

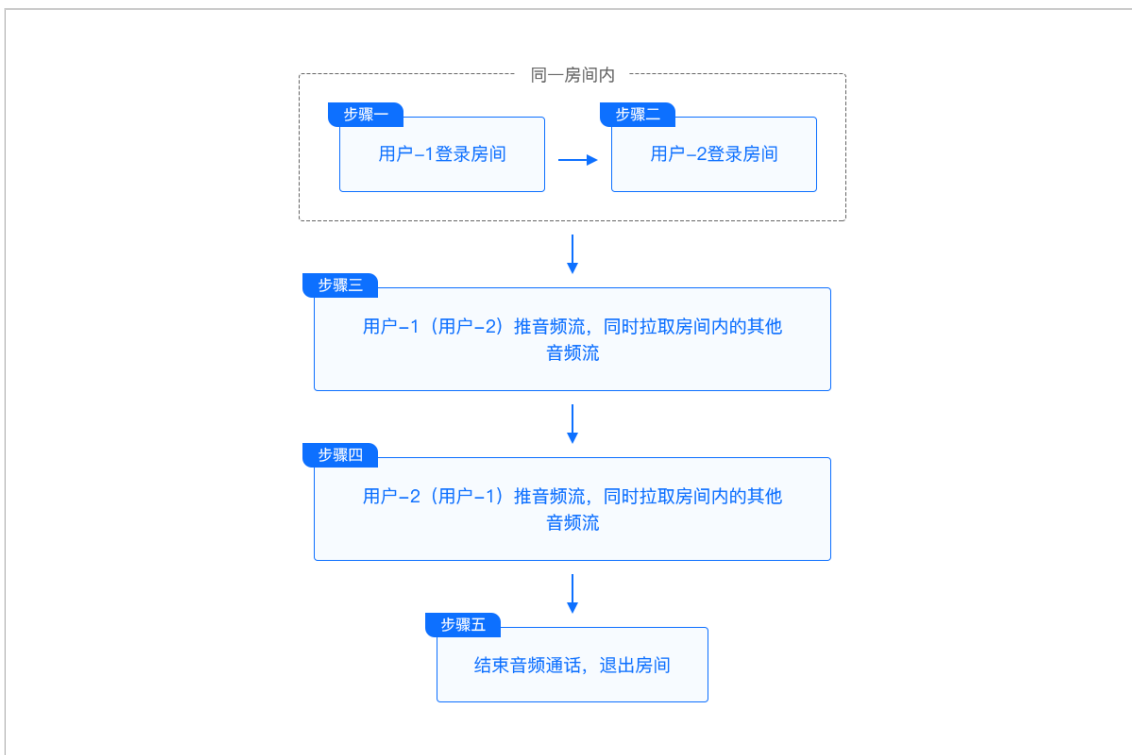
# 功能实现流程

## 1 功能说明

实时语音场景的典型之一是，同一房间中的成员进行实时语音对话。

## 2 流程说明

以 2 人间的实时语音为例，主要流程如下：



请注意：

上面流程中以 2 名房间成员间的实时语音为例，实际上 **Zego SDK 支持多人实时语音**。建议开发者按需设计。

详细步骤如下：

1. 初始化 SDK：SDK 在初始化后才可以进行登录和推拉流操作。
2. 设置回调：包括房间状态回调、推流回调、拉流回调。
3. 登录房间。
4. 推音频流：把本地的音频实时推送出去。
5. 拉音频流：播放房间内其它用户的音频。
6. 退出房间：退出房间需要停止推流、停止拉流，然后再登出房间

## 3 代码说明

### 3.1 初始化 SDK

```
/** 定义 SDK 引擎对象 */
ZegoExpressEngine engine;

/** 填写 appID 和 appSign */
long appID = ; // 请通过官网注册获取，格式为 123456789L
String appSign = ; //64个字符，请通过官网注册获取，格式为"012345678901234567
8901234567890123456789012345678901234567890123"

/** 初始化SDK，使用测试环境，通用场景接入 */
engine = ZegoExpressEngine.createEngine(appID, appSign, true, ZegoScenar
io.GENERAL, getApplication(), null);
```

请参考 [快速开始 - 实现流程](#) 的“2.1 创建引擎”。

### 3.2 设置回调

设置回调接口，需要监听处理房间状态回调接口、推流回调接口、拉流回调接口。

```
ZegoExpressEngine engine; // SDK 引擎对象
...
engine.setEventHandler(new IZegoEventHandler() {

    /** 以下为常用的房间相关回调 */

    @Override
    public void onRoomStateUpdate(String roomID, ZegoRoomState state, in
```

```

t errorCode, JSONObject extendedData) {
    /** 房间状态更新回调, 登录房间后, 当房间连接状态发生变更 (如出现房间断开, 登录认证失败等情况), SDK会通过该回调通知 */
    //....
}

@Override
public void onRoomUserUpdate(String roomID, ZegoUpdateType updateType, ArrayList<ZegoUser> userList) {
    /** 用户状态更新, 登录房间后, 当房间内有用户新增或删除时, SDK 会通过该回调通知 */
    //....
}

@Override
public void onRoomStreamUpdate(String roomID, ZegoUpdateType updateType, ArrayList<ZegoStream> streamList) {
    /** 流状态更新, 登录房间后, 当房间内有用户新推送或删除音频流时, SDK 会通过该回调通知 */
    //....
}

/** 以下为常用的推流相关回调 */

@Override
public void onPublisherStateUpdate(String streamID, ZegoPublisherState state, int errorCode, JSONObject extendedData){
    /** 调用推流接口成功后, 当推流器状态发生变更, 如出现网络中断导致推流异常等情况, SDK 在重试推流的同时, 会通过该回调通知 */
    //....
}

/** 以下为常用的拉流相关回调 */

@Override
public void onPlayerStateUpdate(String streamID, ZegoPlayerState state, int errorCode, JSONObject extendedData){
    /** 调用拉流接口成功后, 当拉流器状态发生变更, 如出现网络中断导致推流异常等情况, SDK在重试拉流的同时, 会通过该回调通知 */
    //....
}

//....
});

```

### 3.3 登录房间

用户间进行实时语音对话前, 必须先登录到同一个房间。

注意：房间 ID 只支持最大长度为 128 字节的字符串，格式仅支持数字，英文字符和 '~', '!', '@', '#', '\$', '%', '^', '&', '\*', '(', ')', '\_', '+', '=', '-', '\', ';', ':', ',', '<', '>', '/', " 特殊字符。

```
ZegoExpressEngine engine; // SDK 引擎对象
...
/** 创建用户 */
ZegoUser user = new ZegoUser("user1");

/** 开始登录房间 */
engine.loginRoom("room1", user);
```

请参考 [快速开始 – 实现流程](#) 的“2.2 登录房间”。

### 3.4 开始推音频流

登录房间成功后可推音频流。开发者可调用 `startPublishingStream` 进行推流，如果不需要继续推流，请调用 `stopPublishingStream` 停止推流。

注意：streamID 只支持长度不超过 256 的字符串，需要在整个 AppID 内全局唯一，若出现在同一个 AppID 内，不同的用户各推了一条流且流名相同，将会导致后推流的用户推流失败。不可以包含 URL 关键字，否则推拉流失败。格式仅支持数字，英文字符和 '~', '!', '@', '#', '\$', '%', '^', '&', '\*', '(', ')', '\_', '+', '=', '-', '\', ';', ':', ',', '<', '>', '/', " 特殊字符。

```
ZegoExpressEngine engine; // SDK 引擎对象
...
/** 开始推流 */
engine.startPublishingStream("stream1");
```

请参考 [快速开始 – 实现流程](#) 的“2.3 推流”。

### 3.5 播放音频流

登录房间成功后可播放直播音频流。开发者可调用 `startPlayingStream` 进行拉流，如果不需要继续拉流，请调用 `stopPlayingStream` 停止拉流。

注意：streamID 只支持长度不超过 256 的字符串。不可以包含 URL 关键字，否则推拉流失败。格式仅支持数字，英文字符和 '~', '!', '@', '#', '\$', '%', '^', '&', '\*', '(', ')', '\_', '+',

'=', '-', '\', ';', '\'', ',', '!', '<', '>', '/', " 特殊字符。

```
ZegoExpressEngine engine; // SDK 引擎对象
...
/**
 * 开始拉流，流名从登录房间后的 onRoomStreamUpdate(String roomId, ZegoUpdateType updateType, ArrayList<ZegoStream> streamList) 回调中获取，有新增/删除流时都会收到此回调
 */
engine.startPlayingStream("stream1");
```

请参考 [快速开始 - 实现流程](#) 的“2.4 拉流”。

### 3.6 结束音频直播

音频通话结束后的操作主要是登出房间、停止推流、停止拉流、清理视图或数据等，视业务需求情况来判断是否需要释放 SDK。

```
ZegoExpressEngine engine; // SDK 引擎对象
...
/** 停止推流 */
engine.stopPublishingStream();
/** 停止拉流 */
engine.stopPlayingStream(streamID);
/** 登出房间 */
engine.logoutRoom("room1");

/** 如果不需要再继续使用 SDK 可以调用如下函数释放 SDK。
 * 注意：释放 SDK 后需要再使用 SDK 时，必须重新初始化 SDK。
 */
ZegoExpressEngine.destroyEngine(null);
```