# NVR端SDK API接口

## 概述

头文件stream.hpp，class Stream 中的音视频传输协议接口(NVR使用)，用于接收IPC发送的音视频数据、对讲和接收控制数据。

## 流程



## API

* 1. int Start(const std::string &priUrl)
* 功能
  + 开始连接，并设置连接的IP地址
* 参数
  + priUrl IPC的IP地址
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. Int Stop()
* 功能
  + 停止连接，停止连接将停止所有功能，之后可以重新使用Star()来开始连接
* 参数
  + 无
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. bool Status()
* 功能
  + 查询连接状态
* 参数
  + 无
* 返回值
  + true 表示与IPC处于连接中的状态
  + flase 表示与IPC处于未连接状态，可能是未连接，连接中，连接失败或已掉线
  1. int PlayTalk()
* 功能
  + 发送对讲音频给IPC
* 参数
  + 无
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. Int StopTalk()
* 功能
  + 停止发送对讲音频给IPC
* 参数
  + 无
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. Int SendCmd(const std::string& \_cmd)
* 功能
  + 发送控制命令给IPC，具体内容见协议文档
* 参数
  + \_cmd 命令字符串
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. void SetUserdata(void \*\_ptr)
* 功能
  + 设置一个用户数据的指针，之后所有的回调函数都将获得这个指针
* 参数
  + \_ptr 指针
* 返回值
  + 无
  1. void SetFrameCb(std::function<void(sFrame\_t &st, void\* userdata)> \_cb)
* 功能
  + 设置音视频帧数据的回调函数，当协议接收到完整的一帧音频或视频数据时，将调用该回调函数
* 参数
  + \_cb 需要设置的回调函数
* 返回值
  + 无
* 说明
  + 该回调函数不得执行耗时操作，否则会影响数据收发效率，耗时的解码等操作应在其他线程中进行
  1. int AddAacFrame(sFrame\_t \_audio\_frame)
* 功能
  + 添加Aac数据接口，对讲开启时会发送这个函数给出的Aac数据
* 参数
  + \_audio\_frame 音频帧数据，具体格式见结构体
* 返回值
  + 0 表示成功
  + 非0 表示失败，返回值为错误码
  1. void SetCtrlCb(std::function<void(std::string \_cmd, void\* userdata)> \_cb)
* 功能
  + 设置控制信息处理的回调函数，IPC发送给NVR控制命令将会通过该回调函数执行
* 参数
  + \_cb 回调函数
* 返回值
  + 无
* 说明
  + 该回调函数不得执行耗时操作，否则会影响数据收发效率，耗时的解码等操作应在其他线程中进行

## 枚举和结构体

* 1. 枚举 Stream::CODEC\_TYPE

|  |  |
| --- | --- |
| VIDEO\_CODEC\_H264 | 视频H264编码 |
| VIDEO\_CODEC\_H265 | 视频H265编码 |
| VIDEO\_CODEC\_AAC | 音频AAC编码 |

* 1. 枚举 Stream::CHANNEL\_TYPE

|  |  |
| --- | --- |
| CHANNEL\_AUDIO | 音频类型 |
| CHANNEL\_VIDEO | 视频类型 |
|  |  |

* 1. 结构体Stream::sFrame\_t

|  |  |
| --- | --- |
| channel\_type | 音视频类型 |
| codec\_type | 编码类型 |
| timestamp | 时间戳 |
| data | 数据 |