

SDK 参考

SDK 参考

文档版本

01

发布日期

2020-02-08



华为技术有限公司



版权所有 © 华为技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目 录

1 SDK 介绍.....	1
2 SDK 下载.....	2
3 Android SDK.....	3
3.1 开发前准备.....	3
3.2 集成步骤.....	5
3.3 可选功能.....	6
3.4 接口参考.....	6
4 iOS SDK.....	10
5 Web SDK.....	11

1 SDK 介绍

2 SDK 下载

3 Android SDK

[3.1 开发前准备](#)

[3.2 集成步骤](#)

[3.3 可选功能](#)

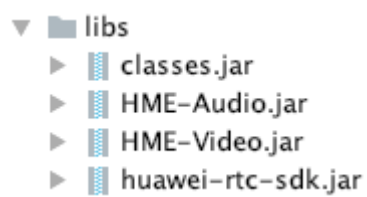
[3.4 接口参考](#)

3.1 开发前准备

SDK 集成

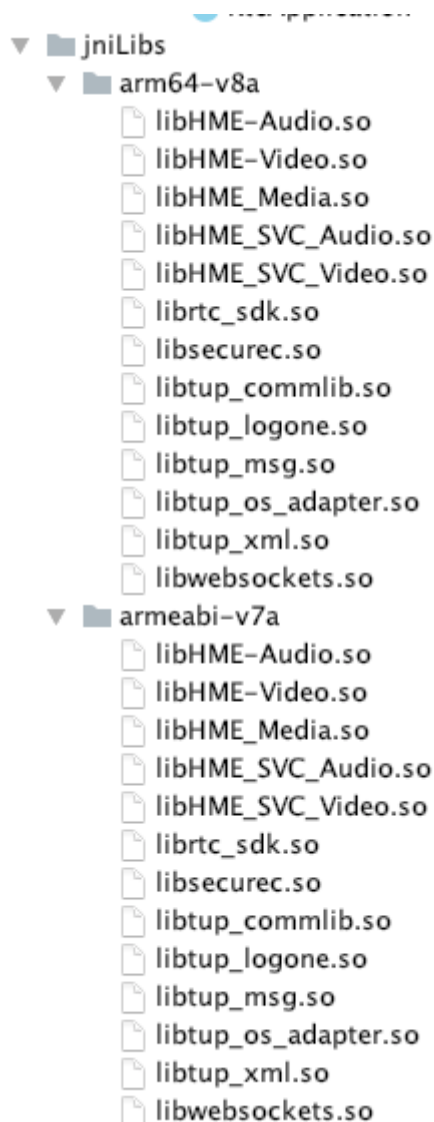
步骤1 添加libs文件包。

图 3-1



步骤2 添加jniLibs库文件

图 3-2



步骤3 在build.gradle文件中设置so库的存放路径。

```
sourceSets {
    main {
        jniLibs.srcDirs = ['src/main/jniLibs']
    }
}
```

步骤4 添加aar依赖。

```
implementation (name: 'hwinteractivesdk-release', ext: 'aar')
```

步骤5 在app/src/main/res/values/strings.xml文件中配置App ID。

```
<string name="setting_default_server" translatable="false">139.9.130.54</string>
<string name="setting_appld_title" translatable="false">Appld</string>
```

App ID：如何获取。

步骤6 在AndroidManifest.xml文件中配置App权限。

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

```
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.BLUETOOTH" />
```

步骤7 同步项目文件。

单击Sync Project With Gradle Files，完成SDK集成。

----结束

3.2 集成步骤

步骤1、创建引擎

```
try {
    mHwRtcEngine = RtcEngine.create(getApplicationContext(), serverAddr, appId, this);
} catch (Exception e) {
    e.printStackTrace();
}
```

步骤2、设置本地窗口

```
SurfaceView surface = RtcEngine.createRenderer(getApplicationContext());
mHwRtcEngine.setupLocalVideo(surface, RtcEnums.ViewMode.VIEW_MODE_CROP);
```

步骤3、加入房间

```
UserInfo userInfo = new UserInfo();
userInfo.setUserId(mUserId);
userInfo.setUserName(mUserId);
userInfo.setRole(UserInfo.RoleType.values()[mRole]);
mHwRtcEngine.joinRoom(userInfo, mChannelId, RtcEnums.MediaType.MEDIA_TYPE_AUDIO_VIDEO);
```

步骤4、监听远端用户加入房间，并设置远端窗口

```
@Override
public void onUserJoined(String roomId, final String userId, String nickname) {
    if (userId.equals(mUserId)||mRole == ROLE_TYPE_PUBLISHER.ordinal()) {
        return;
    }
    Log.e(TAG, "onUserJoined userId: " + userId);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Toast.makeText(LiveActivity.this, userId + "加入了房间", Toast.LENGTH_SHORT).show();
            SurfaceView surface = RtcEngine.createRenderer(getApplicationContext());
            mHwRtcEngine.setupRemoteVideo(surface, RtcEnums.ViewMode.VIEW_MODE_CROP,
            RtcEnums.StreamType.STREAM_TYPE_SD, userId);
            mVideoGridContainer.addUserVideoSurface(userId, surface);
        }
    });
}
```

步骤5、监听远端用户离开房间，并删除远端窗口

```
@Override
public void onUserOffline(String roomId, final String userId, int reason) {
    Log.i(TAG, "HwRtcDemo onUserJoined roomId:" + roomId + ", userId:" + userId + ", reason:" + reason);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            Log.e(TAG, "HwRtcDemo run removeRemoteUser! ");
            Toast.makeText(LiveActivity.this, userId + "离开了房间", Toast.LENGTH_SHORT).show();
            mHwRtcEngine.setupRemoteVideo(null, RtcEnums.ViewMode.VIEW_MODE_CROP,
            RtcEnums.StreamType.STREAM_TYPE_HD, userId);
            mVideoGridContainer.removeUserVideo(userId, false);
        }
    });
}
```



```
    }  
    });  
}
```

步骤6、离开房间

```
mHwRtcEngine.leaveRoom();  
finish();
```

3.3 可选功能

3.4 接口参考

RtcEngine

表 3-1 方法说明

方法	描述
public static synchronized RtcEngine create(Context context, String domain, String appld, IRtcEngineEventHandler eventHandler)	<p>【功能说明】 创建连麦引擎实例。</p> <p>【请求参数】</p> <ul style="list-style-type: none">context：上下文domain：域名appld：应用IDeventHandler：引擎事件句柄，用来做加入房间、离开房间等事件回调。 <p>【返回参数】 返回引擎实例对象。</p> <p>【注意事项】 无</p>
public static synchronized void destroy()	<p>【功能说明】 销毁连麦引擎。</p> <p>【请求参数】 无。</p> <p>【返回参数】 无。</p> <p>【注意事项】 无</p>

方法	描述
<code>public abstract int joinRoom(UserInfo userInfo, String roomId, MediaType mediaType);</code>	<p>【功能说明】 加入房间。</p> <p>【请求参数】</p> <ul style="list-style-type: none">• userInfo: 用户信息,• roomId: 房间ID, 房间唯一标识。• mediaType: 媒体类型。 MEDIA_TYPE_AUDIO: 音频 MEDIA_TYPE_AUDIO_VIDEO: 音频+视频 MEDIA_TYPE_AUDIO_VIDEO_DATA: 保留 <p>【返回参数】 返回0: 成功, 1: 失败, 2: 上下文为空。</p> <p>【注意事项】 无</p>
<code>public abstract int leaveRoom();</code>	<p>【功能说明】 离开房间。</p> <p>【请求参数】 无。</p> <p>【返回参数】 返回0: 成功, 1失败。</p> <p>【注意事项】 无</p>
<code>public static SurfaceView createRenderer(Context context)</code>	<p>【功能说明】 创建渲染视图。</p> <p>【请求参数】</p> <ul style="list-style-type: none">• context: 上下文。 <p>【返回参数】 创建的渲染视图view。</p> <p>【注意事项】 无</p>

方法	描述
<pre>public abstract int setupLocalVideo(SurfaceView view, ViewMode viewMode);</pre>	<p>【功能说明】 设置本地窗口。</p> <p>【请求参数】</p> <ul style="list-style-type: none"> view: 远端窗口视图 viewMode: 显示模式， VIEW_MODE_ZOOM: 拉伸模式，不保持宽高比，默认为此方式 VIEW_MODE_PAD: 通过扩边的方式保持宽高比 VIEW_MODE_CROP: 通过裁剪的方式保持宽高比 VIEW_MODE_ORIG: 通过显示区域的移动进行显示，显示模块有效，编码前处理该参数无效，只支持PC和MAC平台 <p>【返回参数】 返回0: 成功，1失败。</p> <p>【注意事项】 无</p>
<pre>public abstract int setupRemoteVideo(SurfaceView view, ViewMode viewMode, StreamType streamType, String userId);</pre>	<p>【功能说明】 设置远端窗口。</p> <p>【请求参数】</p> <ul style="list-style-type: none"> view: 远端窗口视图 viewMode: 显示模式， VIEW_MODE_ZOOM: 拉伸模式，不保持宽高比，默认为此方式 VIEW_MODE_PAD: 通过扩边的方式保持宽高比 VIEW_MODE_CROP: 通过裁剪的方式保持宽高比 VIEW_MODE_ORIG: 通过显示区域的移动进行显示，显示模块有效，编码前处理该参数无效，只支持PC和MAC平台 streamType: 流类型， STREAM_TYPE_SD: 标清 STREAM_TYPE_HD: 高清 userId: 用户ID，用户唯一标识。 <p>【返回参数】 返回0: 成功，1失败。</p> <p>【注意事项】 无</p>

方法	描述
<code>public abstract int switchCamera();</code>	<p>【功能说明】 切换摄像头。</p> <p>【请求参数】 无。</p> <p>【返回参数】 返回0：成功，非0失败。</p> <p>【注意事项】 无</p>
<code>public abstract int muteLocalAudio(boolean mute);</code>	<p>【功能说明】 设置本地麦克风静音。</p> <p>【请求参数】</p> <ul style="list-style-type: none">• mute, true：本地麦克风静音 false：解除本地麦克风静音 <p>【返回参数】 返回0：成功，非0失败。</p> <p>【注意事项】 无</p>
<code>public abstract int setSpeakerModel(SpeakerModel speakerModel);</code>	<p>【功能说明】 设置声音播放模式。</p> <p>【请求参数】</p> <ul style="list-style-type: none">• SpeakerModel：声音播放模式， AUDIO_EARPIECE：听筒模式 AUDIO_SPEAKER：扬声器模式 <p>【返回参数】 返回0：成功，非0失败。</p> <p>【注意事项】 无</p>

4 iOS SDK

5 Web SDK
