GB28181 设备端补充接口（android）版

GB28181-2016 属于安防监控的传输协议标准，仅明确了设备端与服务端（平台）的交互协议标准，且所有控制均由平台端发起。无法满足在单兵，执法仪，安全帽等智能终端在强终端应用下，特别是终端与终端之间互联，交互的场景。因此基于原 SDK 扩展如下功能接口。

1. 视频与语音的对讲

本接口以视跃音视频融合平台 V3.0 版（及以上版本）为中枢，视频的交互与语音的对讲以发布（推流）与播放（拉流）接口予以体现。一方终端或者平台 web 用户调用接口将自己的视频（音频）信息发布（推送）到平台，另一方终端或者平台 web 用户调用接口播放（拉流）该设备（web 用户）的流，即为一次单向的音视频流交互。同理，双向音视频对讲则各自发布（推送）自己本地的流，然后播放（拉流）对方的流即可。多方视频会议或集群对讲亦是如此，多方推送，多方播放。

1.1 主要接口说明

音视频的发布与播放通过类 SYRTCMediaAdapter 予以实现，通过该类，可以方便的完成音视频交互的所有功能。使用前引入该类与相关类与相关类如下:

```
import com.shiyue.syrtc.SYRTCMediaSink;
import com.shiyue.syrtc.SYRTCMediaAdapter;
```

1.1.1 对象的创建

原型：`public SYRTCMediaAdapter(String domain, SYRTCMediaSink sink, String ice, boolean isSingleAudio)`

参数：

类型	名称	描述
String	Domain(in)	服务器 IP 或域名
SYRTCMediaSink	sink(in)	远端视频回调对象，详见注 1
String	ice(in)	Ice 服务，设备 p2p 穿透用，该场景可填 null
boolean	isSingleAudio(in)	是否是纯音频，纯音频对讲为 true，音视频对讲为 false

调用示例：

```
SYRTCMediaAdapter symediaadapter = new SYRTCMediaAdapter("gbs.realgbs.com", this, null,true);
```

注 1：SYRTCMediaSink 为播放（拉流）时远端视频 track 的回调基接口，其原如下：

```
public interface SYRTCMediaSink {
    void onRemoteVideoTrack(VideoTrack var1);
}
```

当对象继承 SYRTCMediaSink 时，把该对象 this 作为参数传入，当调用播放接口时，远端流会通过 onRemoteVideoTrack 函数将远端 VideoTrack 回调。一般此时将该远端 VideoTrack 加入 **SurfaceViewRenderer** 控件即可显示远端视频流。示例代码如下：

```
@Override
public void onRemoteVideoTrack(VideoTrack videotrack)
{
    runOnUiThread(() -> {
        videotrack.addSink(remoteView);
    });
}
```

1.1.2 对象的初始化

原型： **public** VideoTrack init(Context eglBaseContext, android.content.Context context)

参数：

类型	名称	描述
Context	eglBaseContext (in)	**Egl** 的 Context
android.content.Context	context (in)	当前的 Context
VideoTrack	Return	本地的 videotrack，可以绑定 **SurfaceViewRenderer** 预览，纯音频场景返回为 null

例子：

```
eglBaseContext = EglBase.create().getEglBaseContext();
VideoTrack videoTrack = symediaadapter.init(eglBaseContext, this);
```

1.1.3 设置服务器 IP（域名）

原型：**public void** setdomain(String domain)

参数：

类型	名称	描述
String	domain (in)	服务器 IP 或域名

1.1.4 设置本地视频的分辨率与帧率

原型：**public void** setvideoparam (**int** w, **int** h, **int** f)
参数：

类型	名称	描述
int	w(in)	视频的宽
int	h(in)	视频的高
int	f(in)	视频的帧率

注意：如果调用该接口，需在初始化之前。不调用该接口为默认参数。默认参数为：分辨率 480*640，fps 为 30

1.1.5 设置本地视频采集的旋转角度

原型：**public void** setFrameRotation(**int** rotation)
参数：

类型	名称	描述
int	rotation (in)	旋转角度

注意：如果调用该接口，需在初始化之前。

1.1.6 发布（推流）音视频流

原型：**public void** startpush(String id,SYRTCCallback cb)
参数：

类型	名称	描述
String	id(in)	发布的流 id，需保持唯一性，建议为国标 id
SYRTCCallback	Cb	发布中的状态回调，一般可以为 null

1.1.7 发布（推流）音视频流（群组语音视频模式）

原型：**public void** startpush(String id,SYRTCCallback cb, String groupinfo)
参数：

类型	名称	描述
String	id(in)	发布的流 id，需保持唯一性，建议为国标 id
SYRTCCallback	Cb	发布中的状态回调，一般可以为 null

String	groupinfo	群组 id 列表，id 以”；”隔开，例如“42000000013200000001;admin”

通过该接口发起视频或者语音多方会议（会谈）

1.1.8 播放（拉取）音视频流

原型：**public void** startplay(String id ,SYRTCCallback cb)

参数：

类型	名称	描述
String	id(in)	拉取的流 id
SYRTCCallback	Cb	发布中的状态回调，一般可以为 null

注意：在融合通信里也可以拉取国标接入的流，拉取国标流时 id 的格式为：”/0/国标 id”

1.1.9 停止播放（发布）音指定视频流

原型：**public void** CloseConn(String id)

参数：

类型	名称	描述
String	id(in)	流 id

注意：调用该接口将停止播放（发布）一路流的会话。

1.1.10 停止播放（发布）所有音视频流

原型：**public void** CloseAllConns ()

参数：无

注意：调用该接口将停止播放（发布）所有的流。

1.1.11 销毁音视频交互环境

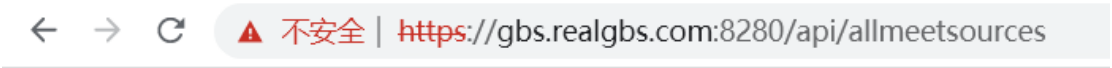
原型：**public void** destroy()

参数：无

注意：一般退出时调用该接口，调用该接口后，如果后面想进行发布或播放流，必须重新调用初始化 init 接口。

1.1.12 获取平台上所有音视频流

该功能需使用 http 或者 https 像平台发送 get 请求。请求 url：https://gbs.realgbs.com:8280/api/allmeetsources，**请求结果以 json 格式返回如下：name 即为流 id**




```
{
        "items" :
        [
                {
                        "ip" : "",
                        "name" : "admin"
                }
        ],
        "limit" : 10,
        "offset" : 0,
        "total" : 1
}
```


示例代码如下：

```
boolean requesthttps(String requestURL)
{
    requestURL = "https://gbs.realgbs.com:8280/api/allmeetsources";
    boolean ret = false;
    try
    {
        URL url = new URL(requestURL);
        HttpURLConnection connection =
(HttpURLConnection)url.openConnection();

        connection.setConnectTimeout(30 * 1000); //30 秒连接超时
        connection.setReadTimeout(30 * 1000);   //30 秒读取超时
        connection.setUseCaches(false);  //不允许使用缓存
        connection.setRequestMethod("GET");
        connection.connect();
        int code = connection.getResponseCode();
        if(connection.getResponseCode() == HttpsURLConnection.HTTP_OK)
{
            InputStreamReader in = new
InputStreamReader(connection.getInputStream());
            BufferedReader bf = new BufferedReader(in);
            String recieveData = null;
            String result = "";
            while ((recieveData = bf.readLine()) != null){
```

```
                result += recieveData;
            }
            in.close();
            connection.disconnect();

            JSONObject payload = new JSONObject(result);
            int total = payload.getInt("total");
            JSONArray ja = payload.getJSONArray("items");
            int length = ja.length();
            if(length > 0)
            {
                audiodevs = new String[length];
                mAdapter.devstatus = new boolean[length];
            }
            for(int i = 0; i< length;i++)
            {
                JSONObject  dt  =  ja.getJSONObject(i);
                String ip  = dt.getString("ip");
                audiodevs[i] = dt.getString("name");
            }

        }
         ret = true;
    } catch (MalformedURLException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
    catch (Exception e) {
        e.printStackTrace();
    }
    return ret;
}
```

1.2 多方音频对讲视频会议的信令交互

Sdk 为满足用户的集群对讲与多方视频会议要求，内部通过国标与 http 实现了信令的交互。集群对讲与视频会议统称为多方多媒体交互。

1.2.1 平台多方多媒体交互的发起

当平台发起多方交互时，web 平台推流时会携带群组 id 列表的信息（这里的 id 为终端的国标 id 或平台的 web 用户名）。

1.2.2app 收到呼叫

继承接口 SYGbtSink，可以收到后台发起的消息回调，包括注册（连接）状态和被动呼叫回调。本章详细说明被呼回调，接口原型如下：

```
public interface  SYGbtSink
{
    /*
        连接状态回调
    */
    void  ConnectStatus(int status, long statuscode, String info);

    /*
        音视频呼叫回调
    */
    void  callin(int status,int type, String remoteid, String
groupinfo);
}
```

继承 SYGbtSink，实现 callin 函数，当收到来自后台的其他设备或 web 用户呼叫时，callin 函数响应，参数意义如下：

类型	名称	描述
int	status (in)	为呼叫等于 201
int	Type(in)	0/1/2/3-停止呼叫/音频呼叫/视频呼叫/音视频呼叫
String	remoteid	远端呼叫方的 id
String	groupinfo	群组 id 信息，直接作为参数传入 startpush

多方交互的逻辑是：收到该被呼信息后，调用 startplay 拉取该远端 id 的流，同时调用 startpush 发布本设备 id 的流（带 groupinfo 参数）。即可由后台完成调度实现多方视频会议。

注意：多方视频会议时会收到所有设备的 callin，每次都到都 staryplay 对方的流，但是 startpush 一次，需要判断如果前面已经 startpush 了，后面不用再调用了。

1.3 多方多媒体交互扩展协议

DeviceControl 信令扩展

< PushStream >id1;id2;id3</ PushStream > 国标 id 或 web 用户 id 列表

< mediatype >3<mediatype> //0:断开呼叫/1:音频呼叫/2:视频呼叫/3 :音视频呼叫