

天舟 VoIP SDK 集成文档 V1.0.5 - 2022.11.23

1. Android SDK集成

1.1 集成准备

获取 SDK

获取 server, username, password

```
String server = "请设置 server地址 ip:port";// 服务器地址
String username = "请设置 sip 账号";
String password = "请填写sip 密码";
```

1.2 支持版本

SDK支持JDK1.6和Android SDK version 14以上的系统,支持'armeabi','armeabi-v7a','arm64-v8a','x86','x86_64'

1.3 具体集成方案

创建一个新工程，并在项目的build.gradle中添加以下内容:

```
repositories{
    flatDir {
        dirs 'libs'
    }
}

defaultConfig {
    ndk {
        //选择要添加的对应cpu类型的.so库
        abiFilters 'armeabi','armeabi-v7a','arm64-v8a','x86','x86_64'
    }
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')

    implementation(name: 'SH_VOIP_SDK', ext: 'aar')

    implementation 'com.squareup.okhttp3:okhttp:3.10.0'
}
```

2. SDK 基本概念及集成

2.1 接口调用顺序

- 初始化，设置状态监听
- 登录，并启动
- 呼出、等待呼入

2.2 初始化，设置状态监听

2.2.1 第一步，初始化

```
/**
 * 第一步，初始化
 *
 * 设置 log 等级， 默认 5
 *
 * Level 0 Display fatal error only.
 * Level 1 Display error messages and more severe verbosity level only.
 * Level 2 Display Warning messages and more severe verbosity level only.
 * Level 3 Info verbosity (normally used by applications).
 * Level 4 Important VoIP events.
 * Level 5 Detailed VoIP events.
 * Level 6 Very detailed VoIP events.
 * /
SHVoIPSDK.getInstance().init(@NonNull Context context);
SHVoIPSDK.getInstance().setLogLevel(5);
```

2.2.2 第二步，设置监听

```
/**
 * 第二步，设置监听
 * 确保 SHVoIPListener 的实现类 不被销毁，否则将导致无法收到回调
 * 推荐在 Application 中实现 SHVoIPListener
 */
SHVoIPSDK.getInstance().setSHVoIPListener(@NonNull SHVoIPListener listener);
```

2.2.3 第三步，登录，启动VoIP

设置接口访问地址帐号信息，server，username，password

```
// 设置接口访问地址帐号信息，server，username，password
shVoIPSDK.setAccount(username, password, server, "", false);
```

2.2.4. 状态监听说明

```
public interface SHVoIPListener {

    /**
```

```

    * 注册成功
    */
void onRegisterSuccess(SHVoIPAccount account);

/**
    * 注册失败
    * @param sipCode 状态码
    */
void onRegisterFail(SHVoIPAccount account, SipCode sipCode);

/**
    * 来电，需在此处播放来电铃声
    * 可使用，SHAudioManager.getInstance().startRingtone();
    * @param number 来电号码
    */
void onIncomingCall(String number);

/**
    * 对方振铃
    * @param number 号码
    * @param sipCode
    */
void onCallRing(String number, SipCode sipCode);

/**
    * 已接通，需在此处停止来电铃声
    * @param number
    * @param sipCode
    */
void onCallOK(String number, SipCode sipCode);

/**
    * 已挂断，需在此处停止来电铃声
    * @param number
    * @param sipCode
    */
void onCallEnd(String number, SipCode sipCode);
}

```

2.3 VoIP 注册信令相关

2.3.1 开始注册

发送register 信令

```
SHVoIPSDK.getInstance().register();
```

2.3.2 停止注册

停止发送发送register 信令

```
SHVoIPSDK.getInstance().unregister();
```

2.3.3 注册状态

```
// 注册成功回调
SHVoIPListener.onRegisterSuccess(SHVoIPAccount account);

// 注册失败回调
SHVoIPListener.onRegisterFail(SHVoIPAccount account, SipCode sipCode)
```

2.4 VoIP 呼叫相关

2.4.1 呼出

需要检查麦克风录音权限

```
/**
 * 需要检查麦克风录音权限
 * @param number
 *         实号
 *         手机号、电话号(区号+号码)
 *         会议号
 * @return
 */
SHVoIPSDK.getInstance().callOut(String number);
```

2.4.2 接听

需要检查麦克风录音权限

```
/**
 * 需要检查麦克风录音权限
 */
SHVoIPSDK.getInstance().answerCall();
```

2.4.3 挂断

```
SHVoIPSDK.getInstance().hangup();
```

2.4.4 呼叫保持

```
SHVoIPSDK.getInstance().setHoldOn(boolean hold);
```

2.4.5 呼叫状态

```
/**
 * 来电，需在此处播放来电铃声
 * 可使用，SHAudioManager.getInstance().startRingtone();
 * @param number 来电号码
 */
void onIncomingCall(String number);

/**
 * 对方振铃
 * @param number 号码
 * @param sipCode
 */
void onCallRing(String number, SipCode sipCode);

/**
 * 已接通，需在此处停止来电铃声
 * @param number
 * @param sipCode
 */
void onCallOK(String number, SipCode sipCode);

/**
 * 已挂断，需在此处停止来电铃声
 * @param number
 * @param sipCode
 */
void onCallEnd(String number, SipCode sipCode);
```

2.4.6 静音、免提

```
// 静音
SHVoIPSDK.getInstance().setMute(boolean on);
// 免提
SHVoIPSDK.getInstance().setSpeaker(boolean on);
```

2.4.7 通话中按键交互(DTMF)

```
SHVoIPSDK.getInstance().sendDTMF(String str);
```

附录

SIP 状态码速查

状态码	描述
1xx	通知性应答
100	正在尝试
180	正在拨打
181	正被转接
182	正在排队
183	通话进展
2xx	成功应答
200	OK
202	被接受：用于转介
3xx	转接应答
300	多项选择
301	被永久迁移
302	被暂时迁移
305	使用代理服务器
380	替代服务
4xx	呼叫失败
400	呼叫不当
401	未经授权：只供注册机构使用，代理服务器应使用代理服务器授权407
402	要求付费（预订为将来使用）
403	被禁止的
404	未发现：未发现用户
405	不允许的方法

406	不可接受
407	需要代理服务器授权
408	呼叫超时：在预定时间内无法找到用户
410	已消失：用户曾经存在，但已从此处消失
413	呼叫实体过大
414	呼叫URI过长
415	不支持的媒体类型
416	不支持的URI方案
420	不当扩展：使用了不当SIP协议扩展，服务器无法理解该扩展
421	需要扩展
423	时间间隔过短
480	暂时不可使用
481	通话/事务不存在
482	检测到循环
483	跳数过多
484	地址不全
485	模糊不清
486	此处太忙
487	呼叫被终止
488	此处不可接受
491	呼叫待批
493	无法解读：无法解读 S/MIME文体部分
5xx	服务器失败
5xx	服务器失败
501	无法实施：SIP呼叫方法在此处无法实施
502	不当网关
503	服务不可使用
504	服务器超时
505	不支持该版本：服务器不支持SIP协议的这个版本

513	消息过长
6xx	全局失败
600	各处均忙
603	拒绝
604	无处存在
606	不可使用