#) 之前调用，才会生效。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

enableCustomAudioCapture

enableCustomAudioCapture

int enableCustomAudioCapture	(boolean enable)
------------------------------	------------------

开启/关闭自定义音频采集

@brief 开启/关闭自定义音频采集。

在自定义音频采集模式下，SDK 不再从麦克风采集声音，只保留编码和发送能力。

参数	描述
enable	true: 开启自定义采集; false: 关闭自定义采集。【默认值】：false。

注意

需要在 [startPush](#) 前调用才会生效。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

sendCustomVideoFrame

sendCustomVideoFrame

int sendCustomVideoFrame	(V2TXLiveVideoFrame videoFrame)
--------------------------	--

在自定义视频采集模式下，将采集的视频数据发送到SDK

在自定义视频采集模式下，SDK不再采集摄像头数据，仅保留编码和发送功能。

参数	描述
videoFrame	向 SDK 发送的 视频帧数据 V2TXLiveVideoFrame 。

注意

需要在 [startPush](#) 之前调用 [enableCustomVideoCapture](#) 开启自定义采集。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 发送失败，视频帧数据不合法。

sendCustomAudioFrame

sendCustomAudioFrame

int sendCustomAudioFrame	(V2TXLiveAudioFrame audioFrame)
--------------------------	--

在自定义音频采集模式下，将采集的音频数据发送到SDK

@info 在自定义音频采集模式下，将采集的音频数据发送到SDK，SDK不再采集麦克风数据，仅保留编码和发送功能。

@param audioFrame 向 SDK 发送的 音频帧数据 [V2TXLiveAudioFrame](#)。

@return 返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 发送失败，音频帧数据不合法。

@note 需要在 [startPush](#) 之前调用 [enableCustomAudioCapture](#) 开启自定义采集。

enableAudioProcessObserver

enableAudioProcessObserver

int enableAudioProcessObserver	(boolean enable
	V2TXLiveDef.V2TXLiveAudioFrameObserverFormat format)

开启/关闭对经过前处理后的本地音频帧的监听回调

参数	描述
enable	是否开启。【默认值】：false。
format	设置回调出的 AudioFrame 的格式。

注意

需要在 [startPush](#) 之前调用，才会生效。

sendSeiMessage

sendSeiMessage

int sendSeiMessage	(int payloadType
	byte[] data)

发送 SEI 消息

播放端 [V2TXLivePlayer](#) 通过 [V2TXLivePlayerObserver](#) 中的 `onReceiveSeiMessage` 回调来接收该消息。

参数	描述
data	待发送的数据。
payloadType	数据类型，支持 5、242。推荐填：242。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startSystemAudioLoopback

startSystemAudioLoopback

打开系统声音采集

开启后可以采集整个操作系统的播放声音，并将其混入到当前麦克风采集的声音中一起发送到云端。

注意

1. 该接口只在 Android API 29 及以上的版本生效。
2. 您需要先使用该接口来开启系统声音采集，当使用屏幕分享接口开启屏幕分享时才会真正生效。
3. 您需要添加一个前台服务来确保系统声音采集不被静默，并设置 `android:foregroundServiceType="mediaProjection"`。
4. SDK 只采集满足捕获策略和音频用法的应用音频，目前采集的音频用法包括 `USAGE_MEDIA`, `USAGE_GAME`。

stopSystemAudioLoopback

stopSystemAudioLoopback

关闭系统声音采集

showDebugView

showDebugView

<code>void showDebugView</code>	(boolean isShow)
---------------------------------	------------------

显示仪表盘。

参数	描述
isShow	是否显示。【默认值】： <code>false</code> 。

setProperty

setProperty

<code>int setProperty</code>	(String key
	Object value)

调用 V2TXLivePusher 的高级 API 接口。

参数	描述
key	高级 API 对应的 key, 详情请参考 V2TXLiveProperty 定义。
value	调用 key 所对应的高级 API 时, 需要的参数。

注意

该接口用于调用一些高级功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败, key 不允许为空。

setMixTranscodingConfig

setMixTranscodingConfig

int setMixTranscodingConfig	(V2TXLiveTranscodingConfig config)
-----------------------------	---

设置云端的混流转码参数

如果您在实时音视频 [控制台](#) 中的功能配置页开启了“启用旁路推流”功能，房间里的每一路画面都会有一个默认的直播 **CDN 地址**。

一个直播间中可能有不止一位主播，而且每个主播都有自己的画面和声音，但对于 CDN 观众来说，他们只需要一路直播流，所以您需要将多路音视频流混成一路标准的直播流，这就需要混流转码。

当您调用 `setMixTranscodingConfig()` 接口时，SDK 会向腾讯云的转码服务器发送一条指令，目的是将房间里的多路音视频流混合为一，您可以通过 `mixStreams` 参数来调整每一路画面的位置，以及是否只混合声音，也可以通过 `videoWidth`、`videoHeight`、`videoBitrate` 等参数控制混合音视频流的编码参数。

```
【画面1】=> 解码====> \\
\\
【画面2】=> 解码=> 画面混合=> 编码=> 【混合后的画面】
/
【画面3】=> 解码====> /

【声音1】=> 解码====> \\
\\
【声音2】=> 解码=> 声音混合=> 编码=> 【混合后的声音】
/
【声音3】=> 解码====> /
```

参考文档：[云端混流转码](#)。

参数	描述
config	请参考 V2TXLiveDef.java 中关于 V2TXLiveTranscodingConfig 的介绍。如果传入 null 则取消云端混流转码。

注意

关于云端混流的注意事项：

仅支持 RTC 模式混流。

云端转码会引入一定的 CDN 观看延时，大概会增加1 - 2秒。

调用该函数的用户，会将连麦中的多路画面混合到自己当前这路画面或者 config 中指定的 streamId 上。

请注意，若您还在房间中且不再需要混流，请务必传入 null 进行取消，因为当您发起混流后，云端混流模块就会开始工作，不及时取消混流可能会引起不必要的计费损失。

请放心，您退房时会自动取消混流状态。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 未开启推流时，不允许设置混流转码参数。

startLocalRecording

startLocalRecording

int startLocalRecording	(V2TXLiveLocalRecordingParams params)
-------------------------	--

开始录制音视频流

注意

推流开启后才能开始录制，非推流状态下开启录制无效。

录制过程中不要动态切换分辨率和软/硬剪辑，生成的视频极有可能出现异常。

返回值说明：

返回值 [V2TXLiveCode](#)。

`V2TXLIVE_OK` : 成功。

`V2TXLIVE_ERROR_INVALID_PARAMETER` : 参数不合法，比如filePath 为空。

`V2TXLIVE_ERROR_REFUSED` : API被拒绝，推流尚未开始。

stopLocalRecording

stopLocalRecording

停止录制音视频流

注意

当停止推流后，如果视频还在录制中，SDK 内部会自动结束录制。

enableVoiceActivityDetection

enableVoiceActivityDetection

void enableVoiceActivityDetection	(boolean enable)
-----------------------------------	------------------

启用人声检测

注意

开启后可以在 `OnVoiceActivityDetectionUpdate` 回调中获取到语音活动的开始与停止

V2TXLivePusherObserver

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePusherObserver @ TXLiteAVSDK

Function: 腾讯云直播推流的回调通知

功能

腾讯云直播的推流回调通知。

介绍

可以接收 [V2TXLivePusher](#) 推流器的一些推流通知，包括推流器连接状态、音视频首帧回调、统计数据、警告和错误信息等。

V2TXLivePusherObserver

V2TXLivePusherObserver

函数列表	描述
onError	直播推流器错误通知，推流器出现错误时，会回调该通知
onWarning	直播推流器警告通知
onCaptureFirstAudioFrame	首帧音频采集完成的回调通知
onCaptureFirstVideoFrame	首帧视频采集完成的回调通知
onMicrophoneVolumeUpdate	麦克风采集音量值回调
onPushStatusUpdate	推流器连接状态回调通知
onStatisticsUpdate	直播推流器统计数据回调

onSnapshotComplete	截图回调
onGLContextCreated	SDK 内部的 OpenGL 环境的创建通知
onProcessAudioFrame	本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调
onProcessVideoFrame	自定义视频处理回调
onGLContextDestroyed	SDK 内部的 OpenGL 环境的销毁通知
onSetMixTranscodingConfig	设置云端的混流转码参数的回调，对应于 setMixTranscodingConfig 接口
onScreenCaptureStarted	当屏幕分享开始时，SDK 会通过此回调通知
onScreenCaptureStopped	当屏幕分享停止时，SDK 会通过此回调通知
onLocalRecordBegin	录制任务开始的事件回调
onLocalRecording	录制任务正在进行中的进展事件回调
onLocalRecordComplete	录制任务已经结束的事件回调
onVoiceActivityDetectionUpdate	调用 enableVoiceActivityDetection 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

onError

onError

<code>void onError</code>	(int code
	String msg
	Bundle extraInfo)

直播推流器错误通知，推流器出现错误时，会回调该通知

参数	描述
code	错误码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	错误信息。

onWarning

onWarning

void onWarning	(int code
	String msg
	Bundle extraInfo)

直播推流器警告通知

参数	描述
code	警告码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	警告信息。

onCaptureFirstAudioFrame

onCaptureFirstAudioFrame

首帧音频采集完成的回调通知

onCaptureFirstVideoFrame

onCaptureFirstVideoFrame

首帧视频采集完成的回调通知

onMicrophoneVolumeUpdate

onMicrophoneVolumeUpdate

void onMicrophoneVolumeUpdate	(int volume)
-------------------------------	--------------

麦克风采集音量值回调

参数	描述
----	----

volume	音量大小。
--------	-------

注意

调用 [enableVolumeEvaluation](#) 开启采集音量大小提示之后，会收到这个回调通知。

onPushStatusUpdate

onPushStatusUpdate

void onPushStatusUpdate	(V2TXLivePushStatus status
	String msg
	Bundle extraInfo)

推流器连接状态回调通知

参数	描述
extraInfo	扩展信息。
msg	连接状态信息。
status	推流器连接状态 V2TXLivePushStatus 。

onStatisticsUpdate

onStatisticsUpdate

void onStatisticsUpdate	(V2TXLivePusherStatistics statistics)
-------------------------	--

直播推流器统计数据回调

参数	描述
statistics	推流器统计数据 V2TXLivePusherStatistics

onSnapshotComplete

onSnapshotComplete

void onSnapshotComplete	(Bitmap image)
-------------------------	----------------

截图回调

参数	描述
image	已截取的视频画面。

注意

调用 [snapshot](#) 截图之后，会收到这个回调通知。

onGLContextCreated

onGLContextCreated

SDK 内部的 OpenGL 环境的创建通知

onProcessAudioFrame

onProcessAudioFrame

void onProcessAudioFrame	(V2TXLiveDef.V2TXLiveAudioFrame frame)
--------------------------	--

本地采集并经过音频模块前处理、音效处理和混 BGM 后的音频数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把刚采集到并经过前处理、音效处理和混 BGM 之后的数据，在最终进行网络编码之前，以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长固定为 0.02s，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 $\text{采样率} \times \text{时间帧长} \times \text{声道数} \times \text{采样点位宽}$ 。

以 SDK 默认的音频录制格式 48000 采样率、单声道、16 采样点位宽为例，字节帧长为 $48000 \times 0.02\text{s} \times 1 \times 16\text{bit} = 15360\text{bit} = 1920\text{字节}$ 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

onProcessVideoFrame

onProcessVideoFrame

int onProcessVideoFrame	(V2TXLiveVideoFrame srcFrame
	V2TXLiveVideoFrame dstFrame)

自定义视频处理回调

参数	描述
dstFrame	用于承载处理过的视频画面。
srcFrame	用于承载未处理的视频画面。

注意

需要调用 [enableCustomVideoProcess](#) 开启自定义视频处理，才会收到这个回调通知。

【情况一】美颜组件会产生新的纹理

如果您使用的美颜组件会在处理图像的过程中产生一帧全新的纹理（用于承载处理后的图像），那请您在回调函数中将 `dstFrame.textureId` 设置为新纹理的 ID。

```
java
@Override
public void onGLContextCreated() {
    mFURenderer.onSurfaceCreated();
    mFURenderer.setUseTexAsync(true);
}
@Override
public int onProcessVideoFrame(V2TXLiveVideoFrame srcFrame, V2TXLiveVideoFrame dstFrame) {
    dstFrame.texture.textureId = mFURenderer.onDrawFrameSingleInput(
        srcFrame.texture.textureId, srcFrame.width, srcFrame.height);
    return 0;
}
@Override
public void onGLContextDestroyed() {
```

```
mFURenderer.onSurfaceDestroyed();  
}
```

【情况二】美颜组件并不自身产生新纹理

如果您使用的第三方美颜模块并不生成新的纹理，而是需要您设置给该模块一个输入纹理和一个输出纹理，则可以考虑如下方案：

```
    `java  
    int onProcessVideoFrame(V2TXLiveVideoFrame srcFrame, V2TXLiveVideoFrame dstFrame) {  
        thirdparty_process(srcFrame.texture.textureId, srcFrame.width, srcFrame.height, dstFrame.texture.textureId);  
        return 0;  
    }
```

onGLContextDestroyed

onGLContextDestroyed

SDK 内部的 OpenGL 环境的销毁通知

onSetMixTranscodingConfig

onSetMixTranscodingConfig

void onSetMixTranscodingConfig	(int code
	String msg)

设置云端的混流转码参数的回调，对应于 {@link setMixTranscodingConfig} 接口

参数	描述
code	0表示成功，其余值表示失败。
msg	具体错误原因。

onScreenCaptureStarted

onScreenCaptureStarted

当屏幕分享开始时，SDK 会通过此回调通知

onScreenCaptureStopped

onScreenCaptureStopped

void onScreenCaptureStopped	(int reason)
-----------------------------	--------------

当屏幕分享停止时，SDK 会通过此回调通知

参数	描述
reason	停止原因 0：表示用户主动停止。 1：iOS 表示录屏被系统中断；Mac、Windows 表示屏幕分享窗口被关闭。 2：Windows 表示屏幕分享的显示屏状态变更（如接口被拔出、投影模式变更等）；其他平台不抛出。

onLocalRecordBegin

onLocalRecordBegin

void onLocalRecordBegin	(int code
	String storagePath)

录制任务开始的事件回调

参数	描述
code	状态码。 0：录制任务启动成功。 -1：内部错误导致录制任务启动失败。 -2：文件后缀名有误（比如不支持的录制格式）。 -6：录制已经启动，需要先停止录制。 -7：录制文件已存在，需要先删除文件。 -8：录制目录无写入权限，请检查目录权限问题。
storagePath	录制的文件地址。

onLocalRecording

onLocalRecording

void onLocalRecording	(long durationMs
	String storagePath)

录制任务正在进行的进展事件回调

参数	描述
durationMs	录制时长。
storagePath	录制的文件地址。

onLocalRecordComplete

onLocalRecordComplete

void onLocalRecordComplete	(int code
	String storagePath)

录制任务已经结束的事件回调

参数	描述
code	状态码。 0：结束录制任务成功。 -1：录制失败。 -2：切换分辨率或横竖屏导致录制结束。 -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
storagePath	录制的文件地址。

onVoiceActivityDetectionUpdate

onVoiceActivityDetectionUpdate

void onVoiceActivityDetectionUpdate	(boolean active)
-------------------------------------	------------------

调用 `{@link enableVoiceActivityDetection}` 开启人声检测之后，当主播开始或结束说话时，会收到这个回调通知。

参数	描述
active	人声开始或停止。

V2TXLivePlayer

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePlayer @ TXLiteAVSDK

Function: 腾讯云直播播放器

功能

腾讯云直播播放器。
主要负责从指定的直播流地址拉取音视频数据，并进行解码和本地渲染播放。

介绍

播放器包含如下能力：

- 支持 RTMP、HTTP-FLV、HLS、TRTC、WebRTC 协议。
- 屏幕截图，可以截取当前直播流的视频画面。
- 延时调节，可以设置播放器缓存自动调整的最小和最大时间。
- 自定义的视频数据处理，您可以根据项目需要处理直播流中的视频数据后，再进行渲染以及播放。

V2TXLivePlayer

V2TXLivePlayer

函数列表	描述
setObserver	设置播放器回调
setRenderView	设置播放器的视频渲染 View，该控件负责显示视频内容
setRenderView	设置播放器的视频渲染 View。该控件负责显示视频内容
setRenderView	设置播放器的视频渲染 View。该控件负责显示视频内容

setRenderRotation	设置播放器画面的旋转角度
setRenderFillMode	设置画面的填充模式
startLivePlay	开始播放音视频流
stopPlay	停止播放音视频流
isPlaying	播放器是否正在播放中
pauseAudio	暂停播放器的音频流
resumeAudio	恢复播放器的音频流
pauseVideo	暂停播放器的视频流
resumeVideo	恢复播放器的视频流
setPlayoutVolume	设置播放器音量
setCacheParams	设置播放器缓存自动调整的最小和最大时间 (单位：秒)
switchStream	直播流无缝切换，支持 FLV 和 LEB
getStreamList	获取码流信息
enableVolumeEvaluation	启用播放音量大小提示
snapshot	截取播放过程中的视频画面
enableObserveVideoFrame	开启/关闭对视频帧的监听回调
enableObserveAudioFrame	开启/关闭对音频数据的监听回调
enableReceiveSeiMessage	开启接收 SEI 消息
showDebugView	是否显示播放器状态信息的调试浮层
setProperty	调用 V2TXLivePlayer 的高级 API 接口
startLocalRecording	开始录制音视频流
stopLocalRecording	停止录制音视频流

setObserver

setObserver

--	--

```
void setObserver      (V2TXLivePlayerObserver observer)
```

设置播放器回调

通过设置回调，可以监听 V2TXLivePlayer 播放器的一些回调事件，包括播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等。

参数	描述
observer	播放器的回调目标对象，更多信息请查看 V2TXLivePlayerObserver

setRenderView

setRenderView

```
int setRenderView      (TXCloudVideoView view)
```

设置播放器的视频渲染 View，该控件负责显示视频内容

参数	描述
view	播放器渲染 View。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK：成功。

setRenderView

setRenderView

```
int setRenderView      (TextureView view)
```

设置播放器的视频渲染 View。 该控件负责显示视频内容

参数	描述
view	播放器渲染 View。

返回值说明：

返回值 [V2TXLiveCode](#)。
V2TXLIVE_OK：成功。

setRenderView

setRenderView

int setRenderView	(SurfaceView view)
-------------------	--------------------

设置播放器的视频渲染 View。该控件负责显示视频内容

参数	描述
view	播放器渲染 View。

返回值说明：
返回值 [V2TXLiveCode](#)。
V2TXLIVE_OK：成功。

setRenderRotation

setRenderRotation

int setRenderRotation	(V2TXLiveRotation rotation)
-----------------------	--

设置播放器画面的旋转角度

参数	描述
rotation	旋转角度 V2TXLiveRotation 。 V2TXLiveRotation0【默认值】：0度, 不旋转。 V2TXLiveRotation90: 顺时针旋转90度。 V2TXLiveRotation180: 顺时针旋转180度。 V2TXLiveRotation270: 顺时针旋转270度。

返回值说明：
返回值 [V2TXLiveCode](#)。
V2TXLIVE_OK: 成功。

setRenderFillMode

setRenderFillMode

int setRenderFillMode	(V2TXLiveFillMode mode)
-----------------------	-------------------------

设置画面的填充模式

参数	描述
mode	画面填充模式 V2TXLiveFillMode 。 V2TXLiveFillModeFill 【默认值】：图像铺满屏幕，不留黑边，如果图像宽高比不同于屏幕宽高比，部分画面内容会被裁剪掉。 V2TXLiveFillModeFit: 图像适应屏幕，保持画面完整，但如果图像宽高比不同于屏幕宽高比，会有黑边的存在。 V2TXLiveFillModeScaleFill: 图像拉伸铺满，因此长度和宽度可能不会按比例变化。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

startLivePlay

startLivePlay

int startLivePlay	(String url)
-------------------	--------------

开始播放音视频流

参数	描述
url	音视频流的播放地址，支持 RTMP，HTTP-FLV，TRTC。

注意

10.7 版本开始，需要通过 [setLicence](#) 或者 [setLicence](#) 设置 Licence 后方可成功播放，否则将播放失败（黑屏），全局仅设置一次即可。直播 Licence、短视频 Licence 和视频播放 Licence 均可使用，若您暂未获取上述 Licence，可[快速免费申请测试版 Licence](#) 以正常播放，正式版 Licence 需[购买](#)。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 操作成功，开始连接并播放。

V2TXLIVE_ERROR_INVALID_PARAMETER：操作失败，url 不合法。

V2TXLIVE_ERROR_REFUSED：RTC 不支持同一设备上同时推拉同一个 StreamId。

V2TXLIVE_ERROR_INVALID_LICENSE：licence 不合法，播放失败。

stopPlay

stopPlay

停止播放音视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

isPlaying

isPlaying

播放器是否正在播放中

返回值说明：

是否正在播放。

1: 正在播放中。

0: 已经停止播放。

pauseAudio

pauseAudio

暂停播放器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeAudio

resumeAudio

恢复播放器的音频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

pauseVideo

pauseVideo

暂停播放器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

resumeVideo

resumeVideo

恢复播放器的视频流

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setPlayoutVolume

setPlayoutVolume

int setPlayoutVolume	(int volume)
----------------------	--------------

设置播放器音量

参数	描述
volume	音量大小，取值范围0 - 100。【默认值】：100。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

setCacheParams

setCacheParams

int setCacheParams	(float minTime
	float maxTime)

设置播放器缓存自动调整的最小和最大时间 (单位：秒)

参数	描述
maxTime	缓存自动调整的最大时间，取值需要大于0。【默认值】：5。
minTime	缓存自动调整的最小时间，取值需要大于0。【默认值】：1。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，minTime 和 maxTime 需要大于0。

V2TXLIVE_ERROR_REFUSED: 播放器处于播放状态，不支持修改缓存策略。

switchStream

switchStream

int switchStream	(String newUrl)
------------------	-----------------

直播流无缝切换，支持 FLV 和 LEB

参数	描述
newUrl	新的拉流地址。

getStreamList

getStreamList

获取码流信息

enableVolumeEvaluation

enableVolumeEvaluation

int enableVolumeEvaluation	(int intervalMs)
----------------------------	------------------

启用播放音量大小提示

开启后可以在 [V2TXLivePlayerObserver](#) 回调中获取到 SDK 对音量大小值的评估。

参数	描述
intervalMs	决定了 onPlayoutVolumeUpdate 回调的触发间隔，单位为ms，最小间隔为100ms，如果小于等于0则会关闭回调，建议设置为300ms；【默认值】：0，不开启。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

snapshot

snapshot

截取播放过程中的视频画面

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_REFUSED: 播放器处于停止状态，不允许调用截图操作。

enableObserveVideoFrame

enableObserveVideoFrame

int enableObserveVideoFrame	(boolean enable

	V2TXLivePixelFormat pixelFormat
	V2TXLiveBufferType bufferType)

开启/关闭对视频帧的监听回调

SDK 在您开启此开关后将不再渲染视频画面，您可以通过 [V2TXLivePlayerObserver](#) 获得视频帧，并执行自定义的渲染逻辑。

参数	描述
bufferType	自定义渲染回调的视频数据格式 V2TXLiveBufferType 。
enable	是否开启自定义渲染。【默认值】：false。
pixelFormat	自定义渲染回调的视频像素格式 V2TXLivePixelFormat 。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_NOT_SUPPORTED: 像素格式或者数据格式不支持。

enableObserveAudioFrame

enableObserveAudioFrame

int enableObserveAudioFrame	(boolean enable)
-----------------------------	------------------

开启/关闭对音频数据的监听回调

如果您开启此开关，您可以通过 [V2TXLivePlayerObserver](#) 获得音频数据，并执行自定义的逻辑。

参数	描述
enable	是否开启音频数据回调。【默认值】：false。

返回值说明：

返回值 [V2TXLiveCode](#)

V2TXLIVE_OK: 成功

enableReceiveSeiMessage

enableReceiveSeiMessage

int enableReceiveSeiMessage	(boolean enable
	int payloadType)

开启接收 SEI 消息

参数	描述
enable	true: 开启接收 SEI 消息; false: 关闭接收 SEI 消息。【默认值】: false。
payloadType	指定接收 SEI 消息的 payloadType, 支持 5、242、243, 请与发送端的 payloadType 保持一致。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

showDebugView

showDebugView

void showDebugView	(boolean isShow)
--------------------	------------------

是否显示播放器状态信息的调试浮层

参数	描述
isShow	是否显示。【默认值】: false。

setProperty

setProperty

int setProperty	(String key
	Object value)

调用 V2TXLivePlayer 的高级 API 接口

参数	描述
----	----

key	高级 API 对应的 key。
value	调用 key 所对应的高级 API 时，需要的参数。

注意

该接口用于调用一些高级功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，key 不允许为 null。

startLocalRecording

startLocalRecording

int startLocalRecording	(V2TXLiveLocalRecordingParams params)
-------------------------	--

开始录制音视频流

注意

拉流开启后才能开始录制，非拉流状态下开启录制无效。

录制过程中不要动态切换软/硬解，生成的视频极有可能出现异常。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK : 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER : 参数不合法，比如filePath 为空。

V2TXLIVE_ERROR_REFUSED : API被拒绝，拉流尚未开始。

stopLocalRecording

stopLocalRecording

停止录制音视频流

注意

当停止拉流后，如果视频还在录制中，SDK 内部会自动结束录制。

V2TXLivePlayerObserver

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePlayerObserver @ TXLiteAVSDK

Function: 腾讯云直播的播放器回调通知

功能

腾讯云直播的播放器回调通知。

介绍

可以接收 [V2TXLivePlayer](#) 播放器的一些回调通知，包括播放器状态、播放音量回调、音视频首帧回调、统计数据、警告和错误信息等。

V2TXLivePlayerObserver

V2TXLivePlayerObserver

函数列表	描述
onError	直播播放器错误通知，播放器出现错误时，会回调该通知
onWarning	直播播放器警告通知
onVideoResolutionChanged	直播播放器分辨率变化通知
onConnected	已经成功连接到服务器
onVideoPlaying	视频播放事件
onAudioPlaying	音频播放事件
onVideoLoading	视频加载事件

<code>onAudioLoading</code>	音频加载事件
<code>onPlayoutVolumeUpdate</code>	播放器音量大小回调
<code>onStatisticsUpdate</code>	直播播放器统计数据回调
<code>onSnapshotComplete</code>	截图回调
<code>onRenderVideoFrame</code>	自定义视频渲染回调
<code>onPlayoutAudioFrame</code>	音频数据回调
<code>onReceiveSeiMessage</code>	收到 SEI 消息的回调，发送端通过 <code>V2TXLivePusher</code> 中的 <code>sendSeiMessage</code> 来发送 SEI 消息
<code>onStreamSwitched</code>	分辨率无缝切换回调
<code>onLocalRecordBegin</code>	录制任务开始的事件回调
<code>onLocalRecording</code>	录制任务正在进行中的进展事件回调
<code>onLocalRecordComplete</code>	录制任务已经结束的事件回调

onError

onError

<code>void onError</code>	(<code>V2TXLivePlayer</code> player
	int code
	String msg
	Bundle extraInfo)

直播播放器错误通知，播放器出现错误时，会回调该通知

参数	描述
code	错误码 <code>V2TXLiveCode</code> 。
extraInfo	扩展信息。
msg	错误信息。
player	回调该通知的播放器对象。

onWarning

onWarning

void onWarning	(V2TXLivePlayer player
	int code
	String msg
	Bundle extraInfo)

直播播放器警告通知

参数	描述
code	警告码 V2TXLiveCode 。
extraInfo	扩展信息。
msg	警告信息。
player	回调该通知的播放器对象。

onVideoResolutionChanged

onVideoResolutionChanged

void onVideoResolutionChanged	(V2TXLivePlayer player
	int width
	int height)

直播播放器分辨率变化通知

参数	描述
height	视频高。
player	回调该通知的播放器对象。
width	视频宽。

onConnected

onConnected

void onConnected	(V2TXLivePlayer player
	Bundle extraInfo)

已经成功连接到服务器

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。

onVideoPlaying

onVideoPlaying

void onVideoPlaying	(V2TXLivePlayer player
	boolean firstPlay
	Bundle extraInfo)

视频播放事件

参数	描述
extraInfo	扩展信息。
firstPlay	第一次播放标志。
player	回调该通知的播放器对象。

onAudioPlaying

onAudioPlaying

void onAudioPlaying	(V2TXLivePlayer player
	boolean firstPlay

	Bundle extraInfo)
--	-------------------

音频播放事件

参数	描述
extraInfo	扩展信息。
firstPlay	第一次播放标志。
player	回调该通知的播放器对象。

onVideoLoading

onVideoLoading

void onVideoLoading	(V2TXLivePlayer player
	Bundle extraInfo)

视频加载事件

参数	描述
extraInfo	扩展信息。
player	回调该通知的播放器对象。

onAudioLoading

onAudioLoading

void onAudioLoading	(V2TXLivePlayer player
	Bundle extraInfo)

音频加载事件

参数	描述
extraInfo	扩展信息。

player

回调该通知的播放器对象。

onPlayoutVolumeUpdate

onPlayoutVolumeUpdate

void onPlayoutVolumeUpdate	(V2TXLivePlayer player
	int volume)

播放器音量大小回调

参数	描述
player	回调该通知的播放器对象。
volume	音量大小。

注意

调用 [enableVolumeEvaluation](#) 开启播放音量大小提示之后，会收到这个回调通知。

onStatisticsUpdate

onStatisticsUpdate

void onStatisticsUpdate	(V2TXLivePlayer player
	V2TXLivePlayerStatistics statistics)

直播播放器统计数据回调

参数	描述
player	回调该通知的播放器对象。
statistics	播放器统计数据 V2TXLivePlayerStatistics 。

onSnapshotComplete

onSnapshotComplete

--	--

void onSnapshotComplete	(V2TXLivePlayer player
	Bitmap image)

截图回调

参数	描述
image	已截取的视频画面。
player	回调该通知的播放器对象。

onRenderVideoFrame

onRenderVideoFrame

void onRenderVideoFrame	(V2TXLivePlayer player
	V2TXLiveVideoFrame videoFrame)

自定义视频渲染回调

参数	描述
player	回调该通知的播放器对象。
videoFrame	视频帧数据 V2TXLiveVideoFrame 。

注意

需要您调用 [enableObserveVideoFrame](#) 开启回调开关。

onPlayoutAudioFrame

onPlayoutAudioFrame

void onPlayoutAudioFrame	(V2TXLivePlayer player
	V2TXLiveAudioFrame audioFrame)

音频数据回调

参数	描述
----	----

audioFrame	音频帧数据 V2TXLiveAudioFrame 。
player	回调该通知的播放器对象。

注意

需要您调用 [enableObserveAudioFrame](#) 开启回调开关。请在当前回调中使用 audioFrame 的 data。

onReceiveSeiMessage

onReceiveSeiMessage

void onReceiveSeiMessage	(V2TXLivePlayer player
	int payloadType
	byte[] data)

收到 SEI 消息的回调，发送端通过 {@link [V2TXLivePusher](#)} 中的 `sendSeiMessage` 来发送 SEI 消息

参数	描述
data	数据。
payloadType	回调数据的SEI payloadType。
player	回调该通知的播放器对象。

注意

调用 [V2TXLivePlayer](#) 中的 `enableReceiveSeiMessage` 开启接收 SEI 消息之后，会收到这个回调通知。

onStreamSwitched

onStreamSwitched

void onStreamSwitched	(V2TXLivePlayer player
	String url
	int code)

分辨率无缝切换回调

参数	描述
code	状态码，0：成功，-1：切换超时，-2：切换失败，服务端错误，-3：切换失败，客户端错误。
player	回调该通知的播放器对象。
url	切换的播放地址。

注意

调用 `V2TXLivePlayer` 中的 `switchStream` 切换分辨率，会收到这个回调通知。

onLocalRecordBegin

onLocalRecordBegin

void onLocalRecordBegin	(<code>V2TXLivePlayer</code> player
	int code
	String storagePath)

录制任务开始的事件回调

参数	描述
code	状态码。 0：录制任务启动成功。 -1：内部错误导致录制任务启动失败。 -2：文件后缀名有误（比如不支持的录制格式）。 -6：录制已经启动，需要先停止录制。 -7：录制文件已存在，需要先删除文件。 -8：录制目录无写入权限，请检查目录权限问题。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

onLocalRecording

onLocalRecording

void onLocalRecording	(<code>V2TXLivePlayer</code> player

	long durationMs
	String storagePath)

录制任务正在进行的进展事件回调

参数	描述
durationMs	录制时长。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

onLocalRecordComplete

onLocalRecordComplete

void onLocalRecordComplete	(V2TXLivePlayer player
	int code
	String storagePath)

录制任务已经结束的事件回调

参数	描述
code	状态码。 0：结束录制任务成功。 -1：录制失败。 -2：切换分辨率或横竖屏导致录制结束。 -3：录制时间太短，或未采集到任何视频或音频数据，请检查录制时长，或是否已开启音、视频采集。
player	回调该通知的播放器对象。
storagePath	录制的文件地址。

V2TXLivePremier

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLivePremier @ TXLiteAVSDK

Function: V2TXLive 高级接口

V2TXLivePremier

V2TXLivePremierObserver

函数列表	描述
onLog	自定义 Log 输出回调接口
onLicenceLoaded	setLicence 接口回调
onCaptureAudioFrame	本地麦克风采集到的音频数据回调
onPlayoutAudioFrame	将各路待播放音频混合之后并在最终提交系统播放之前的数据回调
onVoiceEarMonitorAudioFrame	耳返的音频数据

V2TXLivePremier

函数列表	描述
getSDKVersionStr	获取 SDK 版本号
setObserver	设置 V2TXLivePremier 回调接口
setLogConfig	设置 Log 的配置信息
setEnvironment	设置 SDK 接入环境
setLicence	设置 SDK 的授权 License
setSocks5Proxy	设置 SDK socks5 代理配置

enableAudioCaptureObserver	开启/关闭对音频采集数据的监听回调（可读写）
enableAudioPlayoutObserver	开启/关闭对最终系统要播放出的音频数据的监听回调
enableVoiceEarMonitorObserver	开启/关闭耳返音频数据的监听回调
setUserId	设置 <code>userId</code>
callExperimentalAPI	调用实验性 API 接口

onLog

onLog

<code>void onLog</code>	(int level
	String log)

自定义 Log 输出回调接口

onLicenceLoaded

onLicenceLoaded

<code>void onLicenceLoaded</code>	(int result
	String reason)

setLicence 接口回调

参数	描述
<code>reason</code>	设置 <code>licence</code> 失败原因。
<code>result</code>	设置 <code>licence</code> 结果 0 成功，负数失败。

onCaptureAudioFrame

onCaptureAudioFrame

<code>void onCaptureAudioFrame</code>	(V2TXLiveDef. V2TXLiveAudioFrame frame)
---------------------------------------	---

本地麦克风采集到的音频数据回调

参数	描述
frame	音频数据。

注意

请不要在此回调函数中做任何耗时操作，建议直接拷贝到另一线程进行处理，否则会导致各种声音问题。

此接口回调出的音频数据支持修改。

此接口回调出的音频时间帧长固定为0.02s。

由时间帧长转化为字节帧长的公式为【采样率 × 时间帧长 × 声道数 × 采样点位宽】。

以SDK默认的音频录制格式48000采样率、单声道、16采样点位宽为例，字节帧长为【48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节】。

此接口回调出的音频数据**不包含**背景音、音效、混响等前处理效果，延迟极低。

需要您调用 [enableAudioCaptureObserver](#) 开启回调开关。

onPlayoutAudioFrame

onPlayoutAudioFrame

void onPlayoutAudioFrame	(V2TXLiveDef.V2TXLiveAudioFrame frame)
--------------------------	--

将各路待播放音频混合之后并在最终提交系统播放之前的数据回调

当您设置完音频数据自定义回调之后，SDK 内部会把各路待播放的音频混合之后的音频数据，在提交系统播放之前，以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长固定为 0.02s，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 采样率 × 时间帧长 × 声道数 × 采样点位宽 。

以 SDK 默认的音频录制格式 48000 采样率、单声道、16 采样点位宽为例，字节帧长为 48000 × 0.02s × 1 × 16bit = 15360bit = 1920字节 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，由于 SDK 每隔 20ms 就要处理一帧音频数据，如果您的处理时间超过 20ms，就会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。
3. 此接口回调出的是对各路待播放音频数据的混合，但其中并不包含耳返的音频数据。

onVoiceEarMonitorAudioFrame

onVoiceEarMonitorAudioFrame

void onVoiceEarMonitorAudioFrame	(V2TXLiveDef.V2TXLiveAudioFrame frame)
----------------------------------	--

耳返的音频数据

当您设置完音频数据自定义回调之后，SDK 内部会把耳返的音频数据在播放之前以 PCM 格式的形式通过本接口回调给您。

此接口回调出的音频时间帧长不固定，格式为 PCM 格式。

由时间帧长转化为字节帧长的公式为 $\text{采样率} \times \text{时间帧长} \times \text{声道数} \times \text{采样点位宽}$ 。

以 TRTC 默认的音频录制格式 48000 采样率、单声道、16 采样点位宽为例，0.02s 的音频数据字节帧长为

$48000 \times 0.02\text{s} \times 1 \times 16\text{bit} = 15360\text{bit} = 1920\text{字节}$ 。

参数	描述
frame	PCM 格式的音频数据帧。

注意

1. 请不要在此回调函数中做任何耗时操作，否则会导致声音异常。
2. 此接口回调出的音频数据是可读写的，也就是说您可以在回调函数中同步修改音频数据，但请保证处理耗时。

getSDKVersionStr

getSDKVersionStr

获取 SDK 版本号

setObserver

setObserver

void setObserver	(V2TXLivePremierObserver observer)
------------------	------------------------------------

设置 V2TXLivePremier 回调接口

setLogConfig

setLogConfig

void setLogConfig	(V2TXLiveDef.V2TXLiveLogConfig config)
-------------------	--

设置 Log 的配置信息

setEnvironment

setEnvironment

void setEnvironment	(String env)
---------------------	--------------

设置 SDK 接入环境

参数	描述
env	目前支持 “default” 和 “GDPR” 两个参数。 default：默认环境，SDK 会在全球寻找最佳接入点进行接入。 GDPR：所有音视频数据和质量统计数据都不会经过中国大陆地区的服务器。

注意

如您的应用无特殊需求，请不要调用此接口进行设置。

setLicence

setLicence

void setLicence	(Context context
	String url
	String key)

设置 SDK 的授权 License

文档地址：<https://cloud.tencent.com/document/product/454/34750>。

参数	描述
context	
key	licence的密钥。

url	licence的地址。
-----	-------------

setSocks5Proxy

setSocks5Proxy

void setSocks5Proxy	(String host
	int port
	String username
	String password
	V2TXLiveDef.V2TXLiveSocks5ProxyConfig config)

设置 SDK socks5 代理配置

参数	描述
config	配置使用 socks5 代理服务器的协议。
host	socks5 代理服务器的地址。
password	socks5 代理服务器的验证的密码。
port	socks5 代理服务器的端口。
username	socks5 代理服务器的验证的用户名。

enableAudioCaptureObserver

enableAudioCaptureObserver

void enableAudioCaptureObserver	(boolean enable
	V2TXLiveDef.V2TXLiveAudioFrameObserverFormat format)

开启/关闭对音频采集数据的监听回调（可读写）

参数	描述
enable	是否开启。【默认值】：false。

format	设置回调出的 AudioFrame 的格式。
--------	------------------------

注意

需要在 [startPush](#) 之前调用，才会生效。

enableAudioPlayoutObserver

enableAudioPlayoutObserver

void enableAudioPlayoutObserver	(boolean enable
	V2TXLiveDef.V2TXLiveAudioFrameObserverFormat format)

开启/关闭对最终系统要播放出的音频数据的监听回调

参数	描述
enable	是否开启。【默认值】：false。
format	设置回调出的 AudioFrame 的格式。

enableVoiceEarMonitorObserver

enableVoiceEarMonitorObserver

void enableVoiceEarMonitorObserver	(boolean enable)
------------------------------------	------------------

开启/关闭耳返音频数据的监听回调

参数	描述
enable	是否开启。【默认值】：false。

setUserId

setUserId

void setUserId	(String userId)
----------------	-----------------

设置 userId

参数	描述
userId	业务侧自身维护的用户/设备id。

callExperimentalAPI

callExperimentalAPI

int callExperimentalAPI	(String jsonStr)
-------------------------	------------------

调用实验性 API 接口

参数	描述
jsonStr	接口及参数描述的 JSON 字符串。

注意

该接口用于调用一些实验性功能。

返回值说明：

返回值 [V2TXLiveCode](#)。

V2TXLIVE_OK: 成功。

V2TXLIVE_ERROR_INVALID_PARAMETER: 操作失败，参数非法。

TXAudioEffectManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 背景音乐、短音效和人声特效的管理类

Function: 用于对背景音乐、短音效和人声特效进行设置的管理类

TXAudioEffectManager

TXMusicPreloadObserver

函数列表	描述
onLoadProgress	背景音乐预加载进度
onLoadError	背景音乐预加载出错

TXMusicPlayObserver

函数列表	描述
onStart	背景音乐开始播放
onPlayProgress	背景音乐的播放进度
onComplete	背景音乐已经播放完毕

TXAudioEffectManager

函数列表	描述
enableVoiceEarMonitor	开启耳返
setVoiceEarMonitorVolume	设置耳返音量

setVoiceReverbType	设置人声的混响效果
setVoiceChangerType	设置人声的变声特效
setVoiceCaptureVolume	设置语音音量
setVoicePitch	设置语音音调
setMusicObserver	设置背景音乐的事件回调接口
startPlayMusic	开始播放背景音乐
stopPlayMusic	停止播放背景音乐
pausePlayMusic	暂停播放背景音乐
resumePlayMusic	恢复播放背景音乐
setAllMusicVolume	设置所有背景音乐的本地音量和远端音量的大小
setMusicPublishVolume	设置某一首背景音乐的远端音量的大小
setMusicPlayoutVolume	设置某一首背景音乐的本地音量的大小
setMusicPitch	调整背景音乐的音调高低
setMusicSpeedRate	调整背景音乐的变速效果
getMusicCurrentPosInMS	获取背景音乐的播放进度（单位：毫秒）
getMusicDurationInMS	获取背景音乐的总时长（单位：毫秒）
seekMusicToPosInMS	设置背景音乐的播放进度（单位：毫秒）
setMusicScratchSpeedRate	调整搓碟的变速效果
setPreloadObserver	设置预加载事件回调
preloadMusic	预加载背景音乐
getMusicTrackCount	获取背景音乐的音轨数量
setMusicTrack	指定背景音乐的播放音轨

结构体类型

函数列表	描述
------	----

[AudioMusicParam](#)

背景音乐的播放控制信息

枚举类型

枚举类型	描述
TXVoiceReverbType	混响特效
TXVoiceChangerType	变声特效

onLoadProgress

onLoadProgress

void onLoadProgress	(int id
	int progress)

背景音乐预加载进度

onLoadError

onLoadError

void onLoadError	(int id
	int errorCode)

背景音乐预加载出错

参数	描述
errorCode	-4001:打开文件失败，如音频数据无效，FFMPEG 协议未找到等；-4002:解码失败，如音频文件损坏，网络音频文件服务器无法访问等；-4003:预加载数量超上限，请先调用 <code>stopPlayMusic</code> 释放无用的预加载；-4005:非法路径导致打开文件失败，请检查您传入的路径参数是否指向一个合法的音乐文件；-4006:非法URL导致打开文件失败，请用浏览器检查您传入的 URL 地址是否可以下载到期望的音乐文件；-4007:无音频流导致打开文件失败，请确认您传入的文件是否是合法的音频文件，以及文件是否被损坏；-4008:格式不支持导致打开文件失败，请确认您传入的文件格式是否是支持的文件格式，移动端支持【mp3，

aac, m4a, wav, ogg, mp4, mkv】，桌面端支持【mp3, aac, m4a, wav, mp4, mkv】。

onStart

onStart

void onStart	(int id
	int errCode)

背景音乐开始播放

在背景音乐开始播放成功或者失败时调用。

参数	描述
errCode	错误码。0: 开始播放成功；-4001:打开文件失败，如音频数据无效，FFMPEG 协议未找到等；-4005:非法路径导致打开文件失败，请检查您传入的路径参数是否指向一个合法的音乐文件；-4006:非法URL导致打开文件失败，请用浏览器检查您传入的 URL 地址是否可以下载到期望的音乐文件；-4007:无音频流导致打开文件失败，请确认您传入的文件是否是合法的音频文件，以及文件是否被损坏；-4008:格式不支持导致打开文件失败，请确认您传入的文件格式是否是支持的文件格式，移动端支持【mp3, aac, m4a, wav, ogg, mp4, mkv】，桌面端支持【mp3, aac, m4a, wav, mp4, mkv】。
id	音乐 ID。

onPlayProgress

onPlayProgress

void onPlayProgress	(int id
	long curPtsMS
	long durationMS)

背景音乐的播放进度

onComplete

onComplete

void onComplete	(int id
	int errCode)

背景音乐已经播放完毕

在背景音乐播放完毕或播放错误时调用。

参数	描述
errCode	错误码。0: 播放结束；-4002: 解码失败，如音频文件损坏，网络音频文件服务器无法访问等。
id	音乐 ID。

enableVoiceEarMonitor

enableVoiceEarMonitor

void enableVoiceEarMonitor	(boolean enable)
----------------------------	------------------

开启耳返

主播开启耳返后，可以在耳机里听到麦克风采集到的自己发出的声音，该特效适用于主播唱歌的应用场景中。

需要您注意的是，由于蓝牙耳机的硬件延迟非常高，所以在主播佩戴蓝牙耳机时无法开启此特效，请尽量在用户界面上提示主播佩戴有线耳机。

同时也需要注意，并非所有的手机开启此特效后都能达到优秀的耳返效果，我们已经对部分耳返效果不佳的手机屏蔽了该特效。

参数	描述
enable	true：开启；false：关闭。

注意

仅在主播佩戴耳机时才能开启此特效，同时请您提示主播佩戴有线耳机。

setVoiceEarMonitorVolume

setVoiceEarMonitorVolume

void setVoiceEarMonitorVolume	(int volume)
-------------------------------	--------------

设置耳返音量

通过该接口您可以设置耳返特效中声音的音量大小。

参数	描述
volume	音量大小，取值范围为 0 - 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoiceReverbType

setVoiceReverbType

void setVoiceReverbType	(TXVoiceReverbType type)
-------------------------	---

设置人声的混响效果

通过该接口您可以设置人声的混响效果，具体特效请参见枚举定义 [TXVoiceReverbType](#)。

注意

设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceChangerType

setVoiceChangerType

void setVoiceChangerType	(TXVoiceChangerType type)
--------------------------	--

设置人声的变声特效

通过该接口您可以设置人声的变声特效，具体特效请参见枚举定义 [TXVoiceChangeType](#)。

注意

设置的效果在退出房间后会自动失效，如果下次进房还需要对应特效，需要调用此接口再次进行设置。

setVoiceCaptureVolume

setVoiceCaptureVolume

void setVoiceCaptureVolume	(int volume)
----------------------------	--------------

设置语音音量

该接口可以设置语音音量的大小，一般配合音乐音量的设置接口 [setAllMusicVolume](#) 协同使用，用于调谐语音和音乐在混音前各自的音量占比。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：100。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setVoicePitch

setVoicePitch

void setVoicePitch	(double pitch)
--------------------	----------------

设置语音音调

该接口可以设置语音音调，用于实现变调不变速的目的。

参数	描述
pitch	音调，取值范围为-1.0f~1.0f，默认值：0.0f。

setMusicObserver

setMusicObserver

void setMusicObserver	(int id
	TXMusicPlayObserver observer)

设置背景音乐的事件回调接口

请在播放背景音乐之前使用该接口设置播放事件回调，以便感知背景音乐的播放进度。

参数	描述

musicId	音乐 ID。
observer	具体参考 ITXMusicPlayObserver 中定义接口。

注意

1. 如果该 ID 不需要使用，observer 可设置为 NULL 彻底释放。

startPlayMusic

startPlayMusic

boolean startPlayMusic	(final AudioMusicParam musicParam)
------------------------	--

开始播放背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
musicParam	音乐参数。

注意

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

stopPlayMusic

stopPlayMusic

void stopPlayMusic	(int id)
--------------------	----------

停止播放背景音乐

参数	描述
id	音乐 ID。

pausePlayMusic

pausePlayMusic

void pausePlayMusic	(int id)
---------------------	----------

暂停播放背景音乐

参数	描述
id	音乐 ID。

resumePlayMusic

resumePlayMusic

void resumePlayMusic	(int id)
----------------------	----------

恢复播放背景音乐

参数	描述
id	音乐 ID。

setAllMusicVolume

setAllMusicVolume

void setAllMusicVolume	(int volume)
------------------------	--------------

设置所有背景音乐的本地音量和远端音量的大小

该接口可以设置所有背景音乐的本地音量和远端音量。

本地音量：即主播本地可以听到的背景音乐的音量大小。

远端音量：即观众端可以听到的背景音乐的音量大小。

参数	描述
volume	音量大小，取值范围为0 - 100，默认值：60。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPublishVolume

setMusicPublishVolume

void setMusicPublishVolume	(int id
	int volume)

设置某一首背景音乐的远端音量的大小

该接口可以细粒度地控制每一首背景音乐的远端音量，也就是观众端可听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100；默认值：60。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPlayoutVolume

setMusicPlayoutVolume

void setMusicPlayoutVolume	(int id
	int volume)

设置某一首背景音乐的本地音量的大小

该接口可以细粒度地控制每一首背景音乐的本地音量，也就是主播本地可以听到的背景音乐的音量大小。

参数	描述
id	音乐 ID。
volume	音量大小，取值范围为0 - 100，默认值：60。

注意

如果将 volume 设置成 100 之后感觉音量还是太小，可以将 volume 最大设置成 150，但超过 100 的 volume 会有爆音的风险，请谨慎操作。

setMusicPitch

setMusicPitch

void setMusicPitch	(int id
	float pitch)

调整背景音乐的音调高低

参数	描述
id	音乐 ID。
pitch	音调，默认值是0.0f，范围是：[-1 ~ 1] 之间的浮点数。

setMusicSpeedRate

setMusicSpeedRate

void setMusicSpeedRate	(int id
	float speedRate)

调整背景音乐的变速效果

参数	描述
id	音乐 ID。
speedRate	速度，默认值是1.0f，范围是：[0.5 ~ 2] 之间的浮点数。

getMusicCurrentPosInMS

getMusicCurrentPosInMS

long getMusicCurrentPosInMS	(int id)
-----------------------------	----------

获取背景音乐的播放进度（单位：毫秒）

参数	描述

id	音乐 ID。
----	--------

返回值说明：

成功返回当前播放时间，单位：毫秒，失败返回 -1。

getMusicDurationInMS

getMusicDurationInMS

long getMusicDurationInMS	(String path)
---------------------------	---------------

获取背景音乐的总时长（单位：毫秒）

参数	描述
path	音乐文件路径。

返回值说明：

成功返回时长，失败返回 -1。

seekMusicToPosInMS

seekMusicToPosInMS

void seekMusicToPosInMS	(int id
	int pts)

设置背景音乐的播放进度（单位：毫秒）

参数	描述
id	音乐 ID。
pts	单位: 毫秒。

注意

请尽量避免过度频繁地调用该接口，因为该接口可能会再次读写音乐文件，耗时稍高。

因此，当用户拖拽音乐的播放进度条时，请在用户完成拖拽操作后再调用本接口。

因为 UI 上的进度条控件往往会以很高的频率反馈用户的拖拽进度，如不做频率限制，会导致较差的用户体验。

setMusicScratchSpeedRate

setMusicScratchSpeedRate

void setMusicScratchSpeedRate	(int id
	float scratchSpeedRate)

调整搓碟的变速效果

参数	描述
id	音乐 ID。
scratchSpeedRate	搓碟速度，默认值是1.0f，范围是：[-12.0 ~ 12.0] 之间的浮点数, 速度值正/负表示方向正/反，绝对值大小表示速度快慢。

注意

前置条件 preloadMusic 成功。

setPreloadObserver

setPreloadObserver

void setPreloadObserver	(TXMusicPreloadObserver observer)
-------------------------	--

设置预加载事件回调

请在预加载背景音乐之前使用该接口设置回调，以便感知背景音乐的预加载进度。

参数	描述
observer	具体参考 ITXMusicPreloadObserver 中定义接口。

preloadMusic

preloadMusic

boolean preloadMusic	(final AudioMusicParam preloadParam)
----------------------	--

预加载背景音乐

每个音乐都需要您指定具体的 ID，您可以通过该 ID 对音乐的开始、停止、音量等进行设置。

参数	描述
preloadParam	预加载音乐参数。

注意

1. 预先加载最多同时支持2个不同 ID 的预加载，且预加载时长不超过10分钟，使用完需 stopPlayMusic，否则内存不释放。
2. 若该ID对应的音乐正在播放中，预加载会失败，需先调用 stopPlayMusic。
3. 当 musicParam 和传入 startPlayMusic 的 musicParam 完全相同时，预加载有效。

getMusicTrackCount

getMusicTrackCount

int getMusicTrackCount	(int id)
------------------------	----------

获取背景音乐的音轨数量

参数	描述
id	音乐 ID。

setMusicTrack

setMusicTrack

void setMusicTrack	(int id
	int trackIndex)

指定背景音乐的播放音轨

参数	描述
id	音乐 ID。
index	默认播放第一个音轨。取值范围[0, 音轨总数)。

注意

音轨总数量可通过 `getMusicTrackCount` 接口获取。

TXVoiceReverbType

TXVoiceReverbType

混响特效

混响特效可以作用于人声之上，通过声学算法对声音进行叠加处理，模拟出各种不同环境下的临场感受，目前支持如下几种混响效果：

0：关闭；1：KTV；2：小房间；3：大会堂；4：低沉；5：洪亮；6：金属声；7：磁性；8：空灵；9：录音棚；10：悠扬；11：录音棚2。

枚举	取值	描述
TXLiveVoiceReverbType_0	0	关闭特效
TXLiveVoiceReverbType_1	1	KTV
TXLiveVoiceReverbType_2	2	小房间
TXLiveVoiceReverbType_3	3	大会堂
TXLiveVoiceReverbType_4	4	低沉
TXLiveVoiceReverbType_5	5	洪亮
TXLiveVoiceReverbType_6	6	金属声
TXLiveVoiceReverbType_7	7	磁性
TXLiveVoiceReverbType_8	8	空灵
TXLiveVoiceReverbType_9	9	录音棚
TXLiveVoiceReverbType_10	10	悠扬
TXLiveVoiceReverbType_11	11	录音棚2

TXVoiceChangeType

TXVoiceChangeType

变声特效

变声特效可以作用于人声之上，通过声学算法对人声进行二次处理，以获得与原始声音所不同的音色，目前支持如下几种变声特效：

0：关闭；1：熊孩子；2：萝莉；3：大叔；4：重金属；5：感冒；6：外语腔；7：困兽；8：肥宅；9：强电流；10：重机械；11：空灵。

枚举	取值	描述
TXLiveVoiceChangerType_0	0	关闭
TXLiveVoiceChangerType_1	1	熊孩子
TXLiveVoiceChangerType_2	2	萝莉
TXLiveVoiceChangerType_3	3	大叔
TXLiveVoiceChangerType_4	4	重金属
TXLiveVoiceChangerType_5	5	感冒
TXLiveVoiceChangerType_6	6	外语腔
TXLiveVoiceChangerType_7	7	困兽
TXLiveVoiceChangerType_8	8	肥宅
TXLiveVoiceChangerType_9	9	强电流
TXLiveVoiceChangerType_10	10	重机械
TXLiveVoiceChangerType_11	11	空灵

TXAudioMusicParam

TXAudioMusicParam

背景音乐的播放控制信息

该信息用于在接口 [startPlayMusic](#) 中指定背景音乐的相关信息，包括播放 ID、文件路径和循环次数等：

1. 如果要多次播放同一首背景音乐，请不要每次播放都分配一个新的 ID，我们推荐使用相同的 ID。
2. 若您希望同时播放多首不同的音乐，请为不同的音乐分配不同的 ID 进行播放。
3. 如果使用同一个 ID 播放不同音乐，SDK 会先停止播放旧的音乐，再播放新的音乐。

枚举类型	描述
endTimeMS	【字段含义】音乐结束播放时间点，单位毫秒，0表示播放至文件结尾。

id	【字段含义】音乐 ID。 【特殊说明】SDK 允许播放多路音乐，因此需要使用 ID 进行标记，用于控制音乐的开始、停止、音量等。
isShortFile	【字段含义】播放的是否为短音乐文件。 【推荐取值】true：需要重复播放的短音乐文件；false：正常的音乐文件。默认值：false。
loopCount	【字段含义】音乐循环播放的次数。 【推荐取值】取值范围为0 - 任意正整数，默认值：0。0 表示播放音乐一次；1 表示播放音乐两次；以此类推。
path	【字段含义】音效文件的完整路径或 URL 地址。支持的音频格式包括 MP3、AAC、M4A、WAV。
publish	【字段含义】是否将音乐传到远端。 【推荐取值】true：音乐在本地播放的同时，远端用户也能听到该音乐；false：主播只能在本地听到该音乐，远端观众听不到。默认值：false。
startTimeMS	【字段含义】音乐开始播放时间点，单位：毫秒。

TXBeautyManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 美颜与图像处理参数设置类

Function: 修改美颜、滤镜、绿幕等参数

TXBeautyManager

TXBeautyManager

函数列表	描述
setBeautyStyle	设置美颜（磨皮）算法
setBeautyLevel	设置美颜级别
setWhitenessLevel	设置美白级别
enableSharpnessEnhancement	开启清晰度增强
setRuddyLevel	设置红润级别
setFilter	设置色彩滤镜效果
setFilterStrength	设置色彩滤镜的强度
setGreenScreenFile	设置绿幕背景视频
setEyeScaleLevel	设置大眼级别
setFaceSlimLevel	设置瘦脸级别
setFaceVLevel	设置 V 脸级别
setChinLevel	设置下巴拉伸或收缩
setFaceShortLevel	设置短脸级别
setFaceNarrowLevel	设置窄脸级别

setNoseSlimLevel	设置瘦鼻级别
setEyeLightenLevel	设置亮眼级别
setToothWhitenLevel	设置牙齿美白级别
setWrinkleRemoveLevel	设置祛皱级别
setPouchRemoveLevel	设置祛眼袋级别
setSmileLinesRemoveLevel	设置法令纹去除级别
setForeheadLevel	设置发际线调整级别
setEyeDistanceLevel	设置眼距
setEyeAngleLevel	设置眼角调整级别
setMouthShapeLevel	设置嘴型调整级别
setNoseWingLevel	设置鼻翼调整级别
setNosePositionLevel	设置鼻子位置
setLipsThicknessLevel	设置嘴唇厚度
setFaceBeautyLevel	设置脸型
setMotionTpl	选择 AI 动效挂件
setMotionMute	是否在动效素材播放时静音

枚举类型

枚举类型	描述
TXBeautyStyle	美颜（磨皮）算法

setBeautyStyle

setBeautyStyle

void setBeautyStyle	(int beautyStyle)
---------------------	-------------------

设置美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案：

参数	描述
beautyStyle	美颜风格，TXBeautyStyleSmooth：光滑；TXBeautyStyleNature：自然；TXBeautyStylePitu：优图。

setBeautyLevel

setBeautyLevel

void setBeautyLevel	(float beautyLevel)
---------------------	---------------------

设置美颜级别

参数	描述
beautyLevel	美颜级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

setWhitenessLevel

setWhitenessLevel

void setWhitenessLevel	(float whitenessLevel)
------------------------	------------------------

设置美白级别

参数	描述
whitenessLevel	美白级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

enableSharpnessEnhancement

enableSharpnessEnhancement

void enableSharpnessEnhancement	(boolean enable)
---------------------------------	------------------

开启清晰度增强

setRuddyLevel

setRuddyLevel

void setRuddyLevel	(float ruddyLevel)
--------------------	--------------------

设置红润级别

参数	描述
ruddyLevel	红润级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

setFilter

setFilter

void setFilter	(Bitmap image)
----------------	----------------

设置色彩滤镜效果

色彩滤镜，是一副包含色彩映射关系的颜色查找表图片，您可以在我们提供的官方 **Demo** 中找到预先准备好的几张滤镜图片。

SDK 会根据该查找表中的映射关系，对摄像头采集出的原始视频画面进行二次处理，以达到预期的滤镜效果。

参数	描述
image	包含色彩映射关系的颜色查找表图片，必须是 png 格式。

setFilterStrength

setFilterStrength

void setFilterStrength	(float strength)
------------------------	------------------

设置色彩滤镜的强度

该数值越高，色彩滤镜的作用强度越明显，经过滤镜处理后的视频画面跟原画面的颜色差异越大。

我默认的滤镜浓度是 0.5，如果您觉得默认的滤镜效果不明显，可以设置为 0.5 以上的数字，最大值为 1。

参数	描述
strength	从 0 到 1，数值越大滤镜效果越明显，默认值为 0.5。

setGreenScreenFile

setGreenScreenFile

int setGreenScreenFile	(String path)
------------------------	---------------

设置绿幕背景视频

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。此接口所开启的绿幕功能不具备智能去除背景的能力，需要被拍摄者的背后有一块绿色的幕布来辅助产生特效。

参数	描述
path	MP4格式的视频文件路径; 设置空值表示关闭特效。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeScaleLevel

setEyeScaleLevel

int setEyeScaleLevel	(float eyeScaleLevel)
----------------------	-----------------------

设置大眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeScaleLevel	大眼级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceSlimLevel

setFaceSlimLevel

int setFaceSlimLevel	(float faceSlimLevel)
----------------------	-----------------------

设置瘦脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceSlimLevel	瘦脸级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceVLevel

setFaceVLevel

int setFaceVLevel	(float faceVLevel)
-------------------	--------------------

设置 V 脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceVLevel	V 脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setChinLevel

setChinLevel

int setChinLevel	(float chinLevel)
------------------	-------------------

设置下巴拉伸或收缩

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
chinLevel	下巴拉伸或收缩级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示收缩，大于 0 表示拉伸。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceShortLevel

setFaceShortLevel

int setFaceShortLevel	(float faceShortLevel)
-----------------------	------------------------

设置短脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceShortLevel	短脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceNarrowLevel

setFaceNarrowLevel

int setFaceNarrowLevel	(float faceNarrowLevel)
------------------------	-------------------------

设置窄脸级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
level	窄脸级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseSlimLevel

setNoseSlimLevel

int setNoseSlimLevel	(float noseSlimLevel)
----------------------	-----------------------

设置瘦鼻级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
noseSlimLevel	瘦鼻级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeLightenLevel

setEyeLightenLevel

int setEyeLightenLevel	(float eyeLightenLevel)
------------------------	-------------------------

设置亮眼级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeLightenLevel	亮眼级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setToothWhitenLevel

setToothWhitenLevel

int setToothWhitenLevel	(float toothWhitenLevel)
-------------------------	--------------------------

设置牙齿美白级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
toothWhitenLevel	白牙级别，取值范围 0 - 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setWrinkleRemoveLevel

setWrinkleRemoveLevel

int setWrinkleRemoveLevel	(float wrinkleRemoveLevel)
---------------------------	----------------------------

设置祛皱级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
wrinkleRemoveLevel	祛皱级别，取值范围0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setPouchRemoveLevel

setPouchRemoveLevel

int setPouchRemoveLevel	(float pouchRemoveLevel)
-------------------------	--------------------------

设置祛眼袋级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
pouchRemoveLevel	祛眼袋级别，取值范围 0 - 9；0 表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setSmileLinesRemoveLevel

setSmileLinesRemoveLevel

int setSmileLinesRemoveLevel	(float smileLinesRemoveLevel)
------------------------------	-------------------------------

设置法令纹去除级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
smileLinesRemoveLevel	法令纹级别，取值范围 0 - 9；0表示关闭，9 表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setForeheadLevel

setForeheadLevel

int setForeheadLevel	(float foreheadLevel)
----------------------	-----------------------

设置发际线调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
foreheadLevel	发际线级别，取值范围-9 - 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeDistanceLevel

setEyeDistanceLevel

int setEyeDistanceLevel	(float eyeDistanceLevel)
-------------------------	--------------------------

设置眼距

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述

eyeDistanceLevel

眼距级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setEyeAngleLevel

setEyeAngleLevel

int setEyeAngleLevel

(float eyeAngleLevel)

设置眼角调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
eyeAngleLevel	眼角调整级别，取值范围-9 - 9；0表示关闭，9表示效果最明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMouthShapeLevel

setMouthShapeLevel

int setMouthShapeLevel

(float mouthShapeLevel)

设置嘴型调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
mouthShapeLevel	嘴型级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNoseWingLevel

setNoseWingLevel

int setNoseWingLevel	(float noseWingLevel)
----------------------	-----------------------

设置鼻翼调整级别

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
noseWingLevel	鼻翼调整级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setNosePositionLevel

setNosePositionLevel

int setNosePositionLevel	(float nosePositionLevel)
--------------------------	---------------------------

设置鼻子位置

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
nosePositionLevel	鼻子位置级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示抬高，大于 0 表示降低。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setLipsThicknessLevel

setLipsThicknessLevel

int setLipsThicknessLevel	(float lipsThicknessLevel)
---------------------------	----------------------------

设置嘴唇厚度

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
lipsThicknessLevel	嘴唇厚度级别，取值范围 -9 - 9；0 表示关闭，小于 0 表示拉伸，大于 0 表示收缩。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setFaceBeautyLevel

setFaceBeautyLevel

int setFaceBeautyLevel	(float faceBeautyLevel)
------------------------	-------------------------

设置脸型

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
faceBeautyLevel	美型级别，取值范围 0 - 9；0 表示关闭，1 - 9 值越大，效果越明显。

返回值说明：

0：成功；-5：当前 License 对应 feature 不支持。

setMotionTpl

setMotionTpl

void setMotionTpl	(String tplPath)
-------------------	------------------

选择 AI 动效挂件

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。

参数	描述
tplPath	动效素材文件所在目录。

setMotionMute

setMotionMute

void setMotionMute	(boolean motionMute)
--------------------	----------------------

是否在动效素材播放时静音

该接口仅在 企业版 SDK（旧版已下线，新版本SDK如需使用高级美颜功能请参见[腾讯美颜特效SDK](#)）中生效。
有些挂件本身会有声音特效，通过此 API 可以关闭这些特效播放时所带的声音效果。

参数	描述
motionMute	true：静音；false：不静音。

TXBeautyStyle

TXBeautyStyle

美颜（磨皮）算法

TRTC 内置多种不同的磨皮算法，您可以选择最适合您产品定位的方案。

枚举	取值	描述
TXBeautyStyleSmooth	0	光滑，算法比较激进，磨皮效果比较明显，适用于秀场直播。
TXBeautyStyleNature	1	自然，算法更多地保留了面部细节，磨皮效果更加自然，适用于绝大多数直播场景。
TXBeautyStylePitu	2	优图，由优图实验室提供，磨皮效果介于光滑和自然之间，比光滑保留更多皮肤细节，比自然磨皮程度更高。

TXDeviceManager

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: 音视频设备管理模块

Function: 用于管理摄像头、麦克风和扬声器等音视频相关的硬件设备

TXDeviceManager

TXDeviceManager

函数列表	描述
isFrontCamera	判断当前是否为前置摄像头（仅适用于移动端）
switchCamera	切换前置或后置摄像头（仅适用于移动端）
getCameraZoomMaxRatio	获取摄像头的最大缩放倍数（仅适用于移动端）
setCameraZoomRatio	设置摄像头的缩放倍数（仅适用于移动端）
isAutoFocusEnabled	查询是否支持自动识别人脸位置（仅适用于移动端）
enableCameraAutoFocus	开启自动对焦功能（仅适用于移动端）
setCameraFocusPosition	设置摄像头的对焦位置（仅适用于移动端）
enableCameraTorch	开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）
setAudioRoute	设置音频路由（仅适用于移动端）
setExposureCompensation	设置摄像头的曝光参数，取值范围从-1到1
setCameraCapturerParam	设置摄像头采集偏好
setSystemVolumeType	设置系统音量类型（仅适用于移动端）

结构体类型

函数列表	描述
TXCameraCaptureParam	摄像头采集参数

枚举类型

枚举类型	描述
TXSystemVolumeType	系统音量类型（仅适用于移动设备）
TXAudioRoute	音频路由（即声音的播放模式）
TXCameraCaptureMode	摄像头采集偏好

isFrontCamera

isFrontCamera

判断当前是否为前置摄像头（仅适用于移动端）

switchCamera

switchCamera

int switchCamera	(boolean frontCamera)
------------------	-----------------------

切换前置或后置摄像头（仅适用于移动端）

getCameraZoomMaxRatio

getCameraZoomMaxRatio

获取摄像头的最大缩放倍数（仅适用于移动端）

setCameraZoomRatio

setCameraZoomRatio

--	--

int setCameraZoomRatio	(float zoomRatio)
------------------------	-------------------

设置摄像头的缩放倍数（仅适用于移动端）

参数	描述
zoomRatio	取值范围1 - 5，取值为1表示最远视角（正常镜头），取值为5表示最近视角（放大镜头）。最大值推荐为5，若超过5，视频数据会变得模糊不清。

isAutoFocusEnabled

isAutoFocusEnabled

查询是否支持自动识别人脸位置（仅适用于移动端）

enableCameraAutoFocus

enableCameraAutoFocus

int enableCameraAutoFocus	(boolean enabled)
---------------------------	-------------------

开启自动对焦功能（仅适用于移动端）

开启后，SDK 会自动检测画面中的人脸位置，并将摄像头的焦点始终对焦在人脸位置上。

setCameraFocusPosition

setCameraFocusPosition

int setCameraFocusPosition	(int x
	int y)

设置摄像头的对焦位置（仅适用于移动端）

您可以通过该接口实现如下交互：

1. 在本地摄像头的预览画面上，允许用户单击操作。
2. 在用户的单击位置显示一个矩形方框，以示摄像头会在此处对焦。
3. 随后将用户点击位置的坐标通过本接口传递给 SDK，之后 SDK 会操控摄像头按照用户期望的位置进行对焦。

参数	描述
----	----

position	对焦位置，请传入期望对焦点的坐标值
----------	-------------------

注意

使用该接口的前提是先通过 [enableCameraAutoFocus](#) 关闭自动对焦功能。

返回值说明：

0：操作成功；负数：操作失败。

enableCameraTorch

enableCameraTorch

boolean enableCameraTorch	(boolean enable)
---------------------------	------------------

开启/关闭闪光灯，也就是手电筒模式（仅适用于移动端）

setAudioRoute

setAudioRoute

int setAudioRoute	(TXAudioRoute route)
-------------------	---------------------------------------

设置音频路由（仅适用于移动端）

手机有两个音频播放设备：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。

设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

setExposureCompensation

setExposureCompensation

int setExposureCompensation	(float value)
-----------------------------	---------------

设置摄像头的曝光参数，取值范围从-1到1

setCameraCapturerParam

setCameraCapturerParam

```
void setCameraCapturerParam (TXCameraCaptureParam params)
```

设置摄像头采集偏好

setSystemVolumeType

setSystemVolumeType

```
int setSystemVolumeType (TXSystemVolumeType type)
```

设置系统音量类型（仅适用于移动端）

@deprecated v9.5 版本开始不推荐使用，建议使用 `TRTCCLoud` 中的 `startLocalAudio(quality)` 接口替代之，通过 `quality` 参数来决策音质。

TXSystemVolumeType(Deprecated)

TXSystemVolumeType(Deprecated)

系统音量类型（仅适用于移动设备）

@deprecated v9.5 版本开始不推荐使用。

现代智能手机中一般都具备两套系统音量类型，即“通话音量”和“媒体音量”。

通话音量：手机专门为接打电话所设计的音量类型，自带回声抵消（AEC）功能，并且支持通过蓝牙耳机上的麦克风进行拾音，缺点是音质比较一般。

当您通过手机侧面的音量按键下调手机音量时，如果无法将其调至零（也就是无法彻底静音），说明您的手机当前处于通话音量。

媒体音量：手机专门为音乐场景所设计的音量类型，无法使用系统的 AEC 功能，并且不支持通过蓝牙耳机的麦克风进行拾音，但具备更好的音乐播放效果。

当您通过手机侧面的音量按键下调手机音量时，如果能够将手机音量调至彻底静音，说明您的手机当前处于媒体音量。

SDK 目前提供了三种系统音量类型的控制模式：自动切换模式、全程通话音量模式、全程媒体音量模式。

枚举	取值	描述
TXSystemVolumeTypeAuto	Not Defined	自动切换模式

TXSystemVolumeTypeMedia	Not Defined	全程媒体音量
TXSystemVolumeTypeVOIP	Not Defined	全程通话音量

TXAudioRoute

TXAudioRoute

音频路由（即声音的播放模式）

音频路由，即声音是从手机的扬声器还是从听筒中播放出来，因此该接口仅适用于手机等移动端设备。

手机有两个扬声器：一个是位于手机顶部的听筒，一个是位于手机底部的立体声扬声器。

设置音频路由为听筒时，声音比较小，只有将耳朵凑近才能听清楚，隐私性较好，适合用于接听电话。

设置音频路由为扬声器时，声音比较大，不用将手机贴脸也能听清，因此可以实现“免提”的功能。

枚举	取值	描述
TXAudioRouteSpeakerphone	Not Defined	Speakerphone ：使用扬声器播放（即“免提”），扬声器位于手机底部，声音偏大，适合外放音乐。
TXAudioRouteEarpiece	Not Defined	Earpiece ：使用听筒播放，听筒位于手机顶部，声音偏小，适合需要保护隐私的通话场景。

TXCameraCaptureMode

TXCameraCaptureMode

摄像头采集偏好

该枚举类型用于摄像头采集参数设置。

枚举	取值	描述
TXCameraResolutionStrategyAuto	Not Defined	自动调整采集参数。 SDK 根据实际的采集设备性能及网络情况，选择合适的摄像头输出参数，在设备性能及视频预览质量之间，维持平衡。
TXCameraResolutionStrategyPerformance	Not Defined	优先保证设备性能。 SDK 根据用户设置编码器的分辨率和帧率，选择最接近的摄像头输出参数，从而

		保证设备性能。
TXCameraResolutionStrategyHighQuality	Not Defined	优先保证视频预览质量。 SDK选择较高的摄像头输出参数，从而提高预览视频的质量。在这种情况下，会消耗更多的 CPU 及内存做视频前处理。
TXCameraCaptureManual	Not Defined	允许用户设置本地摄像头采集的视频宽高。

TXCameraCaptureParam

TXCameraCaptureParam

摄像头采集参数

该设置能决定本地预览图像画质。

枚举类型	描述
height	字段含义：采集图像宽度
mode	字段含义：摄像头采集偏好，请参见 TXCameraCaptureMode
width	字段含义：采集图像长度

错误码表

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLiveCode @ TXLiteAVSDK

Function: 腾讯云直播服务(LVB)错误码和警告码的定义。

错误码表

枚举类型

枚举类型	描述
V2TXLiveCode	V2 错误码和警告码

V2TXLiveCode

V2TXLiveCode

V2 错误码和警告码

枚举	取值	描述
V2TXLIVE_OK	0	没有错误。
V2TXLIVE_ERROR_FAILED	-1	暂未归类的通用错误。
V2TXLIVE_ERROR_INVALID_PARAMETER	-2	调用 API 时，传入的参数不合法。
V2TXLIVE_ERROR_REFUSED	-3	API 调用被拒绝。
V2TXLIVE_ERROR_NOT_SUPPORTED	-4	当前 API 不支持调用。
V2TXLIVE_ERROR_INVALID_LICENSE	-5	license 不合法，调用失败。
V2TXLIVE_ERROR_REQUEST_TIMEOUT	-6	请求服务器超时。

V2TXLIVE_ERROR_SERVER_PROCESS_FAILED	-7	服务器无法处理您的请求。
V2TXLIVE_ERROR_DISCONNECTED	-8	连接断开。
V2TXLIVE_ERROR_NO_AVAILABLE_HEVC_DECODERS	-2304	找不到可用的 HEVC 解码器。
V2TXLIVE_WARNING_NETWORK_BUSY	1101	网络状况不佳：上行带宽太小，上传数据受阻。
V2TXLIVE_WARNING_VIDEO_BLOCK	2105	当前视频播放出现卡顿。
V2TXLIVE_WARNING_CAMERA_START_FAILED	-1301	摄像头打开失败。
V2TXLIVE_WARNING_CAMERA_OCCUPIED	-1316	摄像头正在被占用中，可尝试打开其他摄像头。
V2TXLIVE_WARNING_CAMERA_NO_PERMISSION	-1314	摄像头设备未授权，通常在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_MICROPHONE_START_FAILED	-1302	麦克风打开失败。
V2TXLIVE_WARNING_MICROPHONE_OCCUPIED	-1319	麦克风正在被占用中，例如移动设备正在通话时，打开麦克风会失败。
V2TXLIVE_WARNING_MICROPHONE_NO_PERMISSION	-1317	麦克风设备未授权，通常在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_SCREEN_CAPTURE_NOT_SUPPORTED	-1309	当前系统不支持屏幕分享。
V2TXLIVE_WARNING_SCREEN_CAPTURE_START_FAILED	-1308	开始录屏失败，如果在移动设备出现，可能是权限被用户拒绝了。
V2TXLIVE_WARNING_SCREEN_CAPTURE_INTERRUPTED	-7001	录屏被系统中断。
V2TXLIVE_WARNING_CURRENT_ENCODE_TYPE_CHANGED	1104	表示编码器发生改变，可以通过 <code>onWarning</code> 函数的扩展信息中的 <code>codec_type</code> 字段来获取当前的编码格式。其中 1 代表 265 编码，0 代表 264 编码。注意 Windows 端不支持此错误码的扩展信息。

V2TXLIVE_WARNING_CURRENT_DECODE_TYPE_CHANGED	2008	表示解码器发生改变，可以通过 <code>onWarning</code> 函数的扩展信息中的 <code>codec_type</code> 字段来获取当前的解码格式。其中 1 代表 265 解码，0 代表 264 解码。注意 Windows 端不支持此错误码的扩展信息。
--	------	---

类型定义

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLiveDef @ TXLiteAVSDK

Function: 腾讯云直播服务(LVB) 关键类型定义

类型定义

结构体类型

函数列表	描述
V2TXLiveVideoEncoderParam	视频编码参数
V2TXLiveVideoFrame	视频帧信息
V2TXLiveAudioFrame	音频帧数据
V2TXLiveAudioFrameObserverFormat	音频帧回调格式
V2TXLivePusherStatistics	推流器的统计数据
V2TXLivePlayerStatistics	播放器的统计数据
V2TXLiveMixStream	云端混流中每一路子画面的位置信息
V2TXLiveTranscodingConfig	云端混流（转码）配置
V2TXLiveLocalRecordingParams	本地录制音视频配置
V2TXLiveSocks5ProxyConfig	socks5 代理的协议配置
V2TXLiveLogConfig	Log配置
V2TXLiveStreamInfo	支持自适应切换的码流信息

枚举类型

枚举类型	描述
V2TXLiveMode	支持协议
V2TXLiveVideoResolution	视频分辨率
V2TXLiveVideoResolutionMode	视频宽高比模式
V2TXLiveMirrorType	本地摄像头镜像类型
V2TXLiveFillMode	视频画面填充模式
V2TXLiveRotation	视频画面顺时针旋转角度
V2TXLivePixelFormat	视频帧的像素格式
V2TXLiveBufferType	视频数据包装格式
V2TXLiveTexture	视频纹理包装类
V2TXLiveAudioQuality	声音音质
V2TXLiveAudioFrameOperationMode	音频回调数据读写模式
V2TXLivePushStatus	直播流的连接状态
V2TXLiveMixInputType	混流输入类型配置
V2TXLiveRecordMode	本地音视频录制模式
V2TXLiveLogLevel	日志级别枚举值

V2TXLiveMode

V2TXLiveMode

支持协议

枚举	取值	描述
TXLiveMode_RTMP	Not Defined	支持协议: RTMP。
TXLiveMode_RTC	Not Defined	支持协议: TRTC。

V2TXLiveVideoResolution

V2TXLiveVideoResolution

视频分辨率

枚举	取值	描述
V2TXLiveVideoResolution160x160	Not Defined	分辨率 160*160，码率范围：100Kbps ~ 150Kbps，帧率：15fps。
V2TXLiveVideoResolution270x270	Not Defined	分辨率 270*270，码率范围：200Kbps ~ 300Kbps，帧率：15fps。
V2TXLiveVideoResolution480x480	Not Defined	分辨率 480*480，码率范围：350Kbps ~ 525Kbps，帧率：15fps。
V2TXLiveVideoResolution320x240	Not Defined	分辨率 320*240，码率范围：250Kbps ~ 375Kbps，帧率：15fps。
V2TXLiveVideoResolution480x360	Not Defined	分辨率 480*360，码率范围：400Kbps ~ 600Kbps，帧率：15fps。
V2TXLiveVideoResolution640x480	Not Defined	分辨率 640*480，码率范围：600Kbps ~ 900Kbps，帧率：15fps。
V2TXLiveVideoResolution320x180	Not Defined	分辨率 320*180，码率范围：250Kbps ~ 400Kbps，帧率：15fps。
V2TXLiveVideoResolution480x270	Not Defined	分辨率 480*270，码率范围：350Kbps ~ 550Kbps，帧率：15fps。
V2TXLiveVideoResolution640x360	Not Defined	分辨率 640*360，码率范围：500Kbps ~ 900Kbps，帧率：15fps。
V2TXLiveVideoResolution960x540	Not Defined	分辨率 960*540，码率范围：800Kbps ~ 1500Kbps，帧率：15fps。
V2TXLiveVideoResolution1280x720	Not Defined	分辨率 1280*720，码率范围：1000Kbps ~ 1800Kbps，帧率：15fps。
V2TXLiveVideoResolution1920x1080	Not Defined	分辨率 1920*1080，码率范围：2500Kbps ~ 3000Kbps，帧率：15fps。

V2TXLiveVideoResolutionMode

V2TXLiveVideoResolutionMode

视频宽高比模式

注意

横屏模式下的分辨率: V2TXLiveVideoResolution640x360 + V2TXLiveVideoResolutionModeLandscape = 640 × 360。

竖屏模式下的分辨率: V2TXLiveVideoResolution640x360 + V2TXLiveVideoResolutionModePortrait = 360 × 640。

枚举	取值	描述
V2TXLiveVideoResolutionModeLandscape	Not Defined	横屏模式。
V2TXLiveVideoResolutionModePortrait	Not Defined	竖屏模式。

V2TXLiveMirrorType

V2TXLiveMirrorType

本地摄像头镜像类型

枚举	取值	描述
V2TXLiveMirrorTypeAuto	Not Defined	系统默认镜像类型，前置摄像头镜像，后置摄像头不镜像。
V2TXLiveMirrorTypeEnable	Not Defined	前置摄像头和后置摄像头，都切换为镜像模式。
V2TXLiveMirrorTypeDisable	Not Defined	前置摄像头和后置摄像头，都切换为非镜像模式。

V2TXLiveFillMode

V2TXLiveFillMode

视频画面填充模式

枚举	取值	描述
V2TXLiveFillModeFill	Not	图像铺满屏幕，超出显示视窗的视频部分将被裁剪，画

	Defined	面显示可能不完整。
V2TXLiveFillModeFit	Not Defined	图像长边填满屏幕，短边区域会被填充黑色，画面的内容完整。
V2TXLiveFillModeScaleFill	Not Defined	图像拉伸铺满，因此长度和宽度可能不会按比例变化。

V2TXLiveRotation

V2TXLiveRotation

视频画面顺时针旋转角度

枚举	取值	描述
V2TXLiveRotation0	Not Defined	不旋转。
V2TXLiveRotation90	Not Defined	顺时针旋转90度。
V2TXLiveRotation180	Not Defined	顺时针旋转180度。
V2TXLiveRotation270	Not Defined	顺时针旋转270度。

V2TXLivePixelFormat

V2TXLivePixelFormat

视频帧的像素格式

枚举	取值	描述
V2TXLivePixelFormatUnknown	Not Defined	未知。
V2TXLivePixelFormatI420	Not Defined	YUV420P I420。
V2TXLivePixelFormatTexture2D	Not Defined	OpenGL 2D 纹理。

V2TXLiveBufferType

V2TXLiveBufferType

视频数据包装格式

注意

在自定义采集和自定义渲染功能，您需要用到下列枚举值来指定您希望以什么类型的容器来包装视频数据。

Texture: 直接使用时效率最高。

枚举	取值	描述
V2TXLiveBufferTypeUnknown	Not Defined	未知。
V2TXLiveBufferTypeByteBuffer	Not Defined	DirectBuffer，装载 I420 等 buffer，在 native 层使用。
V2TXLiveBufferTypeByteArray	Not Defined	byte[]，装载 I420 等 buffer，在 Java 层使用。
V2TXLiveBufferTypeTexture	Not Defined	直接操作纹理 ID，性能最好，画质损失最少。

V2TXLiveTexture

V2TXLiveTexture

视频纹理包装类

枚举	取值	描述
public int textureId	Not Defined	视频纹理 ID。
public javax.microedition.khronos.egl.EGLContext eglContext10	Not Defined	使用 (javax.microedition.khronos.egl.*) 定义的 OpenGL 接口。
public android.opengl.EGLContext eglContext14	Not Defined	使用 (android.opengl.*) 定义的 OpenGL 接口。

V2TXLiveAudioQuality

V2TXLiveAudioQuality

声音音质

枚举	取值	描述
V2TXLiveAudioQualitySpeech	Not Defined	语音音质：采样率：16k；单声道；音频码率：16kbps；适合语音通话为主的场景，比如在线会议，语音通话。
V2TXLiveAudioQualityDefault	Not Defined	默认音质：采样率：48k；单声道；音频码率：50kbps；SDK 默认的音频质量，如无特殊需求推荐选择之。
V2TXLiveAudioQualityMusic	Not Defined	音乐音质：采样率：48k；双声道 + 全频带；音频码率：128kbps；适合需要高保真传输音乐的场景，比如 K 歌、音乐直播等。

V2TXLiveAudioFrameOperationMode

V2TXLiveAudioFrameOperationMode

音频回调数据读写模式

SDK 提供了两种音频回调数据的操作模式。

读写模式（ReadWrite）：可以获取并修改回调的音频数据，默认模式。

只读模式（ReadOnly）：仅从回调中获取音频数据。

枚举	取值	描述
V2TXLiveAudioFrameOperationModeReadWrite	Not Defined	读写模式：可以获取并修改回调的音频数据。
V2TXLiveAudioFrameOperationModeReadOnly	Not Defined	只读模式：仅从回调中获取音频数据。

V2TXLivePushStatus

V2TXLivePushStatus

直播流的连接状态

枚举	取值	描述
V2TXLivePushStatusDisconnected	Not Defined	与服务器断开连接。
V2TXLivePushStatusConnecting	Not Defined	正在连接服务器。
V2TXLivePushStatusConnectSuccess	Not Defined	连接服务器成功。
V2TXLivePushStatusReconnecting	Not Defined	重连服务器中。

V2TXLiveMixInputType

V2TXLiveMixInputType

混流输入类型配置

枚举	取值	描述
V2TXLiveMixInputTypeAudioVideo	Not Defined	混入音视频。
V2TXLiveMixInputTypePureVideo	Not Defined	只混入视频。
V2TXLiveMixInputTypePureAudio	Not Defined	只混入音频。

V2TXLiveRecordMode

V2TXLiveRecordMode

本地音视频录制模式

枚举	取值	描述
V2TXLiveRecordModeBoth	Not Defined	Both mode: 录制音频和视频

V2TXLiveLogLevel

V2TXLiveLogLevel

日志级别枚举值

枚举	取值	描述
V2TXLiveLogLevelAll	0	输出所有级别的 log。
V2TXLiveLogLevelDebug	1	输出 DEBUG, INFO, WARNING, ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelInfo	2	输出 INFO, WARNING, ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelWarning	3	只输出 WARNING, ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelError	4	只输出 ERROR 和 FATAL 级别的 log。
V2TXLiveLogLevelFatal	5	只输出 FATAL 级别的 log。
V2TXLiveLogLevelNULL	6	不输出任何 sdk log。

V2TXLiveVideoEncoderParam

V2TXLiveVideoEncoderParam

视频编码参数

该设置决定远端用户看到的画面质量。

枚举类型	描述
minVideoBitrate	【字段含义】最低视频码率，SDK 会在网络不佳的情况下主动降低视频码率以保持流畅度，最低会降至 minVideoBitrate 所设定的数值。 【推荐取值】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoBitrate	【字段含义】目标视频码率，SDK 会按照目标码率进行编码，只有在弱网络环境下才会主动降低视频码率。 【推荐取值】请参考 V2TXLiveVideoResolution 在各档位注释的最佳码率，也可以在此基础上适当调高。 比如：V2TXLiveVideoResolution1280x720 对应 1200kbps 的目标码率，您也可以设置为 1500kbps 用来获得更好的观感清晰度。

	【特别说明】您可以通过同时设置 videoBitrate 和 minVideoBitrate 两个参数，用于约束 SDK 对视频码率的调整范围： 如果您将 videoBitrate 和 minVideoBitrate 设置为同一个值，等价于关闭 SDK 对视频码率的自适应调节能力。
videoFps	【字段含义】视频采集帧率。 【推荐取值】15fps 或 20fps。5fps 以下，卡顿感明显。10fps 以下，会有轻微卡顿感。20fps 以上，会浪费带宽（电影的帧率为 24fps）。
videoResolution	【字段含义】视频分辨率。 【特别说明】如需使用竖屏分辨率，请指定 videoResolutionMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。 【推荐取值】 桌面平台（Win + Mac）：建议选择 640 × 360 及以上分辨率，videoResolutionMode 选择 Landscape，即横屏分辨率。
videoResolutionMode	【字段含义】分辨率模式（横屏分辨率 or 竖屏分辨率）。 【推荐取值】桌面平台（Windows、Mac）建议选择 Landscape。 【特别说明】如需使用竖屏分辨率，请指定 resMode 为 Portrait，例如：640 × 360 + Portrait = 360 × 640。

V2TXLiveVideoFrame

V2TXLiveVideoFrame

视频帧信息

注意

自定义采集和自定义渲染时使用。自定义采集时，需要使用 V2TXLiveVideoFrame 来包装待发送的视频帧；自定义渲染时，会返回经过 V2TXLiveVideoFrame 包装的视频帧。

枚举类型	描述
buffer	【字段含义】视频数据。
bufferType	【字段含义】视频数据包装格式。
data	【字段含义】视频数据。
height	【字段含义】视频高度。
pixelFormat	【字段含义】视频帧像素格式。
rotation	【字段含义】视频帧的顺时针旋转角度。
texture	【字段含义】视频纹理包装类。

width

【字段含义】 视频宽度。

V2TXLiveAudioFrame

V2TXLiveAudioFrame

音频帧数据

枚举类型	描述
channel	【字段含义】 声道数。
data	【字段含义】 音频数据。
sampleRate	【字段含义】 采样率。
timestamp	【字段含义】 时间戳，单位ms。

V2TXLiveAudioFrameObserverFormat

V2TXLiveAudioFrameObserverFormat

音频帧回调格式

枚举类型	描述
channel	【字段含义】 声道数。 【推荐取值】 默认值：1，代表单声道。可设定的数值只有两个数字：1-单声道，2-双声道。
mode	【字段含义】 回调数据读写模式。 【推荐取值】 V2TXLiveAudioFrameOperationModeReadOnly：仅从回调中获取音频数据。可设定的模式有 V2TXLiveAudioFrameOperationModeReadOnly, V2TXLiveAudioFrameOperationModeReadWrite。
sampleRate	【字段含义】 采样率。 【推荐取值】 默认值：48000Hz。支持 16000, 32000, 44100, 48000。
samplesPerCall	【字段含义】 采样点数。 【推荐取值】 取值必须是 sampleRate/100 的整数倍。

V2TXLivePusherStatistics

V2TXLivePusherStatistics

推流器的统计数据

枚举类型	描述
appCpu	【字段含义】当前 App 的 CPU 使用率（%）。
audioBitrate	【字段含义】音频码率（Kbps）。
fps	【字段含义】帧率（fps）。
height	【字段含义】视频高度。
netSpeed	【字段含义】上行速度（kbps）
rtt	【字段含义】从 SDK 到云端的往返延时（ms）
systemCpu	【字段含义】当前系统的 CPU 使用率（%）。
videoBitrate	【字段含义】视频码率（Kbps）。
width	【字段含义】视频宽度。

V2TXLivePlayerStatistics

V2TXLivePlayerStatistics

播放器的统计数据

枚举类型	描述
appCpu	【字段含义】当前 App 的 CPU 使用率（%）。
audioBitrate	【字段含义】音频码率（Kbps）。
audioBlockRate	【字段含义】音频播放卡顿率，单位（%）。 音频播放卡顿率（audioBlockRate）= 音频播放的累计卡顿时长（audioTotalBlockTime）/ 音频播放的区间时长（2000ms）。
audioPacketLoss	【字段含义】网络音频丢包率（%），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
audioTotalBlockTime	【字段含义】音频播放的累计卡顿时长（ms）。

	该时长为区间（2s）内的卡顿时长。
fps	【字段含义】帧率（fps）。
height	【字段含义】视频高度。
jitterBufferDelay	【字段含义】播放延迟（ms）。
netSpeed	【字段含义】下载速度（kbps）
rtt	【字段含义】从 SDK 到云端的往返延时（ms），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
systemCpu	【字段含义】当前系统的 CPU 使用率（%）。
videoBitrate	【字段含义】视频码率（Kbps）。
videoBlockRate	【字段含义】视频播放卡顿率，单位（%）。 视频播放卡顿率（videoBlockRate）= 视频播放的累计卡顿时长（videoTotalBlockTime）/ 视频播放的区间时长（2000ms）。
videoPacketLoss	【字段含义】网络视频丢包率（%），注：仅支持前缀为 [trtc://] 或 [webrtc://] 的播放地址。
videoTotalBlockTime	【字段含义】视频播放的累计卡顿时长（ms）。 该时长为区间（2s）内的卡顿时长。
width	【字段含义】视频宽度。

V2TXLiveMixStream

V2TXLiveMixStream

云端混流中每一路子画面的位置信息

枚举类型	描述
height	【字段含义】图层位置高度（绝对像素值）。
inputType	【字段含义】该直播流的输入类型。
streamId	【字段含义】参与混流的 userId 所在对应的推流 streamId，nil 表示当前推流 streamId。
userId	【字段含义】参与混流的 userId。
width	【字段含义】图层位置宽度（绝对像素值）。

x	【字段含义】 图层位置 x 坐标（绝对像素值）。
y	【字段含义】 图层位置 y 坐标（绝对像素值）。
zOrder	【字段含义】 图层层次（1 - 15）不可重复。

V2TXLiveTranscodingConfig

V2TXLiveTranscodingConfig

云端混流（转码）配置

枚举类型	描述
audioBitrate	【字段含义】 最终转码后的音频码率。 【推荐取值】 默认值：64kbps，取值范围是 [32, 192]，单位：kbps。
audioChannels	【字段含义】 最终转码后的音频声道数。 【推荐取值】 默认值：1。取值范围为 [1,2] 中的整型。
audioSampleRate	【字段含义】 最终转码后的音频采样率。 【推荐取值】 默认值：48000Hz。支持12000HZ、16000HZ、22050HZ、24000HZ、32000HZ、44100HZ、48000HZ。
backgroundColor	【字段含义】 混合后画面的底色颜色，默认为黑色，格式为十六进制数字，比如：“0x61B9F1”代表 RGB 分别为(97,158,241)。 【推荐取值】 默认值：0x000000，黑色。
backgroundImage	【字段含义】 混合后画面的背景图。 【推荐取值】 默认值：nil，即不设置背景图。 【特别说明】 背景图需要您事先在“控制台 => 应用管理 => 功能配置 => 素材管理”中上传，上传成功后可以获得对应的“图片ID”，然后将“图片ID”转换成字符串类型并设置到 backgroundImage 里即可。 例如：假设“图片ID”为 63，可以设置 backgroundImage = "63"。
mixStreams	【字段含义】 每一路子画面的位置信息。
outputStreamId	【字段含义】 输出到 CDN 上的直播流 ID。 如不设置该参数，SDK 会执行默认逻辑，即房间里的多路流会混合到该接口调用者的视频流上，也就是 $A + B \Rightarrow A$。 如果设置该参数，SDK 会将房间里的多路流混合到您指定的直播流 ID 上，也就是 $A + B \Rightarrow C$。 【推荐取值】 默认值：nil，即房间里的多路流会混合到该接口调用者的视频流上。
videoBitrate	【字段含义】 最终转码后的视频分辨率的码率（kbps）。

	【推荐取值】如果填0，后台会根据 videoWidth 和 videoHeight 来估算码率，您也可以参考枚举定义 V2TXLiveVideoResolution 的注释。
videoFramerate	【字段含义】最终转码后的视频分辨率的帧率（FPS）。 【推荐取值】默认值：15fps，取值范围是 (0,30]。
videoGOP	【字段含义】最终转码后的视频分辨率的关键帧间隔（又称为 GOP）。 【推荐取值】默认值：2，单位为秒，取值范围是 [1,8]。
videoHeight	【字段含义】最终转码后的视频分辨率的高度。 【推荐取值】推荐值：640px，如果你是纯音频推流，请将 width × height 设为 0px × 0px，否则混流后会携带一条画布背景的视频流。
videoWidth	【字段含义】最终转码后的视频分辨率的宽度。 【推荐取值】推荐值：360px，如果你是纯音频推流，请将 width × height 设为 0px × 0px，否则混流后会携带一条画布背景的视频流。

V2TXLiveLocalRecordingParams

V2TXLiveLocalRecordingParams

本地录制音视频配置

枚举类型	描述
filePath	【字段含义】录制的文件地址（必填），请确保路径有读写权限且合法，否则录制文件无法生成。 【推荐取值】该路径需精确到文件名及格式后缀，格式后缀用于决定录制出的文件格式，目前支持的格式暂时只有 MP4。
interval	【字段含义】interval 录制信息更新频率，单位毫秒，有效范围：1000-10000。 【推荐取值】-1，表示不回调。
recordMode	【字段含义】媒体录制模式。 【推荐取值】V2TXLiveRecordModeBoth，即同时录制音频和视频。

V2TXLiveSocks5ProxyConfig

V2TXLiveSocks5ProxyConfig

socks5 代理的协议配置

枚举类型	描述

supportHttps	【字段含义】 是否支持 https。 【推荐取值】 默认值：true。
supportTcp	【字段含义】 是否支持 tcp。 【推荐取值】 默认值：true。
supportUdp	【字段含义】 是否支持 udp。 【推荐取值】 默认值：true。

V2TXLiveLogConfig

V2TXLiveLogConfig

Log配置

枚举类型	描述
enableConsole	【字段含义】 是否允许 SDK 在编辑器（XCoder、Android Studio、Visual Studio 等）的控制台上打印 Log。 【推荐取值】 默认值：false。
enableLogFile	【字段含义】 是否启用本地 Log 文件。 【特殊说明】 如非特殊需要，请不要关闭本地 Log 文件，否则腾讯云技术团队将无法在出现问题时进行跟踪和定位。 【推荐取值】 默认值：true。
enableObserver	【字段含义】 是否通过 V2TXLivePremierObserver 接收要打印的 Log 信息。 【特殊说明】 如果您希望自己实现 Log 写入，可以打开此开关，Log 信息会通过 V2TXLivePremierObserver#onLog 回调给您。 【推荐取值】 默认值：false。
logLevel	【字段含义】 设置 Log 级别。 【推荐取值】 默认值：V2TXLiveLogLevel.V2TXLiveLogLevelAll。
logPath	【字段含义】 设置本地 Log 的存储目录，默认 Log 存储位置： Android：/sdcard/Android/data/应用包名/files/log/liteav/。

V2TXLiveStreamInfo

V2TXLiveStreamInfo

支持自适应切换的码流信息

--	--

枚举类型	描述
bitrate	【字段含义】 码率, 单位bps, 默认值：0, 表示未知。
framerate	【字段含义】 帧率, 默认值：0, 表示未知。
height	【字段含义】 视频高, 默认值：0, 表示未知。
url	【字段含义】 流地址, 通过 SwitchStream 接口调用实现多码率质量切换。
width	【字段含义】 视频宽, 默认值：0, 表示未知。

属性定义

最近更新时间：2024-06-06 15:47:57

Copyright (c) 2021 Tencent. All rights reserved.

Module: V2TXLiveProperty @ TXLiteAVSDK

Function: V2TXLive setProperty 支持的 key

属性定义

结构体类型

函数列表	描述
V2TXLiveProperty	V2 直播属性

V2TXLiveProperty

V2TXLiveProperty

V2 直播属性

枚举类型	描述
kV2ClearLastImage	是否清理最后一帧。 默认值：true。 Value：true/false。
kV2EnableHardwareAcceleration	开启/关闭硬件加速【RTMP协议，拉流】。 默认值：true。 Value：true/false。
kV2EnableHevcEncode	开启/关闭 Hevc 编码【RTMP/RTC协议，推流】。 默认值：false。 Value：true/false。
kV2EnableIPMultiplexing	开启/关闭 IP 复用【FLV，拉流】。 默认值：false。 Value：true/false。

kV2MaxNumberOfReconnection	设置重连次数，【RTMP协议，拉流】。 默认值：3。 Value：int。
kV2SecondsBetweenReconnection	设置重连间隔【RTMP协议，拉流】。 单位：秒。 默认值：3。 Value：int。
kV2SetHeaders	设定播放请求头【FLV，拉流】。 Value：JSON 字符串。 例如： <pre>{ "headers": [{ "key": "key1", "value": "value1" }, { "key": "key2", "value": "value2" }] }</pre>
kV2SetMetaData	设置推流 Meta 信息【RTMP，推流】。 Value：JSON 字符串。 例如： <pre>{ "metadata": [{ "key": "key1", "value": "value1" }, { "key": "key2", "value": "value2" }] }</pre>
kV2SetVideoQualityEx	设置自定义编码参数【RTMP/RTC协议，推流】。 Value：JSON 字符串。 例如： <pre>{</pre>

```
"videoWidth":360,
"videoHeight":640,
"videoFps":15,
"videoBitrate":1000,
"minVideoBitrate":1000
}
```