纯播放端 SDK(Java 版)说明(V1.0.3)

一、基本概念

房间：服务器上一个虚拟的概念，进入同一个房间的客户端才允许数据交互，不同房间之间的客户端并无交集。一个房间内客户端的最大数目理论不限，具体根据服务器端设置。服务器允许最大的房间数目同样依据服务端设置而定。最简单的模型是一个房间内一个发布者客户端，一个接收者客户端。

客户端 UID: 用于标识每个客户端的唯一 ID，在系统中不允许有相同 UID 的客户端同时在线，否则新登录的客户端会将之前的客户端“顶下去”。

音视频发布者: 具备上行音视频到服务器（房间）能力的客户端类型，同时也能接收来自服务器（房间）的音视频数据。

音视频接收者: 只从服务器（房间）接收音视频的客户端类型，不具备上行能力，比如直播业务中的普通观众。

音视频位置: 同一个房间内，允许存在多个音视频发布者发布音视频到不同的“位置”上，接收者可选择接收某个或多个位置的音视频。目前单个房间理论最多允许 32 个“位置”，实际数目由具体项目需求而定。发布者登录后可以选择由服务器自动分配空闲的“位置”供其发布，也可以指定“位置”发布，若当前该“位置”上已有其他发布者，后者将顶替前者以保证同一位置上同时只有一个发布者。

音视频接收掩码: 客户端可以根据业务需要选择接收房间内某个或某几个位置的音频或视频数据，可以通过设置自己期望接收的掩码到服务器。音频和视频使用各自独立的掩码，每个掩码 32bit 对应房间内的 32 个位置。

传输参数: 本文传输参数是指视频通道的 FEC 上行冗余度、上行 FEC Group 组大小、接收 JitterBuff 时间等，音频通道在内部已经根据经验数据配置好了合适的值且不对外开放。对于纯接收者同样也可以设置 FEC 上行冗余度、上行 FEC Group 组大小，只是没有实际意义。

二、SDK 的组成

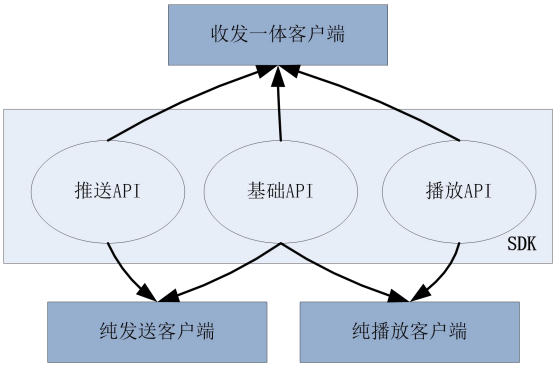
SDK 由如下三类 API 组成：

基础 API 接口，负责基础的 TCP 连接管理（如登录、下线）、音视频码流收发、底层回调反

馈等。

推送 API 接口，负责摄像头、麦克风的采集、滤镜、压缩编码等处理。

播放 API 接口，负责播放远端音视频。



基础 SDK 包括所有类别客户端均需调用的基础功能,纯发送型客户端无需调用播放 API,纯播放型客户端无需调用推送 API。

三、基础 API

以下接口均为线程安全，可以在多线程环境中调用，定义位于 CSDInterface.java 中。

1、系统初始化

系统初始化主要完成日志模块初始化、服务器 IP 地址配置，该 API 在整个系统中只需要调用一次即可。

int SDsysinit(String strServerIp, String strLogFileDir, byte byLogFileLevel)

参数:

@strServerIp，指定服务器 IP 地址

@strLogFileDir，日志文件输出的目录，若目录不存在，SDK 将自动创建，支持相对路径或绝对路径。

@byLogFileLevel，日志输出的级别，只有等于或者高于该级别的日志会输出到文件。级别定义位于 Constant.java:

```
final byte LOG_LEVEL_DEBUG = 1;
final byte LOG_LEVEL_INFO = 2;
final byte LOG_LEVEL_WARN = 3;
final byte LOG_LEVEL_ERROR = 4;
```

final byte LOG_LEVEL_ALARM = 5;

final byte LOG_LEVEL_FATAL = 6;

final byte LOG_LEVEL_NONE = 7;

当指定为 SD_LOG_LEVEL_NONE 时，将不会生成日志文件。

返回值：返回 0 表示初始化成功，返回负数则为失败，负数值为其错误码。

2、系统反初始化

void SDsysexit()

与 SDsysinit 对应，系统退出前调用。

参数：无

返回值：无

3、登录服务器

int SDOnlineUser(int nRoomId, int nUserId, byte byUserType, int nDomainId);

参数：

@nRoomId，表示当前客户端进入的房间号，要求非 0

@nUserId，表示当前客户端使用的唯一用户 ID，需要由用户自行保障唯一性。要求非 0

@byUserType，表示客户端的类型，定义位于 Constant.java：

final byte USER_TYPE_OTHER = 0; //保留未使用的类型

final byte USER_TYPE_AV_SEND_RECV = 1; //同时进行音视频发送和接收

final byte USER_TYPE_AV_RECV_ONLY = 2; //仅接收音视频

final byte USER_TYPE_AV_SEND_ONLY = 3; //仅发送音视频

用户根据自身业务选择合适的客户端类型，以获得资源的最低占用。

@unDomainId，域 ID，默认为 1 即可，具体含义请参考 SRTP-Server 服务器设计相关文档。

返回值：返回 0 表示登录成功，返回负数则为失败，负数值为其错误码。

4、下线服务器

void SDOfflineUser();

参数：无

返回值：无

5、设置音视频下行掩码

void SDSetAvDownMasks(int nAudioDownMask, int nVideoDownMask);

通过设置音视频下行掩码，可以选择从服务器接收哪几个位置的音视频数据。每一个 bit 对应一个位置，最低位对应 0 号位置，最高位对应 31 号位置。比如希望接收某个 index 位置的音视频时，可以设置其对应 bit 设置为 1，否则设置为 0。

允许接收某个 index 位置的音视频时，可以设置为：

```
UINT unMask = 0x1 << (index);
unAudioDownChannelsMask |= unMask;
unVideoDownChannelsMask |= unMask;
```

停止接收某个 index 位置的音视频时，可以设置为：

```
UINT unMask = 0x1 << (index);
unMask = ~unMask;
unAudioDownChannelsMask &= unMask;
unVideoDownChannelsMask &= unMask;
SDSetAvDownMasks(unAudioDownChannelsMask, unVideoDownChannelsMask);
```

当本客户端仅发送音视频，不接收音视频时，设置接收掩码为 0 即可。

参数：

@nAudioDownMask，控制音频接收的掩码。

@nVideoDownMask，控制视频接收的掩码。

返回值：无

说明：请在 Online 成功后调用。

6、设置音视频传输参数

void SDSetTransParams(int nRedunRatio, int nGroupSize, int nEnableNack, int nJitterBufTime);

参数：

@nRedunRatio，固定冗余度时对应的上行冗余比率，比如设置为 30，则表示使用 30%冗余。当设置为 0 时表示使用 AUTO_REDUN 自动冗余度。

@nGroupSize， 为上行 FEC 分组大小，512Kbps 以下建议设置为 8，512Kbps~1Mbps 建议设置为 16，1Mbps~2Mbps 建议设置 24，2Mbps 以上建议设置 28。

@nEnableNack，是否启用 NACK 功能，关于 NACK 请阅读相关文档，建议设置为 1 开启。

@nJitterBufTime，本客户端接收码流时的内部缓存时间（毫秒）,范围 0~600。设置为 0 时，将关闭内部接收 JitterBuff 功能。对延时无过高要求的场景下建议设置 200ms+。

返回值：无

说明：本函数需在 SDOnlineUser 之前调用，本 API 的使用若有疑问，请联系技术支持获得帮助。

四、播放 API

播放 API 定义位于 SDInterfacePlayer.java 中。一个 APP 中可以创建多个播放实例实现多路播放功能。

1、播放初始化

void Init(Activity act, SurfaceViewRenderer surfaceView, boolean playAudioOnly, boolean playVideoOnly, int decodeMode)

参数：

@ act，播放实例所在的 Activity。

@ surfaceView，画面渲染的 surface view。

@ playAudioOnly，是否仅播放音频。仅播放音频时 surfaceView 可设置为 null

@ playVideoOnly，是否仅播放视频。

@ decodeMode，解码模式， 0(软解码) 1(硬解码) 2(硬解码优先)

返回值：无

说明：需要在调用基础 API SDsysinit 之后调用本 API。

2、播放反初始化

void Destroy()

参数：无

返回值：无

3、开始播放

void startPlay(int index, int renderWidth, int renderHeight, boolean renderKeepRatio)

参数：

@ index，播放的音视频“位置”索引。

@ renderWidth，画面渲染窗口的宽度。

@ renderHeight，画面渲染窗口的高度。

@ renderKeepRatio，为 true 时将保持编码分辨率宽高比输出，画面不变形。为 false 时铺满渲染窗口输出，画面可能变形。

返回值：无

说明：无需提供精确的 renderWidth 和 renderHeight，只需要二者比率与 surfaceView 宽高比一致即可。当然若能获得精确宽高最佳。

4、停止播放

void stopPlay()

参数：无

返回值：无

五、反馈回调

底层 SDK 对外提供了状态反馈，外层通过 Handler 对回调进行处理。绝大部分的回调仅做信息通知使用，外层可仅做信息展示用途。具体描述如下：

1、SYS_NOTIFY_RECON_START

因为网络原因，SDK 与服务器断开连接，并自动开始重连。外层无需处理本消息，亦可界面提示用户。

2、SYS_NOTIFY_RECON_SUCCESS

因为网络原因，SDK 与服务器断开连接，并自动重连成功。外层无需处理本消息，亦可界面提示用户。

3、SYS_NOTIFY_ONPOSITION

提示某个用户开始进入到某个“位置”发布音视频流。外层无需处理本消息，亦可界面提示用户。

4、SYS_NOTIFY_OFFPOSITION

提示某个用户离开其发布“位置”。外层无需处理本消息，亦可界面提示用户。

5、SYS_NOTIFY_EXIT_KICKED

提示当前用户的账号在其他位置登录，本用户被顶下去，外层需要结束播放并提示用户。用户应该避免使用重复 UID 的情况。