



视沃科技

大牛直播 SDK(V2)

移动端调用说明

官网: <https://daniulive.com>

Github: <https://github.com/daniulive/SmarterStreaming>

声 明

非常感谢您选用我们的 SDK，您的支持将激励我们持续进步。

随着产品的迭代，产品手册会在每个版本发布后不定期更新，最新版本请以官方网站 (<https://daniulive.com>) 为准。

本手册中内容仅为开发者提供参考指导作用，具体调用请以 SDK 示例为准。

目 录

| | |
|--|----|
| 声 明..... | 2 |
| 1 平台概述..... | 5 |
| 1.1 Android/iOS 平台模块..... | 5 |
| 1.2 Android/iOS 播放端 SDK 扩展模块..... | 6 |
| 1.3 Android/iOS 平台 Demo..... | 6 |
| 1.3.1 Android 平台 Demo..... | 6 |
| 1.3.2 iOS 平台 Demo..... | 7 |
| 1.4 SDK 功能详解..... | 8 |
| 1.4.1 Android/iOS 推送端 SDK..... | 8 |
| 1.4.2 Android/iOS 播放端 SDK..... | 9 |
| 1.4.3 Android/iOS 多路流媒体转发 SDK..... | 10 |
| 1.4.4 Android/iOS 录像 SDK..... | 11 |
| 1.4.5 Android/iOS 一对一互动 SDK..... | 12 |
| 1.4.6 Android/iOS SEI 扩展数据发送/接收 SDK..... | 12 |
| 2 Android SDK 使用说明..... | 13 |
| 2.1 Android 推流端 SDK 说明..... | 13 |
| 2.1.1 系统要求..... | 13 |
| 2.1.2 准备工作..... | 13 |
| 2.1.3 SDK 接口详解..... | 14 |
| 2.2 Android 推流端 SDK Event 回调说明..... | 19 |
| 2.3 Android 推送端录像 SDK 说明..... | 20 |
| 2.4 Android 转发 SDK 说明..... | 21 |
| 2.5 Android 内置轻量级 RTSP 服务 SDK 说明..... | 22 |
| 2.6 Android SEI 扩展数据发送/接收 SDK..... | 24 |
| 2.7 Android 播放端 SDK 说明..... | 26 |
| 2.7.1 系统要求..... | 26 |
| 2.7.2 集成说明..... | 26 |
| 2.7.3 SDK 接口详解..... | 27 |
| 2.8 Android 播放端 SDK Event 回调说明..... | 29 |
| 2.9 Android 播放端录像 SDK 说明..... | 30 |
| 3 iOS SDK 使用说明..... | 32 |
| 3.1 iOS 推流端 SDK 说明..... | 32 |
| 3.1.1 系统要求..... | 32 |
| 3.1.2 准备工作..... | 32 |
| 3.1.3 SDK 接口详解..... | 33 |
| 3.2 iOS 推流端 SDK Event 回调说明..... | 37 |
| 3.3 iOS 推送端录像 SDK 说明..... | 38 |
| 3.4 iOS 转发 SDK 说明..... | 39 |
| 3.5 iOS 内置轻量级 RTSP 服务 SDK 说明..... | 40 |

| | |
|---------------------------------|----|
| 3.6 iOS SEI 扩展数据发送/接收 SDK..... | 42 |
| 3.7 iOS 播放端 SDK 说明..... | 43 |
| 3.7.1 系统要求..... | 43 |
| 3.7.2 准备工作..... | 43 |
| 3.7.3 SDK 接口详解..... | 44 |
| 3.8 iOS 播放端 SDK Event 回调说明..... | 46 |
| 3.9 iOS 播放端录像 SDK 说明..... | 48 |
| 3.10 iOS 一对一互动 SDK 说明..... | 49 |

1 平台概述

视沃科技旗下大牛直播 SDK，始于 2014 年，国内外为数不多致力于极致体验的**超强全自研跨平台**(windows/android/iOS)**流媒体内核**，通过模块化自由组合，支持实时 RTMP 推流、RTSP 推流、RTMP/RTSP 直播播放(支持 RTSP/RTMP H.265)、RTMP 推送播放**整体加密(AES/SM4)方案**、**实时录像**、多路流媒体**转发**(RTSP 转 RTMP，RTMP 转 RTMP)、音视频导播、动态视频合成、音频混音、**一对一互动直播**、内置轻量级 RTSP 服务、RTSP 网关服务等，**比快更快**，业界**真正靠谱**的超低延迟直播 SDK(1 秒内，低延迟模式下 200~400ms)。

1.1 Android/iOS 平台模块

| Android/iOS 平台模块介绍 | |
|--------------------|--|
| 模块名称 | 描述 |
| RTMP 推流端 SDK | Android/iOS 屏幕、摄像头 RTMP 推流 SDK(支持 RTMP 扩展 H.265 推送)，支持逐帧数据加密(加密算法支持 AES128、AES192、AES256、SM4(国密)) |
| RTSP 推流端 SDK | 支持 RTSP H.264/H.265 推送，音频支持 AAC 格式，支持 TCP/UDP 模式推送，支持 RTSP 鉴权，支持重连和异常网络处理，超低延迟 |
| RTMP 直播播放器 SDK | 业内首屈一指的 RTMP 超低延迟直播播放器 SDK(支持 RTMP H.265 扩展播放)，支持 RTMP 推送端加密数据(AES/SM4)正常播放 |
| RTSP 直播播放器 SDK | 支持 RTSP H.264/H.265 推送，音频支持 AAC 格式，支持 TCP/UDP 模式推送，支持 RTSP 鉴权，支持重连和异常网络处理，超低延迟 |
| Unity3D 直播播放器 SDK | 业内首家 Android/iOS 支持 Unity3D 的超低延迟 RTMP/RTSP 直播播放器 SDK，支持快照、录像、实时静音、view 旋转、快速切换 URL 等特性 |
| 录像 SDK | 支持拉取 rtmp/rtsp 流实时 录像 模块/实时 快照 功能，支持纯音频、纯视频、音视频录制模式，支持音频(PCMU/PCMA, Speex 等) 转 AAC 后再录像，业内为数不多的支持 RTSP/RTMP H.265 录制到 MP4 文件 的录像 SDK |
| 转发 SDK | 支持同时 多路拉取 rtmp/rtsp 流 ，并分别转发到服务器，支持转发过程中，拉取的 rtsp/rtmp 流实时内容切换，业内为数不多支持 RTSP/RTMP H.265 |

| | |
|-------------------|--|
| | 拉流转发的 SDK(提供配套 RTMP 扩展 H.265 服务器) |
| 互动 SDK | Android、iOS 平台一对一互动 |
| 轻量级 RTSP 服务 SDK | 为满足内网无纸化/电子教室等内网超低延迟需求，避免让用户配置单独的服务器，大牛直播 SDK 在推送端支持轻量级 RTSP 服务 SDK，推送端 SDK 支持的功能，内置轻量级 RTSP 服务 SDK 后，功能继续支持(支持 H.265)，支持单播、组播模式 |
| SEI 扩展数据发送/接收 SDK | 支持推送端通过 H.264 SEI 信息扩展，实时传输文本/二进制数据信息(如实时字幕/时间戳/题目分发/公告广播等)，播放端做相应解析和回显 |
| 视频处理 SDK(Android) | Android 文字水印、png 图片水印 |
| 音频处理 SDK(Android) | Android 回音消除、噪音抑制、自动增益、VAD 检测模块 |

1.2 Android/iOS 播放端 SDK 扩展模块

| Android/iOS 播放器 SDK 扩展模块 | |
|--------------------------|--------------------------------|
| 功能 | 功能描述 |
| 标准功能 | 支持“Android/iOS 播放端 SDK”列取的常规功能 |
| 扩展接口 | 支持 Unity3d 平台接口(官方接口) |
| | 支持 Flutter 平台接口(官方接口) |
| | 支持 API Cloud 接口(第三方合作) |
| | 支持 React Native 接口(第三方合作) |

1.3 Android/iOS 平台 Demo

1.3.1 Android 平台 Demo

| Android 平台 Demo | |
|-----------------|-------------------------------------|
| Demo 工程 | 描述 |
| SmartPlayerV2 | 大牛直播 SDK RTMP/RTSP 播放端 demo(接口展示更全) |

| | |
|--------------------------------------|--|
| SmartPublisherV2 | 内置推前后摄像头 SDK 调用 demo(接口展示更全)、涵盖 RTMP 推送、RTSP 推送、录像、内置轻量级 RTSP 服务功能 |
| SmartServiceCameraPublisherV2 | android 推屏(5.0 以上版本)和后台推摄像头 SDK 调用 demo (可配合推送 RTMP、推送 RTSP、录像、内置 RTSP 服务使用) |
| SmartServicePublisherV2 | 大牛直播 SDK 无纸化推屏/会议 demo (可配合推送 RTMP、推送 RTSP、录像、内置 RTSP 服务使用) |
| SmartEchoCancellationV2 | android 一对一互动 demo, 内置大牛直播推流端 SDK+播放端 SDK(支持回音消除) |
| SmartRelayDemoV2 | 大牛直播 SDK 转发 demo (支持拉取 RTSP、RTMP 流, 并转发到 RTMP 或 RTSP 服务器) |

1.3.2 iOS 平台 Demo

| iOS 平台 Demo | |
|----------------------------------|--|
| Demo 工程 | 描述 |
| SmartiOSPlayerV2 | 大牛直播 SDK RTMP/RTSP 播放端 demo(接口展示更全) |
| SmartiOSPublisherV2 | 内置推前后摄像头 SDK 调用 demo(接口展示更全)、涵盖 RTMP 推送、RTSP 推送、录像、内置轻量级 RTSP 服务功能 |
| SmartiOSScreenPublisherV2 | iOS 后台推屏(基于 ReplayKit)demo (可配合推送 RTMP、推送 RTSP、录像、内置 RTSP 服务使用) |
| SmartiOSRelayDemoV2 | 大牛直播 SDK 转发 demo (支持拉取 RTSP、RTMP 流, 并转发到 RTMP 或 RTSP 服务器) |
| SmartIOSEchoCancellation | 大牛直播 SDK 一对一互动 demo |

1.4 SDK 功能详解

1.4.1 Android/iOS 推送端 SDK

| Android/iOS 推送端 SDK 功能详解 | |
|--------------------------|--|
| 功能 | 功能描述 |
| 本地预览 | 支持本地前后置摄像头预览 |
| RTMP 推流 | 超低延时的 RTMP 协议直播推流 SDK (支持 RTMP 扩展 H.265 推送) |
| RTMP 扩展 H.265 | 支持 RTMP 扩展 H.265 推送 |
| RTSP 推流 | 超低延时的 RTSP 协议直播推流 SDK (支持 H.265 推送) |
| 音频格式 | Android 支持 AAC/SPEEX(RTMP)音频编码, iOS 支持 AAC 编码 |
| H.264 硬编码 | 支持 H.264 硬编码 |
| H.265 硬编码 | 支持 H.265 硬编码(iOS 需 A10 以上芯片, 11.0 以上版本) |
| 硬编码自适应 | 支持硬编码自适应, 如检测到硬编码不支持, 自动切换到软编 (iOS 如 H.265 硬编, 先切换到 H.264 硬编码, 如不支持再尝试 H.264 软编) |
| 软硬编码参数配置 | 支持 gop 间隔、帧率、bit-rate 设置 |
| 软编码参数配置 | 支持软编码 profile、软编码速度、可变码率设置 |
| 后台录屏直播(Android) | 支持后台录屏直播 (需 Android 5.0 以上版本) |
| 后台摄像头直播(Android) | 支持后台模式下摄像头直播 |
| 多实例推送 | 支持多实例推送(如同时推送屏幕/摄像头和外部数据) |
| 横竖屏推流 | 支持横屏、竖屏推流 |
| 多分辨率支持 | 支持摄像头或屏幕多种分辨率设置 |
| 事件回调 | 支持各种状态实时回调 |
| 水印(Android) | 支持 png 水印、文字水印 |
| RTMP 推送模式 | 支持 rtmp 推送 live record 模式设置 (需服务器支持) |
| RTSP 推送模式 | 支持 TCP、UDP 模式推送 |

| | |
|-------------|--|
| 镜像 | 支持前置摄像头实时镜像功能 |
| 复杂网络处理 | 支持断网重连等各种网络环境自动适配 |
| 动态码率 | 支持根据网络情况自动调整推流码率 |
| 实时静音 | 支持推送过程中，实时静音/取消静音 |
| 纯音频推流 | 支持仅采集音频流并发起推流功能 |
| 纯视频推流 | 支持特殊场景下的纯视频推流功能 |
| 降噪(Android) | 支持环境音、手机干扰等引起的噪音降噪处理、自动增益、VAD 检测 |
| 编码参数配置 | 支持 gop 间隔、帧率、bit-rate、软编码 profile、软编码速度设置 |
| 实时快照 | 支持推流过程中，实时快照 |
| 外部数据对接 | 支持编码前 YUV 数据对接 |
| | 支持 PCM 对接 |
| | 外部 H.264 数据对接 |
| | 外部 AAC/PCMA/PCMU/SPEEX 数据对接 |
| RTMP 加密方案 | 设置 rtmp 推送加密选项，可单独加密视频或音频 |
| | 设置 rtmp 加密算法，当前支持 aes128/aes192/aes256 和国标 sm4 |
| | 设置加密 Key |
| | 设置 rtmp 推送加密 IV(初始化向量) |

1.4.2 Android/iOS 播放端 SDK

| Android/iOS 播放端 SDK 功能详解 | |
|--------------------------|--|
| 功能 | 功能描述 |
| 支持播放协议 | 高稳定、超低延迟、业内首屈一指的 RTMP/RTSP 直播播放器 SDK |
| 多实例播放 | 支持同时播放多路 rtmp/rtsp 流 |
| 事件回调 | 支持网络状态、buffer 状态等回调 |
| 视频解码 | Android/iOS 支持 H.264/H.265 软解； Android/iOS 支持 H.264 硬解； |

| | |
|--------------------------|--|
| | Android 支持 H.265 硬解; Android 支持设置 Surface 模式硬解和普通模式硬解码。 |
| 音频解码 | RTMP/RTSP 支持 AAC/PCMA/PCMU, 此外 RTMP 还支持 Speex |
| RTSP 模式设置 | 支持 RTSP TCP/UDP 模式设置 |
| RTSP TCP/UDP 自动切换 | 支持 RTSP TCP/UDP 模式自动切换 |
| RTSP 超时设置 | 支持 RTSP timeout 设置, 单位: 秒 |
| RTSP 401 认证处理 | 支持上报 RTSP 401 事件, 如 URL 携带鉴权信息, 会自动处理 |
| 首屏秒开 | 支持直播的首屏秒开功能 |
| 低延迟模式 | 设置此模式后, 播放延迟更低 |
| 复杂网络处理 | 支持断网重连等各种网络环境自动适配 |
| 缓冲时间设置 | 支持 buffer time 设置 |
| 快速切换 URL | 支持播放过程中, 快速切换其他 URL |
| 音视频多种 render 机制(Android) | 视频: surfaceview/OpenGL ES |
| | 音频: AudioTrack/OpenSL ES |
| 实时静音 | 支持播放过程中, 实时静音/取消静音 |
| 实时快照 | 支持播放过程中截取当前播放画面 |
| 渲染角度 | 支持 0°, 90°, 180° 和 270° 四个视频画面渲染角度设置 |
| 渲染镜像 | 支持水平反转、垂直反转模式设置 |
| 实时下载速度更新 | 支持当前下载速度实时回调(支持设置回调时间间隔) |
| 音视频数据回调 | 支持解码后 YUV/RGB 数据回调 |
| | 支持 H.264/H.265 数据回调 |
| | 支持 AAC/PCMA/PCMU/SPEEX 数据回调 |
| RTMP 解密方案 | 支持 RTMP 推送端加密数据(AES/SM4)正常播放 |

1.4.3 Android/iOS 多路流媒体转发 SDK

| Android/iOS 多路流媒体转发 SDK | |
|-------------------------|------|
| 功能 | 功能描述 |

| | |
|----------------------|--|
| 标准功能 | 通过“ Android 播放端 SDK ”拉流，回调 H.264/AAC/SPEEX/PCMA/PCMU 数据，调用“ Android 推送端 SDK ”外置数据接口，实现转发 |
| 拉流音频转码 | 支持拉取的 RTMP/RTSP 的 PCMA/PCMU/SPEEX 音频格式 转 AAC 后再转发到 RTMP 服务器 |
| 多实例 | 支持同时转发多路 RTMP/RTSP 音视频流 |
| 本地预览 | 因支持“ Android 播放端 SDK ”功能，支持转发过程中，随时本地预览 |
| 拉流音频调节 | 支持拉取的 RTMP/RTSP 流静音 |
| 切换转发源 | 支持转发过程中，随时切换拉流的数据源(源 URL 变化，推流 URL 不变)，播放端 SDK 无感知(还是同一个拉流 URL)低延迟播放切换后数据源 |
| 逻辑分离 | 支持播放、录像、转发逻辑完全分离，三者可随意组合或单独使用 |
| H.265(HEVC)支持 | 业内为数不多支持 RTSP/RTMP H.265 转 RTMP 推送 的 SDK(提供配套 RTMP 扩展 H.265 服务器); |

1.4.4 Android/iOS 录像 SDK

| Android/iOS 录像 SDK | |
|--------------------|--|
| 功能 | 功能描述 |
| 模块覆盖 | 推送端录像 |
| | 播放端录像 |
| 多实例 | 支持同时录制多路 RTMP/RTSP 音视频流 |
| 更智能的录像模式 | 大牛直播录像 SDK 不同于普通录像接口，更智能，和推送、播放、转发、内置轻量级 RTSP 服务 SDK 功能完全分离，支持随时录像 |
| 录制格式 | MP4 |
| 音频格式 | AAC |
| 视频格式 | H.264/H.265(HEVC) |
| 音频转码后录像 | 支持音频(PCMU/PCMA, Speex 等)转 AAC 后再录像 |
| 录像参数设置 | 支持设置单个录像文件大小、录像路径等，并支持纯音频、纯视频、音视频录制模式 |

| | |
|----------|--|
| 切换数据源 | 支持播放端录像过程中，相同参数配置的多个 URL 间实时切换，录制到同一个 MP4 文件 |
| Event 回调 | 从开始录像，到录像结束均有 event callback 上来，网络堵塞、音视频同步均做了非常友好的处理 |

1.4.5 Android/iOS 一对一互动 SDK

| Android/iOS 一对一互动 SDK | |
|-----------------------|--|
| 功能 | 功能描述 |
| 标准功能 | 支持推送端常规功能 |
| | 支持播放端常规功能 |
| 回音消除 | 支持回音消除 |
| 降噪 | Android 支持环境音、手机干扰等引起的噪音降噪处理、自动增益、VAD 检测 |

1.4.6 Android/iOS SEI 扩展数据发送/接收 SDK

| Android/iOS SEI 扩展数据发送/接收 SDK | |
|-------------------------------|--|
| 功能 | 功能描述 |
| 适用场景 | 公告广播： 推送将相对/绝对时间戳/时间/公告内容发到播放端，播放端实时接收消息并做相应的逻辑处理。 |
| | 冲顶大会： 推流端实时将题目分发到播放端，借助于大牛直播 SDK 低延迟特性，轻松实现“音-画-题”同步接收； |
| | 直播： 推流端将歌词/字幕分发到播放端，播放端实时绘制出歌词 |
| | 应急指挥/单兵： 推送端将 GIS 信息/现场采集到的数据实时写入并分发到播放端 |
| | 在线教育： 推流端将激光笔和涂鸦操作分发到播放端，播放端实时划圈划线，实现特定特效 |

| | |
|------|--|
| 功能支持 | 支持内置 RTSP 服务 SDK 携带 SEI 扩展信息(H.264) |
| | 支持 RTMP 推送 SDK 携带 SEI 扩展信息(H.264) |
| | 推送端支持发送自定义用户数据(如自定义 utf8 字符串) |
| | 推送端支持发送二进制数据 |
| | RTSP/RTMP 直播播放端 SDK 支持 utf8 文本、二进制、和原 SEI 数据解析 |

2 Android SDK 使用说明

2.1 Android 推流端 SDK 说明

2.1.1 系统要求

- SDK 支持 Android 4.4 及以上版本;
- 支持的 CPU 架构: armv7, arm64, x86, x86_64。

2.1.2 准备工作

- 确保 `SmartPublisherJniV2.java` 放到 `com.daniulive.smartpublisher` 包名下(可在其他包名下调用);
- `Smartavengine.jar` 加入到工程;
- 拷贝 `SmartPublisherV2\app\src\main\jniLibs\armeabi-v7a`、`SmartPublisherV2\app\src\main\jniLibs\arm64-v8a`、`SmartPublisherV2\app\src\main\jniLibs\x86` 和 `SmartPublisherV2\app\src\main\jniLibs\x86_64` 下 `libSmartPublisher.so` 到工程;
- 添加相关权限:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" ></uses-permission>
<uses-permission android:name="android.permission.INTERNET" ></uses-permission>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

- Load 库:

```
static {
    System.loadLibrary("SmartPublisher");
}
```

- `build.gradle` 配置 32/64 位库:

```
splits {
  abi {
    enable true
    reset()

    // Specifies a list of ABIs that Gradle should create APKs for
    //include "armeabi"
    include 'armeabi-v7a', 'arm64-v8a', 'x86', 'x86_64' //select ABIs to build APKs for

    // Specify that we do not want to also generate a universal APK that includes all ABIs
    universalApk true
  }
}
```

- 如需集成到自己系统测试，请用大牛直播 SDK 的 app name(不然集成提示 license failed)，正式授权版按照授权 app name 正常使用即可：
- 如何改 app-name:
strings.xml 做以下修改：
<string name="app_name">SmartPublisherSDKDemo</string>

2.1.3 SDK 接口详解

| Android 推送端 SDK 接口详解 | | |
|----------------------|--------------------|--|
| 调用描述 | 接口 | 接口描述 |
| 最先调用，如成功返回推送实例 | SmartPublisherOpen | ctx: 上下文信息； Audio_opt: 0: 不发布 audio； 1: 发布 audio； 2: 对接外部编码后的 audio 数据 (AAC) video_opt: 0: 不发布 video； 1: 发布 video； 2: 对接外部编码后的 video 数据 (H.264) |

| | | | |
|-----------------|------------------------------------|--|---|
| | | | 宽高信息。 |
| Event 回调 | SetSmartPublisherEventCallbackV2 | | 设置 event callback |
| 硬编码设置 | SetSmartPublisherVideoHWEncoder | | 检测是否支持 H.264 硬编码，如果返回 0，则支持，否则自动采用软编码 |
| | SetSmartPublisherVideoHvcHWEncoder | | 检测是否支持 H.265(HEVC)硬编码，如果返回 0，则支持，否则自动采用软编码 |
| 水印 | 文字水印 | SmartPublisherSetFontWatermark | 设置文字水印 支持设置文字的 size、水印位置、是不是添加时间 |
| | 文字字体库路径设置 | SmartPublisherSetTextWatermarkFontFileName | 为防止一些手机字体库缺失，提供设置字体库(*.ttf 格式)路径接口 |
| | 图片水印 | SmartPublisherSetPictureWatermark | 设置图片水印 支持设置图片水印(*.png)路径、位置、宽高 |
| 视频参数配置 | 软编码可变码率 | SmartPublisherSetSwVBRMode | 设置软编码可变码率，可变码率下，相邻帧之间变化不大时码率更低 |
| | GOP 间隔(关键帧) | SmartPublisherSetGopInterval | 设置推送端 GOP 间隔，一般建议在帧率的 1~3 倍，如不设置，用底层默认值 |
| | 软编码码率设置 | SmartPublisherSetSWVideoBitRate | 设置软编码视频 bit-rate, 最大码流一般是平均码流的 2 倍，如不设置，用底层计算的默认值 |

| | | | |
|-----------|---------------|--|--|
| | 帧率 | SmartPublisherSetFPS | 设置 fps, 如不设置, 用底层默认值 |
| | 软编码视频 Profile | SmartPublisherSetSWVideoEncoderProfile | 设置软编码模式下的 video encoder profile, 默认 baseline profile |
| | 软编码编码速度 | SmartPublisherSetSWVideoEncoderSpeed | 设置软编码编码速度, 设置范围 (1,6), 1 最快, 6 最慢, 默认是 6 |
| 视频处理 | 视频裁剪 | SmartPublisherSetClippingMode | 设置裁剪模式(仅用于 640*480 分辨率, 裁剪主要用于移动端宽高适配), 如不设置, 不做裁剪 |
| | 视频镜像 | SmartPublisherSetMirror | 镜像模式: 播放端和推送端本地回显方向显示一致 |
| | 实时快照设置 | SmartPublisherSaveImageFlag | 设置是否启用快照功能 |
| | 实时快照 | SmartPublisherSaveCurImage | 推送或录像过程中, 根据设置路径和文件名, 实时快照 |
| 音频配置 | 音频编码类型 | SmartPublisherSetAudioCodecType | 设置编码类型, 默认 AAC 编码, type 设置为 2 时, 启用 speex 编码(码率更低) |
| | SPEEX 编码质量 | SmartPublisherSetSpeexEncoderQuality | 设置 speex 编码质量, 数值越大, 质量越高, 范围 (0,10), 默认 8 |
| 音频处理 | 噪音抑制 | SmartPublisherSetNoiseSuppression | 噪音抑制开启后, 可去除采集端背景杂音 |
| | 增益控制 | SmartPublisherSetAGC | 设置自动增益控制, 保持声音稳定 |
| | 实时静音 | SmartPublisherSetMute | 设置实时静音、取消静音 |
| RTMP 推送模式 | | SetRtmpPublishingType | 设置 rtmp publisher 类型, 0: live, 1: record, 需服务器支 |

| | | | |
|------------------|-----------------------|--|--|
| | | | 持 |
| | RTMP 推送 URL 设置 | SmartPublisherSetURL | 设置 RTMP 推送 url |
| | RTMP 加密 | SetRtmpEncryptionOption | 设置 rtmp 推送加密选项, 可单独加密视频或音频 |
| | | SetRtmpEncryptionAlgorithm | 设置 rtmp 加密算法, 当前支持 aes 和国标 sm4. 1 为 aes, 2 为 sm4 |
| | | SetRtmpEncryptionKey | 设置 rtmp 推送加密密钥, 支持 AES128、AES192、AES256 和 SM4 国密 |
| | | SetRtmpEncryptionIV | 设置 rtmp 推送加密 IV(初始化向量), 这个接口不调用的话, 将使用默认 IV |
| 编码前实时视频数据 | 自带摄像头数据 | SmartPublisherOnCaptureVideoData | 传递实时采集的 video 数据 (编码前) |
| | YUV(I420) | SmartPublisherOnCaptureVideoI420Data | 第三方 YUV (I420) 接口 |
| | RGBA 数据 | SmartPublisherOnCaptureVideoRGBAData | 第三方 RGBA 数据 |
| | ABGR 垂直翻转数据 | SmartPublisherOnCaptureVideoABGRFlipVerticalData | ABGR flip vertical(垂直翻转) 数据(Demo 中用于传递屏幕数据) |
| | RGB565 数据 | SmartPublisherOnCaptureVideoRGB565Data | RGB565 data |
| | 裁剪过的 RGBA 数据 | SmartPublisherOnCaptureVideoClippedRGBAData | 投递裁剪过的 RGBA 数据 |
| 编码前实时音频数据 | | SmartPublisherOnPCMDData | 实时 PCM 数据 |
| | | SmartPublisherOnFarEndPCMDData | 实时传递远端 PCM 数据(可用于互动级的回音消除处理) |
| 编码后数据回 | 编码后 AAC 回调 | SmartPublisherSetAudioEncodedDataCallback | 设置编码后的 AAC 数据回调 |

| | | | |
|-----------------|--------------|---|---|
| 调 | 编码后 H.264 回调 | SmartPublisherSetVideoEncodedDataCallback | 设置编码后的 H.264 数据回调 |
| 开始 RTMP 推送 | | SmartPublisherStartPublisher | 开始 RTMP 推送 |
| 停止 RTMP 推送 | | SmartPublisherStopPublisher | 停止 RTMP 推送 |
| 设置推送 RTSP 传输方式 | | SetPushRtspTransportProtocol | transport_protocol: 1 表示 UDP 传输 rtp 包; 2 表示 TCP 传输 rtp 包 |
| 设置推送 RTSP 的 URL | | SetPushRtspURL | 设置推送 RTSP 的 URL |
| 开始 RTSP 推送 | | StartPushRtsp | 启动推送 RTSP 流 |
| 停止 RTSP 推送 | | StopPushRtsp | 停止推送 RTSP 流 |
| 关闭推送实例 | | SmartPublisherClose | 结束时必须调用 close 接口释放资源 |

2.2 Android 推流端 SDK Event 回调说明

由于 Android 录像 SDK、内置轻量级 RTSP 服务 SDK 二者和推送端 SDK 可组合使用，相关 Event 同步更新在 Android 推送端 SDK(如下图)：

| Android 推送端 SDK Event 回调说明 | |
|---|--|
| 事件 ID | 事件描述 |
| EVENT_DANIULIVE_ERC_PUBLISHER_STARTED | 开始推送 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTING | 连接中 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTION_FAILED | 连接失败 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTED | 连接成功 |
| EVENT_DANIULIVE_ERC_PUBLISHER_DISCONNECTED | 连接断开 |
| EVENT_DANIULIVE_ERC_PUBLISHER_STOP | 停止推送 |
| EVENT_DANIULIVE_ERC_PUBLISHER_RECORDER_START_NEW_FILE | 开始一个新的录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PUBLISHER_ONE_RECORDER_FILE_FINISHED | 已生成一个录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PUBLISHER_SEND_DELAY | 发送时延 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CAPTURE_IMAGE | 实时快照 |
| EVENT_DANIULIVE_ERC_PUBLISHER_RTSP_URL | 内置 RTSP 服务 SDK 生成的 RTSP URL |
| EVENT_DANIULIVE_ERC_PUSH_RTSP_SERVER_RESPONSE_STATUS_CODE | 推送 RTSP 时服务端相应的 status code 上报， 目前只上报 401 |
| EVENT_DANIULIVE_ERC_PUSH_RTSP_SERVER_NOT_SUPPORT | 推送 RTSP 时服务器不支持 RTSP 推送 |

2.3 Android 推送端录像 SDK 说明

大牛直播推送端录像 SDK，和推送、内置轻量级 RTSP 服务、转发模块完全隔离，可以随意组合使用。

- ◇ 文件格式：MP4；
- ◇ 涉及相关库：libSmartPublisher.so
- ◇ 头文件：SmartPublisherJniV2.java
- ◇ Jar：smartavengine.jar

以下是接口详解：

| Android 推送端录像 SDK 接口详解 | | | |
|------------------------|-----------------------------|--------------------------------------|--|
| 调用描述 | 接口 | 接口描述 | |
| 录像设置 | 是否录像 | SmartPublisherSetRecorder | 设置是否启用本地录像 |
| | 创建录像目录 | SmartPublisherCreateFileDirectory | 创建录像文件目录 |
| | 设置录像目录 | SmartPublisherSetRecorderDirectory | 设置录像文件目录 |
| | 设置录像文件大小 | SmartPublisherSetRecorderFileMaxSize | 设置每个录像文件的大小，比如 100M，超过这个大小后，会自动生成下一个录像文件 |
| 开始录像 | SmartPublisherStartRecorder | 开始录像 | |
| 停止录像 | SmartPublisherStopRecorder | 停止录像 | |

2.4 Android 转发 SDK 说明

Android 转发 SDK，简单来说，播放端 SDK 拉取 RTSP/RTMP 流，并回调编码后的音视频数据到上层，然后，调用我们的推送端 SDK，通过推送端 SDK 扩展数据接口，完成 RTMP 数据转发，整个过程由于不涉及解码、重编码，支持多路转发，超低延迟和低资源占用。

- ◇ 涉及相关库：libSmartPublisher.so libSmartPlayer.so
- ◇ 头文件：SmartPlayerJniV2.java SmartPublisherJniV2.java
- ◇ Jar：smartavengine.jar

以下是接口详解：

| Android 转发 SDK 接口详解 | | |
|---------------------|---|--|
| 调用描述 | 接口 | 接口描述 |
| 创建推流实例和播放实例 | 从播放端拉取 RTSP/RTMP 流，并回调到上层，调用推送端数据转发接口，实现转发逻辑 | 注意：推送端调用 SmartPublisherOpen 时： Audio_opt 填 2： video_opt:填 2. |
| 播放端开始拉流 | SmartPlayerStartPullStream | 播放端开始拉流，用于音视频数据转发 |
| 播放端停止拉流 | SmartPlayerStopPullStream | 播放端停止拉流，用于音视频数据转发 |
| 播放端视频回调 | SmartPlayerSetVideoDataCallback | 回调视频数据 |
| 播放端音频回调 | SmartPlayerSetAudioDataCallback | 回调音频数据 |
| 音频转码 | 设置拉流时音频转 AAC 编码的开关 | 设置拉流时音频转 AAC 编码的开关 (PCMA/PCMU/SPEEX 转 AAC) |
| 视频转发 | SmartPublisherPostVideoEncodedData/SmartPublisherPostVideoEncodedDataV2 | 设置编码后视频数据(H.264) |
| 音频转发 | SmartPublisherPostAudioEncodedData/SmartPublisherPostAudioEncodedDataV2 | 设置音频数据(AAC/PCMA/PCMU/SPEEX) |

2.5 Android 内置轻量级 RTSP 服务 SDK 说明

为满足内网无纸化/电子教室等内网超低延迟需求，避免让用户配置单独的服务器，大牛直播 SDK 在推送端发布了轻量级 RTSP 服务 SDK。

简单来说，之前推送端 SDK 支持的功能，内置轻量级 RTSP 服务 SDK 后，功能继续支持。

内置轻量级 RTSP 服务后，延迟更低，体验更好。

✧ 涉及相关库：libSmartPublisher.so

✧ 头文件：SmartPlayerJniV2.java SmartPublisherJniV2.java

✧ Jar：smartavengine.jar

以下是接口详解：

| Android 内置轻量级 RTSP 服务 SDK 接口详解 | | |
|---|--------------------------------------|---|
| 调用描述 | 接口 | 接口描述 |
| 常规功能 | Android RTMP 推送端所有功能依然有效，亦可同时推送 RTMP | |
| <i>SmartRTSPServerSDK</i> | | |
| 初始化 RTSP Server | InitRtspServer | Init rtsp server(和 UnInitRtspServer 配对使用，即便是启动多个 RTSP 服务，也只需调用一次 InitRtspServer，请确保在 OpenRtspServer 之前调用) |
| 创建一个 rtsp server | OpenRtspServer | 创建一个 rtsp server，返回 rtsp server 句柄 |
| 设置端口 | SetRtspServerPort | 设置 rtsp server 监听端口，在 StartRtspServer 之前必须要设置端口 |
| 设置鉴权用户名、密码 | SetRtspServerUserNamePassword | 设置 rtsp server 鉴权用户名和密码，这个可以不设置，只有需要鉴权的再设置 |
| 获取 rtsp server 当前会话数 | GetRtspServerClientSessionNumbers | 获取 rtsp server 当前的客户会话数，这个接口必须在 StartRtspServer 之后再调用 |
| 启动 rtsp server | StartRtspServer | 启动 rtsp server |
| 停止 rtsp server | StopRtspServer | 停止 rtsp server |
| 关闭 rtsp server | CloseRtspServer | 关闭 rtsp server |
| UnInit rtsp server | UnInitRtspServer | UnInit rtsp server(和 InitRtspServer 配对使用，即便是启动多个 RTSP 服务，也只需调用一次 UnInitRtspServer) |
| <i>SmartRTSPServerSDK 供 Publisher 调用的接口</i> | | |
| 设置 rtsp 的流名称 | SetRtspStreamName | 设置 rtsp 的流名称 |
| 给要发布的 rtsp 流设置 | AddRtspStreamServer | 给要发布的 rtsp 流设置 rtsp server， |

| | | |
|-------------------|-----------------------|--|
| rtsp server | | 一个流可以发布到多个 rtsp server 上, rtsp server 的创建启动请参考 OpenRtspServer 和 StartRtspServer 接口 |
| 清除设置的 rtsp server | ClearRtspStreamServer | 清除设置的 rtsp server |
| 启动 rtsp 流 | StartRtspStream | 启动 rtsp 流 |
| 停止 rtsp 流 | StopRtspStream | 停止 rtsp 流 |

2.6 Android SEI 扩展数据发送/接收 SDK

大牛直播 SDK 支持推送端通过 H.264 SEI 信息扩展，实时传输文本/二进制数据信息，播放端做相应解析和回显。

适用于公告广播、冲顶大会、直播、应急指挥/单兵在线教育等各种场景。

目前使用 sei 机制发送用户自定数据到播放端，这种机制有可能会丢失数据，所以这种方式不保证接收端一定能收到。

优势：能和视频保持同步，虽然有可能丢失，但一般的需求都满足了。

目前提供两种发送方式 第一种发送二进制数据，第二种发送 utf8 字符串。

✧ 涉及相关库：libSmartPublisher.so libSmartPlayer.so

✧ 头文件：SmartPublisherJniV2.java SmartPlayerJniV2.java

✧ Jar：smartavengine.jar

以下是接口详解：

| Android SEI 扩展数据发送/接收 SDK 接口详解 | | |
|--------------------------------|---|--|
| 调用描述 | 接口 | 接口描述 |
| 发送端(置于推送端 SDK) | SmartPublisherSetPostUserDataQueueMaxSize | 设置发送队列大小，为保证实时性，默认大小为 3，必须设置一个大于 0 的数 |
| | SmartPublisherClearPostUserDataQueue | 清空用户数据队列，有些情况可能会用到，比如发送队列里面有 4 条消息再等待发送，又想把最新的消息快速发出去，可以先清除掉正在排队消息，再调用 PostUserXXX |
| | SmartPublisherPostUserData | 发送二进制数据 NOTE: 1. 目前数据大小限制在 256 个字节以内，太大可能会影响视频传输，如果有特殊需求，需要增大限制，请联系我们 2. 如果积累的数据超过了设置的队列大小，之前的队头数据将被丢弃 3. 必须再调用 StartPublisher 之后再发送数据 |

| | | |
|----------------|--------------------------------------|---|
| | SmartPublisherPostUserUTF8StringData | <p>发送 utf8 字符串</p> <p>NOTE:</p> <ol style="list-style-type: none"> 1. 字符串长度不能超过 256, 太大可能会影响视频传输, 如果有特殊需求, 需要增大限制, 请联系我们 2. 如果积累的数据超过了设置的队列大小, 之前的队头数据将被丢弃 3. 必须再调用 StartPublisher 或 RTSP 流发布之后再发送数据 |
| 接收端(置于播放端 SDK) | SmartPlayerSetUserDataCallback | Set user data Callback. |
| | UserDataCallback() | 参见 SmartPlayer.java 实现 |
| | SmartPlayerSetSEIDataCallback | Set SEI data Callback. |
| | SEIDataCallback() | 参见 SmartPlayer.java 实现 |

2.7 Android 播放端 SDK 说明

2.7.1 系统要求

- SDK 支持 Android 4.4 及以上版本;
- 支持的 CPU 架构: armv7, arm64, x86, x86_64。

2.7.2 集成说明

- 确保 **SmartPlayerJniV2.java** 放到 **com.daniulive.smartplayer** 包名下(可在其他包名下调用);
- Smartavengine.jar 加入到工程;
- 拷贝 SmartPlayerV2\app\src\main\jniLibs\armeabi-v7a、SmartPlayerV2\app\src\main\jniLibs\arm64-v8a、SmartPlayerV2\app\src\main\jniLibs\x86 和 SmartPlayerV2\app\src\main\jniLibs\x86_64 下 libSmartPlayer.so 到工程;
- Load 库:

```
static {
    System.loadLibrary("SmartPlayer");
}
```

- 添加相关权限:

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" ></uses-permission>
<uses-permission android:name="android.permission.INTERNET" ></uses-permission>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

- build.gradle 配置 32/64 位库:

```
splits {
    abi {
        enable true
        reset()

        // Specifies a list of ABIs that Gradle should create APKs for
        //include "armeabi"
        include 'armeabi-v7a', 'arm64-v8a', 'x86', 'x86_64' //select ABIs to build APKs for

        // Specify that we do not want to also generate a universal APK that includes all ABIs
        universalApk true
    }
}
```

```
}
}
```

- 如需集成到自己系统测试，请用大牛直播 SDK 的 app name(不然集成提示 license failed)，正式授权版按照授权 app name 正常使用即可；
- 如何改 app-name:
 - strings.xml 做以下修改:
 - <string name="app_name">SmartPlayerSDKDemo</string>

2.7.3 SDK 接口详解

| Android 播放端 SDK 接口详解 | | | |
|----------------------------|---------------------------------------|--|--|
| 调用描述 | 接口 | 接口描述 | |
| 最先调用, 如成功返回 推送实例 | SmartPlayerOpen | player 初始化, 设置上下文信息, 返回 player 句柄 | |
| Event 回调 | SetSmartPlayerEventCallbackV2 | 设置 event callback | |
| 硬解码设置 | SetSmartPlayerVideoHWDecoder | 设置是否用硬解码播放, 如硬解码不支持, 自动适配到软解码 | |
| 设置播放的 surface | SmartPlayerSetSurface | 设置播放的 surface, 如果为 null, 则播放纯音频 | |
| 设置视频硬解码下 Mediacodec 自行绘制模式 | SmartPlayerSetHWRenderMode | 此种模式下, 硬解码兼容性和效率更好, 回调 YUV/RGB 和快照功能将不可用 | |
| 更新硬解码 surface | SmartPlayerUpdateHWRenderSurface | 设置更新硬解码 surface | |
| 音频回调 | YUV/RGB | SmartPlayerSetExternalRender | 提供解码后 YUV/RGB 数据接口, 供用户自己 render 或进一步处理(如视频分析) |
| | Audio | SmartPlayerSetExternalAudioOutput | 回调 audio 数据到上层(供二次处理之用) |
| audio 输出类型 | SmartPlayerSetAudioOutputType | 如果 use_audiotrack 设置为 0, 将会自动选择输出设备, 如果设置为 1, 使用 audiotrack 模式 | |
| Video 输出类型 | NTRenderer.CreateRenderer (上层 demo 内) | 第二个参数, 如果是 true, 用 OpenGL ES 绘制, false 则用默认 surfaceView | |

| | | | |
|----------------------|----------|------------------------------------|---|
| 播放模式 | 缓冲时间设置 | SmartPlayerSetBuffer | 设置播放端缓存数据 buffer，单位：毫秒，如不需 buffer，设置为 0 |
| | 首屏秒开 | SmartPlayerSetFastStartup | 设置快速启动后，如果 CDN 缓存 GOP，实现首屏秒开 |
| | 低延迟模式 | SmartPlayerSetLowLatencyMode | 针对类似于直播娃娃机等期待超低延迟的使用场景，超低延迟播放模式下，延迟可达到 200~400ms |
| | 快速切换 URL | SmartPlayerSwitchPlaybackUrl | 快速切换播放 url，快速切换时，只换播放 source 部分，适用于不同数据流之间，快速切换（如娃娃机双摄像头切换或高低分辨率流切换） |
| RTSP TCP/UDP 模式设置 | | SmartPlayerSetRTSPTcpMode | 设置 RTSP TCP/UDP 模式，如不设置，默认 UDP 模式 |
| RTSP 超时时间设置 | | SmartPlayerSetRTSPTimeout | 设置 RTSP 超时时间，timeout 单位为秒，必须大于 0 |
| 设置 RTSP TCP/UDP 自动切换 | | SmartPlayerSetRTSPAutoSwitchTcpUdp | 对于 RTSP 来说，有些可能支持 rtp over udp 方式，有些可能支持使用 rtp over tcp 方式，为了方便使用，有些场景下可以开启自动尝试切换开关，打开后如果 udp 无法播放，sdk 会自动尝试 tcp，如果 tcp 方式播放不了，sdk 会自动尝试 udp。 |
| 实时静音 | | SmartPlayerSetMute | 实时静音 |
| 视频镜像旋转 | 旋转 | SmartPlayerSetRotation | 设置顺时针旋转，注意除了 0 度之外，其他角度都会额外消耗性能，当前支持 0 度，90 度，180 度，270 度 旋转 |
| | 水平反转 | SmartPlayerSetFlipHorizontal | 设置视频水平反转 |
| | 垂直反转 | SmartPlayerSetFlipVertical | 设置视频垂直反转 |
| 设置 URL | | SmartPlayerSetUrl | 设置需要播放或录像的 RTMP/RTSP url |
| RTMP 解密 | | SmartPlayerSetKey | 设置解密 key，目前只用来解密 rtmp 加密流 |

| | | |
|--------|-----------------------------------|--------------------------------|
| | SmartPlayerSetDecryptionIV | 设置解密向量，目前只用来解密 rtmp 加密流 |
| 开始播放 | SmartPlayerStartPlay | 开始播放 RTSP/RTMP 流 |
| 停止播放 | SmartPlayerStopPlay | 停止播放 RTSP/RTMP 流 |
| 关闭播放实例 | SmartPlayerClose | 结束时必须调用 close 接口释放资源 |

2.8 Android 播放端 SDK Event 回调说明

由于 Android 播放录像 SDK 和播放端 SDK 可组合使用，相关 Event 同步更新在 Android 播放端 SDK(如下图)：

| Android 播放端 SDK Event 回调说明 | |
|---|---|
| 事件 ID | 事件描述 |
| EVENT_DANIULIVE_ERC_PLAYER_STARTED | 开始播放 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTING | 播放端连接中 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTION_FAILED | 播放端连接失败 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTED | 播放端连接成功 |
| EVENT_DANIULIVE_ERC_PLAYER_DISCONNECTED | 播放端连接断开 |
| EVENT_DANIULIVE_ERC_PLAYER_STOP | 停止播放 |
| EVENT_DANIULIVE_ERC_PLAYER_RESOLUTION_INFO | 返回视频宽、高信息 |
| EVENT_DANIULIVE_ERC_PLAYER_NO_MEDIADATA_RECEIVED | 收不到媒体数据(可能是 URL 错误) |
| EVENT_DANIULIVE_ERC_PLAYER_SWITCH_URL | 快速切换 URL |
| EVENT_DANIULIVE_ERC_PLAYER_RECORDER_START_NEW_FILE | 开始一个新的录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PLAYER_ONE_RECORDER_FILE_FINISHED | 已生成一个录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PLAYER_CAPTURE_IMAGE | 播放端实时快照 |
| EVENT_DANIULIVE_ERC_PLAYER_START_BUFFERING | 开始缓冲数据 |

| | |
|---|------------------------|
| EVENT_DANIULIVE_ERC_PLAYER_BUFFERING | 缓冲中(param1 参数会返回缓冲百分比) |
| EVENT_DANIULIVE_ERC_PLAYER_STOP_BUFFERING | 停止缓冲数据 |
| EVENT_DANIULIVE_ERC_PLAYER_DOWNLOAD_SPEED | 返回当前 RTSP/RTMP 流实时下载速度 |
| EVENT_DANIULIVE_ERC_PLAYER_RTSP_STATUS_CODE | RTSP 收到错误码，可能是用户名、密码不对 |
| EVENT_DANIULIVE_ERC_PLAYER_NEED_KEY | RTMP 加密流，请设置播放需要的 Key |
| EVENT_DANIULIVE_ERC_PLAYER_KEY_ERROR | RTMP 加密流，Key 错误，请重新设置 |

2.9 Android 播放端录像 SDK 说明

大牛直播播放端录像 SDK，和播放 SDK、转发模块完全隔离，可以随意组合使用，在复杂网络环境适应、时间戳跳变都做了很好的兼容，也可能是市面上最好用的 RTSP/RTMP 流录像模块(可支持多路录像)。

- ✧ 文件格式：MP4；
- ✧ 涉及相关库：libSmartPlayer.so
- ✧ 头文件：SmartPlayerJniV2.java
- ✧ Jar：smartavengine.jar

以下是接口详解：

| Android 播放端录像 SDK 接口详解 | | | |
|------------------------|----------|-----------------------------------|--|
| 调用描述 | | 接口 | 接口描述 |
| 录像设置 | 创建录像目录 | SmartPlayerCreateFileDirectory | 创建录像文件目录 |
| | 设置录像目录 | SmartPlayerSetRecorderDirectory | 设置录像文件目录 |
| | 设置录像文件大小 | SmartPlayerSetRecorderFileMaxSize | 设置每个录像文件的大小，比如 100M，超过这个大小后，会自动生成下一个录像文件 |

| | | | |
|-------------|-------------|--|--|
| | <p>音频转码</p> | <p>SmartPlayerSetRecorderAudioTranscode AACs</p> | <p>设置拉流时音频转 AAC 编码的开关 aac 比较通用, sdk 增加其他音频编码(比如 speex, pcmu, pcma 等)转 aac 的功能.</p> |
| | <p>录制视频</p> | <p>SmartPlayerSetRecorderVideo</p> | <p>设置是否录视频, 默认的话, 如果视频源有视频就录, 没有就不录, 但有些场景下可能不想录制视频, 只想录音频, 所以增加个开关</p> |
| | <p>录制音频</p> | <p>SmartPlayerSetRecorderAudio</p> | <p>设置是否录音频, 默认的话, 如果视频源有音频就录, 没有就不录, 但有些场景下可能不想录制音频, 只想录视频, 所以增加个开关</p> |
| <p>开始录像</p> | | <p>SmartPlayerStartRecorder</p> | <p>开始录像</p> |
| <p>停止录像</p> | | <p>SmartPlayerStopRecorder</p> | <p>停止录像</p> |

3 iOS SDK 使用说明

3.1 iOS 推流端 SDK 说明

3.1.1 系统要求

- SDK 支持 iOS 7.0 及以上版本(建议在 8.0 及以上版本使用);
- 支持的 CPU 架构: armv7, arm64, i386, x86_64。

3.1.2 准备工作

- 相关库: libSmartPublisherSDK.a
- 相关头文件:
 - nt_common_media_define.h(如需转发或第三方数据对接)
 - SmartRTSPSeverSDK.h(内置轻量级 RTSP 服务 SDK 使用)
 - nt_event_define.h(相关 Event 定义)
 - SmartPublisherSDK.h(推送端 SDK)
- 如集需要引入的 framework
 - libz.tbd
 - libc++.tbd
 - libstdc++.tbd
 - Libz.tbd
 - libiconv.tbd
 - Accelerate.framework
 - AudioToolBox.framework
 - AssetsLibrary.framework
 - AVFoundation.framework
 - CoreMedia.framework
 - Foundation.framework
 - UIKit.framework
 - VideoToolBox.framework
- 如需集成到自己系统测试, 请用大牛直播的 app name:
Info.plist-->右键 Open As-->Source Code
添加或者编辑
<key>CFBundleName</key>
<string>SmartiOSPublisher</string>
- 快照添加到“照片”权限:

```
Info.plist-->右键 Open As-->Source Code
添加
<key>NSPhotoLibraryUsageDescription</key>
<string>1</string>
```

3.1.3 SDK 接口详解

| Android 推送端 SDK 接口详解 | | |
|----------------------|--------------------|--|
| 调用描述 | 接口 | 接口描述 |
| 最先调用，如成功返回 推送实例 | SmartPublisherInit | <p>Audio_opt:</p> <ul style="list-style-type: none"> 0: 不推送音频 1: 推送 SDK 内部采集的音频 2: 推送外部编码后音频(目前支持 AAC/PCMA/PCMU/SPEEX 宽带) 3: 推送外部编码前音频数据 (CMSampleBufferRef 类型), 数据传递对应接口: SmartPublisherPostAudioSampleBuffer 4: 外部 PCM 数据 <p>video_opt:</p> <ul style="list-style-type: none"> 0: 不推送视频 1: 推送 SDK 内部采集的视频 2: 推送外部编码后视频(目前支持 H.264), 数据格式: 000000167.... 3: 推送外部编码前视频数据 (CMSampleBufferRef 类型), 数据传递对应接口: SmartPublisherPostVideoSampleBuffer 4: 推送外部编码前 yuv420 视频数据, 数据传递对应接口: SmartPublisherSetExternalYuvData 5: 推送外部编码前 BGRA 视频数据 (alpha 通道不使用), 数据传递对应接口: SmartPublisherSetExternalBGRADa 6: 推送外部编码前 ARGB 视频数据 |

| | | | |
|----------------------------|--|--|---|
| | | | (alpha 通道不使用), 数据传递对应接口: <code>SmartPublisherSetExternalARGBData</code> |
| Event 回调 | <code>handleSmartPublisherEvent</code> | | 处理 event callback |
| 横竖屏推送设定 | <code>SmartPublisherSetPublishOrientation</code> | | 设置横屏、竖屏推送 |
| 设置视频编码类型 | <code>SmartPublisherSetVideoEncoderType</code> | | 设置视频编码类型(H.264/H.265, 软编码还是硬编码) <code>encoderType: 1: H.264, 2: H.265 编码</code> <code>isHwEncoder: YES: 硬编码 NO: 软编码</code> |
| 设置音频编码类型(目前仅支持 AAC) | <code>SmartPublisherSetAudioEncoderType</code> | | 设置音频编码类型(目前仅支持 AAC) <code>encoderType: 1: AAC</code> <code>isHwEncoder: YES: 硬编码 NO: 软编码</code> |
| 软编码可变码率 | <code>SmartPublisherSetSwVBRMode</code> | | Set software encode vbr mode(软编码可变码率). <code>is_enable_vbr: if false: NOT enable vbr mode, true: enable vbr</code> <code>video_quality: vbr video quality, range with (1,50), default 23</code> <code>vbr_max_kbitrate: vbr max encode bit-rate(kbps)</code> |
| 视频参数配置 | 帧率 | <code>SmartPublisherSetFPS</code> | 设置 fps, 如不设置, 用底层默认值 |
| | GOP 间隔(关键帧) | <code>SmartPublisherSetGopInterval</code> | 设置推送端 GOP 间隔, 一般建议在帧率的 1~3 倍, 如不设置, 用底层默认值 |
| | 软、硬编解码率设置 | <code>SmartPublisherSetVideoBitRate</code> | 设置软编码视频 bit-rate, 最大码流一般是平均码流的 2 倍, 如不设置, 用底层计算的默认值 |

| | | | |
|------|------------------|--|--|
| | 设置美颜或外部采集分辨率 | SmartPublisherSetExternalResolution | 美颜或外部视频采集时使用，设置采集分辨率 |
| | 软编码视频 Profile | SmartPublisherSetSWVideoEncoderProfile | 设置软编码模式下的 video encoder profile, 默认 baseline profile |
| | 软编码编码速度 | SmartPublisherSetSWVideoEncoderSpeed | 设置软编码编码速度，设置范围 (1,6)，1 最快，6 最慢，默认是 6 |
| 视频处理 | 视频裁剪 | SmartPublisherSetClippingMode | 设置裁剪模式(仅用于 640*480 分辨率，裁剪主要用于移动端宽高适配)，如不设置，不做裁剪 |
| | 视频镜像 | SmartPublisherSetMirror | 镜像模式：播放端和推送端本地回显方向显示一致 |
| | 是否使用美颜 | SmartPublisherSetBeauty | 0: 不加美颜 1: 内部 daniulive 基础美颜 2: 第三方美颜对接 |
| | 内部美颜亮度调节 | SmartPublisherSetBeautyBrightness | 内部美颜时使用，亮度调节 |
| | 实时快照设置 | SmartPublisherSaveImageFlag | 设置是否启用快照功能 |
| | 实时快照 | SmartPublisherSaveCurImage | 推送或录像过程中，根据设置路径和文件名，实时快照 |
| | 音频处理 | 实时静音 | SmartPublisherSetMute |
| | RTMP 推送模式 | SetRtmpPublishingType | 设置 rtmp publisher 类型，0: live, 1: record, 需服务器支持 |
| | 设置 video preview | SmartPublisherSetVideoPreview | 此接口仅当用 daniulive 采集视频数据时设置，若视频源来自外部美颜或外部第三方数据，无需调用 |
| | 开始采集音视频数据 | SmartPublisherStartCapture | 开始采集音视频数据 |

| | | |
|----------------|----------------------------|--|
| 停止采集音视频数据 | SmartPublisherStopCaputure | 停止采集音视频数据 |
| 切换前后置摄像头 | SmartPublisherSwitchCamera | 此接口仅当用 daniulive 采集视频数据时设置 |
| RTMP 推送 URL 设置 | SmartPublisherSetURL | 设置 RTMP 推送 url |
| RTMP 加密 | SetRtmpEncryptionOption | 设置 rtmp 推送加密选项,可单独加密视频或音频 |
| | SetRtmpEncryptionAlgorithm | 设置 rtmp 加密算法, 当前支持 aes 和国标 sm4. 1 为 aes, 2 为 sm4 |
| | SetRtmpEncryptionKey | 设置 rtmp 推送加密密钥, 支持 AES128、AES192、AES256 和 SM4 国密 |
| | SetRtmpEncryptionIV | 设置 rtmp 推送加密 IV(初始化向量), 这个接口不调用的话, 将使用默认 IV |
| 编码前实时视频数据 | YUV(I420) | SmartPublisherSetExternalYuvData 第三方 YUV (I420) 接口 |
| | BGRA 数据 | SmartPublisherSetExternalBGRAData 第三方 BGRA 数据 |
| | ARGB 数据 | SmartPublisherSetExternalARGBData 第三方 ARGB 数据 |
| | CMSampleBufferRef | SmartPublisherPostVideoSampleBuffer 设置编码前视频 CMSampleBufferRef 数据 |
| 编码前实时音频数据 | CMSampleBufferRef | SmartPublisherPostAudioSampleBuffer 设置编码前音频 CMSampleBufferRef 数据 1: 麦克风 2: 应用程序音频 |
| | 投递外部 PCM 到 SDK | SmartPublisherPostAudioPCMData 投递 PCM 音频数据给 SDK, 每 10ms 音频数据传入一次 |
| 屏幕采集相关 | 视频宽高比例缩放 | SmartPublisherSetVideoSizeScaleRate 设置编码前视频宽高比例缩放(用于屏幕采集缩放), 数据传输对应 |

| | | |
|-----------------|------------------------------|---|
| | | SmartPublisherPostVideoSampleBuffer 接口 |
| SDK 运行模式 | SmartPublisherSetSDKRunMode | 设置 SDK 运行模式 0 : 正常模式 1 : 后台推送屏幕时 |
| 开始 RTMP 推送 | SmartPublisherStartPublisher | 开始 RTMP 推送 |
| 停止 RTMP 推送 | SmartPublisherStopPublisher | 停止 RTMP 推送 |
| 设置推送 RTSP 传输方式 | SetPushRtspTransportProtocol | transport_protocol: 1 表示 UDP 传输 rtp 包; 2 表示 TCP 传输 rtp 包 |
| 设置推送 RTSP 的 URL | SetPushRtspURL | 设置推送 RTSP 的 URL |
| 开始 RTSP 推送 | StartPushRtsp | 启动推送 RTSP 流 |
| 停止 RTSP 推送 | StopPushRtsp | 停止推送 RTSP 流 |
| 关闭推送实例 | SmartPublisherUnInit | 结束时必须调用 UnInit 接口释放资源 |

3.2 iOS 推流端 SDK Event 回调说明

由于 iOS 录像 SDK、内置轻量级 RTSP 服务 SDK 二者和推送端 SDK 可组合使用，相关 Event 同步更新在 iOS 推送端 SDK(如下图):

| iOS 推送端 SDK Event 回调说明 | |
|---|------------------------------|
| 事件 ID | 事件描述 |
| EVENT_DANIULIVE_ERC_PUBLISHER_STARTED | 开始推送 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTING | 连接中 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTION_FAILED | 连接失败 |
| EVENT_DANIULIVE_ERC_PUBLISHER_CONNECTED | 连接成功 |
| EVENT_DANIULIVE_ERC_PUBLISHER_DISCONNECTED | 连接断开 |
| EVENT_DANIULIVE_ERC_PUBLISHER_STOP | 停止推送 |
| EVENT_DANIULIVE_ERC_PUBLISHER_RECORDER_START_NEW_FILE | 开始一个新的录像文件 (param3 返回包含录像 |

| | |
|---|---|
| | 路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PUBLISHER_ONE_RECORDER_FILE_FINISHED | 已生成一个录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PUBLISHER_CAPTURE_IMAGE | 实时快照 |
| EVENT_DANIULIVE_ERC_PUBLISHER_RTSP_URL | 内置生成的 RTSP URL |
| EVENT_DANIULIVE_ERC_PUSH_RTSP_SERVER_RESPONSE_STATUS_CODE | 推送 RTSP 时服务端相应的 status code 上报, 目前只上报 401 |
| EVENT_DANIULIVE_ERC_PUSH_RTSP_SERVER_NOT_SUPPORT | 推送 RTSP 时服务器不支持 RTSP 推送 |

3.3 iOS 推送端录像 SDK 说明

大牛直播推送端录像 SDK，和推送、内置轻量级 RTSP 服务、转发模块完全隔离，可以随意组合使用。

- ✧ 文件格式：MP4；
- ✧ 涉及相关库：libSmartPublisherSDK.a
- ✧ 头文件：SmartPublisherSDK.h nt_event_define.h

以下是接口详解：

| iOS 推送端录像 SDK 接口详解 | | | |
|--------------------|-----------------------------|--------------------------------------|--|
| 调用描述 | 接口 | 接口描述 | |
| 录像设置 | 是否录像 | SmartPublisherSetRecorder | 设置是否启用本地录像 |
| | 设置录像目录 | SmartPublisherSetRecorderDirectory | 设置录像文件目录 |
| | 设置录像文件大小 | SmartPublisherSetRecorderFileMaxSize | 设置每个录像文件的大小，比如 100M，超过这个大小后，会自动生成下一个录像文件 |
| 开始录像 | SmartPublisherStartRecorder | 开始录像 | |
| 停止录像 | SmartPublisherStopRecorder | 停止录像 | |

3.4 iOS 转发 SDK 说明

iOS 转发 SDK，简单来说，播放端 SDK 拉取 RTSP/RTMP 流，并回调编码后的音视频数据到上层，然后，调用我们的推送端 SDK，通过推送端 SDK 扩展数据接口，完成 RTMP 数据转发，整个过程由于不涉及解码、重编码，支持多路转发，超低延迟和低资源占用。

✧ 涉及相关库：libSmartPublisherSDK.a libSmartPlayerSDK.a

✧ 头文件：

SmartPublisherSDK.h SmartPlayerSDK.h nt_event_define.h nt_common_media_define.h

以下是接口详解：

| iOS 转发 SDK 接口详解 | | |
|-----------------|---|--|
| 调用描述 | 接口 | 接口描述 |
| 创建推流实例和播放实例 | 从播放端拉取 RTSP/RTMP 流，并回调到上层，调用推送端数据转发接口，实现转发逻辑 | 注意：推送端调用 SmartPublisherOpen 时： Audio_opt 填 2： video_opt:填 2. |
| 播放端开始拉流 | SmartPlayerStartPullStream | 播放端开始拉流，用于音视频数据转发 |
| 播放端停止拉流 | SmartPlayerStopPullStream | 播放端停止拉流，用于音视频数据转发 |
| 播放端视频回调 | SmartPlayerSetPullStreamVideoDataBlock | 设置拉流时，视频数据回调 |
| | PullStreamVideoDataBlock | 回调视频数据 |
| 播放端音频回调 | SmartPlayerSetPullStreamAudioDataBlock | 设置拉流时，音频数据回调 |
| | PullStreamAudioDataBlock | 回调音频数据 |
| 音频转码 | 设置拉流时音频转 AAC 编码的开关 | 设置拉流时音频转 AAC 编码的开关 (PCMA/PCMU/SPEEX 转 AAC) |
| 视频转发 | SmartPublisherPostVideoEncodedData/SmartPublisherPostVideoEncodedDataV2 | 设置编码后视频数据(H.264) |
| 音频转发 | SmartPublisherPostAudioEncodedData/SmartPublisherPostAudioEncodedDataV2 | 设置音频数据(AAC/PCMA/PCMU/SPEEX) |

3.5 iOS 内置轻量级 RTSP 服务 SDK 说明

为满足内网无纸化/电子教室等内网超低延迟需求，避免让用户配置单独的服务器，大牛直播 SDK 在推送端发布了轻量级 RTSP 服务 SDK。

简单来说，之前推送端 SDK 支持的功能，内置轻量级 RTSP 服务 SDK 后，功能继续支持。

内置轻量级 RTSP 服务后，延迟更低，体验更好。

✧ 涉及相关库：libSmartPublisherSDK.a

✧ 头文件：SmartRTSPSeverSDK.h SmartPublisherSDK.h nt_event_define.h

以下是接口详解：

| iOS 内置轻量级 RTSP 服务 SDK 接口详解 | | |
|---|-----------------------------------|--|
| 调用描述 | 接口 | 接口描述 |
| 常规功能 | iOS RTMP 推送端所有功能依然有效，亦可同时推送 RTMP | |
| <i>SmartRTSPServerSDK</i> | | |
| 创建一个 rtsp server | OpenRtspServer | 创建一个 rtsp server，返回 rtsp server 句柄 |
| 设置端口 | SetRtspServerPort | 设置 rtsp server 监听端口，在 StartRtspServer 之前必须要设置端口 |
| 设置鉴权用户名、密码 | SetRtspServerUserNameAndPassword | 设置 rtsp server 鉴权用户名和密码，这个可以不设置，只有需要鉴权的再设置 |
| 获取 rtsp server 当前会话数 | GetRtspServerClientSessionNumbers | 获取 rtsp server 当前的客户会话数，这个接口必须在 StartRtspServer 之后再调用 |
| 启动 rtsp server | StartRtspServer | 启动 rtsp server |
| 停止 rtsp server | StopRtspServer | 停止 rtsp server |
| 关闭 rtsp server | CloseRtspServer | 关闭 rtsp server |
| UnInit rtsp server | UnInitRtspServer | UnInit rtsp server(和 InitRtspServer 配对使用，即便是启动多个 RTSP 服务，也只需调用一次 UnInitRtspServer) |
| <i>SmartRTSPServerSDK 供 Publisher 调用的接口</i> | | |
| 设置 rtsp 的流名称 | SetRtspStreamName | 设置 rtsp 的流名称 |
| 给要发布的 rtsp 流设置 rtsp server | AddRtspStreamServer | 给要发布的 rtsp 流设置 rtsp server，一个流可以发布到多个 rtsp server 上，rtsp server 的创建启动请参考 OpenRtspServer 和 StartRtspServer 接口 |
| 清除设置的 rtsp server | ClearRtspStreamServer | 清除设置的 rtsp server |
| 启动 rtsp 流 | StartRtspStream | 启动 rtsp 流 |

| | | |
|-----------|----------------|-----------|
| 停止 rtsp 流 | StopRtspStream | 停止 rtsp 流 |
|-----------|----------------|-----------|

3.6 iOS SEI 扩展数据发送/接收 SDK

大牛直播 SDK 支持推送端通过 H.264 SEI 信息扩展，实时传输文本/二进制数据信息，播放端做相应解析和回显。

适用于公告广播、冲顶大会、直播、应急指挥/单兵在线教育等各种场景。

目前使用 sei 机制发送用户自定数据到播放端，这种机制有可能会丢失数据，所以这种方式不保证接收端一定能收到。

优势：能和视频保持同步，虽然有可能丢失，但一般的需求都满足了。

目前提供两种发送方式 第一种发送二进制数据，第二种发送 utf8 字符串。

✧ 涉及相关库：libSmartPublisherSDK.a libSmartPlayerSDK.a

✧ 头文件：

SmartPublisherSDK.h SmartPlayerSDK.h nt_event_define.h nt_common_media_define.h

以下是接口详解：

| iOS SEI 扩展数据发送/接收 SDK 接口详解 | | |
|----------------------------|---|--|
| 调用描述 | 接口 | 接口描述 |
| 发送端(置于推送端 SDK) | SmartPublisherSetPostUserDataQueueMaxSize | 设置发送队列大小，为保证实时性，默认大小为 3，必须设置一个大于 0 的数 |
| | SmartPublisherClearPostUserDataQueue | 清空用户数据队列，有些情况可能会用到，比如发送队列里面有 4 条消息再等待发送，又想把最新的消息快速发出去，可以先清除掉正在排队消息，再调用 PostUserXXX |
| | SmartPublisherPostUserData | 发送二进制数据 NOTE: 1. 目前数据大小限制在 256 个字节以内，太大可能会影响视频传输，如果有特殊需求，需要增大限制，请联系我们 2. 如果积累的数据超过了设置的队列大小，之前的队头数据将被丢弃 3. 必须再调用 StartPublisher 之后再发送数据 |

| | | |
|----------------|--------------------------------------|---|
| | SmartPublisherPostUserUTF8StringData | <p>发送 utf8 字符串</p> <p>NOTE:</p> <ol style="list-style-type: none"> 1. 字符串长度不能超过 256, 太大可能会影响视频传输, 如果有特殊需求, 需要增大限制, 请联系我们 2. 如果积累的数据超过了设置的队列大小, 之前的队头数据将被丢弃 3. 必须再调用 StartPublisher 或 RTSP 流发布之后再发送数据 |
| 接收端(置于播放端 SDK) | SmartPlayerSetUserDataCallback | Set user data Callback. |
| | SP_SDKUserDataCallBack() | 参见 ViewController.m 实现 |
| | SmartPlayerSetSEIDataCallback | Set SEI data Callback. |
| | SP_SDKSEIDataCallBack() | 参见 ViewController.m 实现 |

3.7 iOS 播放端 SDK 说明

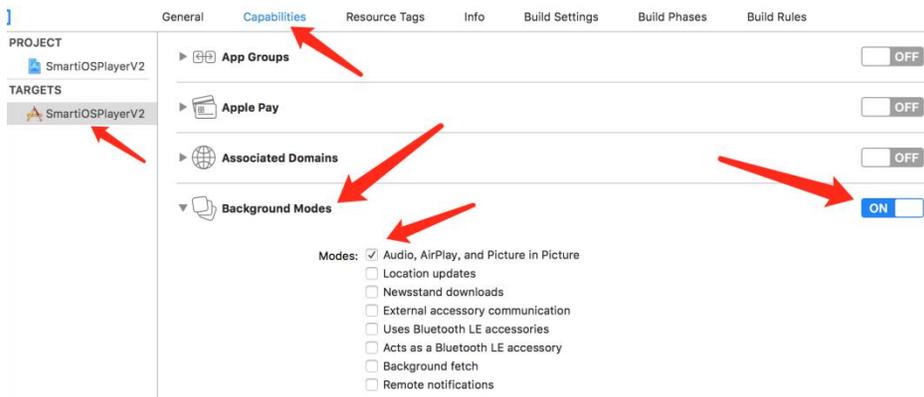
3.7.1 系统要求

- SDK 支持 iOS 7.0 及以上版本(建议在 8.0 及以上版本使用);
- 支持的 CPU 架构: armv7, arm64。

3.7.2 准备工作

- 相关库: libSmartPlayerSDK.a
- 相关头文件:
 - nt_common_media_define.h(如需转发或第三方数据对接)
 - nt_event_define.h
 - SmartPlayerSDK.h
- 如集需要引入的 framework
 - libbz.tbd
 - Libbz2.tbd
 - libiconv.tbd

- libstdc++.tbd
 - Libc++.tbd
 - Accelerate.framework
 - AssetsLibrary.framework
 - AudioToolBox.framework
 - AVFoundation.framework
 - CoreMedia.framework
 - Foundation.framework
 - GLKit.framework
 - OpenGL ES.framework
 - UIKit.framework
 - VideoToolBox.framework
- 如需集成到自己系统测试, 请用大牛直播 SDK 的 app name:
Info.plist-->右键 Open As-->Source Code
添加或者编辑
`<key>CFBundleName</key>`
`<string>SmartiOSPlayer</string>`
- 快照添加到“照片”权限:
Info.plist-->右键 Open As-->Source Code
添加
`<key>NSPhotoLibraryUsageDescription</key>`
`<string>1</string>`
- 如需后台播放音频(添加后台播放权限):



3.7.3 SDK 接口详解

| iOS 播放端 SDK 接口详解 | | |
|---------------------|-----------------------|----------------------------|
| 调用描述 | 接口 | 接口描述 |
| 最先调用, 创建播放实例, 如成功返回 | SmartPlayerInitPlayer | 初始化, 创建 player 实例, 此接口请第一个 |

| | | | |
|-------------------|-----------|--------------------------------|--|
| 回 player 实例 | | | 调用 |
| Event 回调 | | SmartPlayerDelegate | 设置 event callback , 上层由 handleSmartPlayerEvent 处理 |
| 软、硬解码设置 | | SmartPlayerSetVideoDecoderMode | 设置是否用硬解码播放, 如硬解码不支持, 自动适配到软解码 0 : 软解码; 1 : 硬解码. |
| 创建播放 view | | SmartPlayerCreatePlayView | x y width height 指定播放位置 |
| 设置播放 view | | SmartPlayerSetPlayView | 设置播放 view 到底层 SDK |
| 释放播放 view | | SmartPlayerReleasePlayView | 释放播放 view |
| 视频回调 | 设置 YUV 回调 | SmartPlayerSetYuvBlock | 设置拉流时, 视频 YUV 数据回调 |
| | YUV 回调 | PlayerYuvDataBlock | 提供解码后 YUV/RGB 数据接口, 供用户自己 render 或进一步处理(如视频分析) |
| 播放模式 | 缓冲时间设置 | SmartPlayerSetBuffer | 设置播放端缓存数据 buffer, 单位: 毫秒, 如不需 buffer, 设置为 0 |
| | 首屏秒开 | SmartPlayerSetFastStartup | 设置快速启动后, 如果 CDN 缓存 GOP, 实现首屏秒开 |
| | 低延迟模式 | SmartPlayerSetLowLatencyMode | 针对类似于直播娃娃机等期待超低延迟的使用场景, 超低延迟播放模式下, 延迟可达到 200~400ms |
| | 快速切换 URL | SmartPlayerSwitchPlaybackUrl | 快速切换播放 url, 快速切换时, 只换播放 source 部分, 适用于不同数据流之间, 快速切换(如娃娃机双摄像头切换或高低分辨率流切换) |
| RTSP TCP/UDP 模式设置 | | SmartPlayerSetRTSPTcpMode | 设置 RTSP TCP/UDP 模式, 如不设置, 默认 UDP 模式 |
| RTSP 超时时间设置 | | SmartPlayerSetRTSPTimeout | 设置 RTSP 超时时间, timeout 单位为秒, 必 |

| | | | |
|----------------------|------------------------------------|--|--|
| 置 | | 须大于 0 | |
| 设置 RTSP TCP/UDP 自动切换 | SmartPlayerSetRTSPAutoSwitchTcpUdp | 对于 RTSP 来说, 有些可能支持 rtp over udp 方式, 有些可能支持使用 rtp over tcp 方式 为了方便使用, 有些场景下可以开启自动尝试切换开关, 打开后如果 udp 无法播放, sdk 会自动尝试 tcp, 如果 tcp 方式播放不了, sdk 会自动尝试 udp. | |
| 实时静音 | SmartPlayerSetMute | 实时静音 | |
| 视频 镜像 旋转 | 旋转 | SmartPlayerSetRotation | 设置顺时针旋转, 注意除了 0 度之外, 其他角度都会额外消耗性能, 当前支持 0 度, 90 度, 180 度, 270 度 旋转 |
| | 水平反转 | SmartPlayerSetFlipHorizontal | 设置视频水平反转 |
| | 垂直反转 | SmartPlayerSetFlipVertical | 设置视频垂直反转 |
| 设置 URL | SmartPlayerSetPlayURL | 设置播放或录像的 url | |
| RTMP 解密 | SmartPlayerSetKey | 设置解密 key, 目前只用来解密 rtmp 加密流 | |
| | SmartPlayerSetDecryptionIV | 设置解密向量, 目前只用来解密 rtmp 加密流 | |
| 开始播放 | SmartPlayerStart | 开始播放 RTSP/RTMP 流 | |
| 停止播放 | SmartPlayerStop | 停止播放 RTSP/RTMP 流 | |
| 销毁播放实例 | SmartPlayerUnInitPlayer | 结束时必须调用 close 接口释放资源 | |

3.8 iOS 播放端 SDK Event 回调说明

由于 iOS 播放录像 SDK 和播放端 SDK 可组合使用, 相关 Event 同步更新在 iOS 播放端 SDK(如下图):

| iOS 播放端 SDK Event 回调说明 | |
|--|---------|
| 事件 ID | 事件描述 |
| EVENT_DANIULIVE_ERC_PLAYER_STARTED | 开始播放 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTING | 播放端连接中 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTION_FAILED | 播放端连接失败 |
| EVENT_DANIULIVE_ERC_PLAYER_CONNECTED | 播放端连接成功 |

| | |
|---|---|
| EVENT_DANIULIVE_ERC_PLAYER_DISCONNECTED | 播放端连接断开 |
| EVENT_DANIULIVE_ERC_PLAYER_STOP | 停止播放 |
| EVENT_DANIULIVE_ERC_PLAYER_RESOLUTION_INFO | 返回视频宽、高信息 |
| EVENT_DANIULIVE_ERC_PLAYER_NO_MEDIADATA_RECEIVED | 收不到媒体数据(可能是 URL 错误) |
| EVENT_DANIULIVE_ERC_PLAYER_SWITCH_URL | 快速切换 URL |
| EVENT_DANIULIVE_ERC_PLAYER_RECORDER_START_NEW_FILE | 开始一个新的录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PLAYER_ONE_RECORDER_FILE_FINISHED | 已生成一个录像文件 (param3 返回包含录像路径在内的录像文件名) |
| EVENT_DANIULIVE_ERC_PLAYER_CAPTURE_IMAGE | 播放端实时快照 |
| EVENT_DANIULIVE_ERC_PLAYER_START_BUFFERING | 开始缓冲数据 |
| EVENT_DANIULIVE_ERC_PLAYER_BUFFERING | 缓冲中(param1 参数会返回缓冲百分比) |
| EVENT_DANIULIVE_ERC_PLAYER_STOP_BUFFERING | 停止缓冲数据 |
| EVENT_DANIULIVE_ERC_PLAYER_DOWNLOAD_SPEED | 返回当前 RTSP/RTMP 流实时下载速度 |
| EVENT_DANIULIVE_ERC_PLAYER_RTSP_STATUS_CODE | RTSP 收到错误码, 可能是用户名、密码不对 |
| EVENT_DANIULIVE_ERC_PLAYER_NEED_KEY | RTMP 加密流, 请设置播放需要的 Key |
| EVENT_DANIULIVE_ERC_PLAYER_KEY_ERROR | RTMP 加密流, Key 错误, 请重新设置 |

3.9 iOS 播放端录像 SDK 说明

大牛直播播放端录像 SDK，和播放 SDK、转发模块完全隔离，可以随意组合使用，在复杂网络环境适应、时间戳跳变都做了很好的兼容，也可能是市面上最好用的 RTSP/RTMP 流录像模块(可支持多路录像)。

- ✧ 文件格式：MP4；
- ✧ 涉及相关库：libSmartPlayerSDK.a
- ✧ 头文件：SmartPlayerSDK.h nt_event_define.h

以下是接口详解：

| iOS 播放端录像 SDK 接口详解 | | |
|--------------------|---|---|
| 调用描述 | 接口 | 接口描述 |
| 录像 设置 | 设置录像目录 SmartPlayerSetRecorderDirectory | 设置录像文件目录 |
| | 设置录像文件大小 SmartPlayerSetRecorderFileMaxSize | 设置每个录像文件的大小，比如 100M，超过这个大小后，会自动生成下一个录像文件 |
| | 音频转码 SmartPlayerSetRecorderAudioTranscode AAC | 设置录像时音频转 AAC 编码的开关 aac 比较通用，sdk 增加其他音频编码(比如 speex, pcmu, pcma 等)转 aac 的功能. |
| | 录制视频 SmartPlayerSetRecorderVideo | 设置是否录视频，默认的话，如果视频源有视频就录，没有就不录，但有些场景下可能不想录制视频，只想录音频，所以增加个开关 |
| | 录制音频 SmartPlayerSetRecorderAudio | 设置是否录音频，默认的话，如果视频源有音频就录，没有就不录，但有些场景下可能不想录制音频，只想录视频，所以增加个开关 |
| 开始录像 | SmartPlayerStartRecorder | 开始录像 |
| 停止录像 | SmartPlayerStopRecorder | 停止录像 |

3.10 iOS 一对一互动 SDK 说明

大牛直播 SDK iOS 平台一对一互动 SDK，在 iOS RTSP/RTMP 推送甚至内置 RTSP 服务和 iOS RTMP/RTSP 播放基础上，通过添加回音消除接口，实现无需复杂的接口逻辑变化，即可实现非常优异的回音消除效果。

对应 SDK demo: SmartIOSEchoCancellation

✧ 涉及相关库: libSmartPublisherSDK.a libSmartPlayerSDK.a

✧ 头文件:

SmartPublisherSDK.h SmartPlayerSDK.h nt_event_define.h nt_common_media_define.h

以下是接口详解:

| iOS 一对一互动 SDK 接口详解 | | |
|--------------------|------------------------------------|--|
| 调用描述 | 接口 | 接口描述 |
| 推送端 | SmartPublisherSetEchoCancellation | 设置是否启用回音消除 isEnabledEchoCancellation: YES: 打开 NO: 关闭 |
| 播放端 | SmartPlayerSetEchoCancellationMode | 设置音频回音消除模式(如不设置 则用正常播放模式) Mode: 0: 正常播放模式 1: 回音消除模式 |