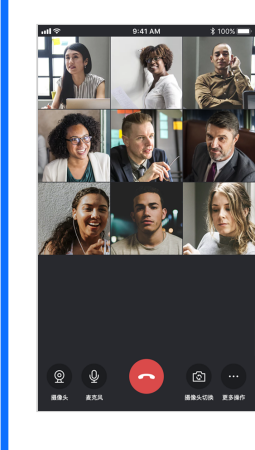
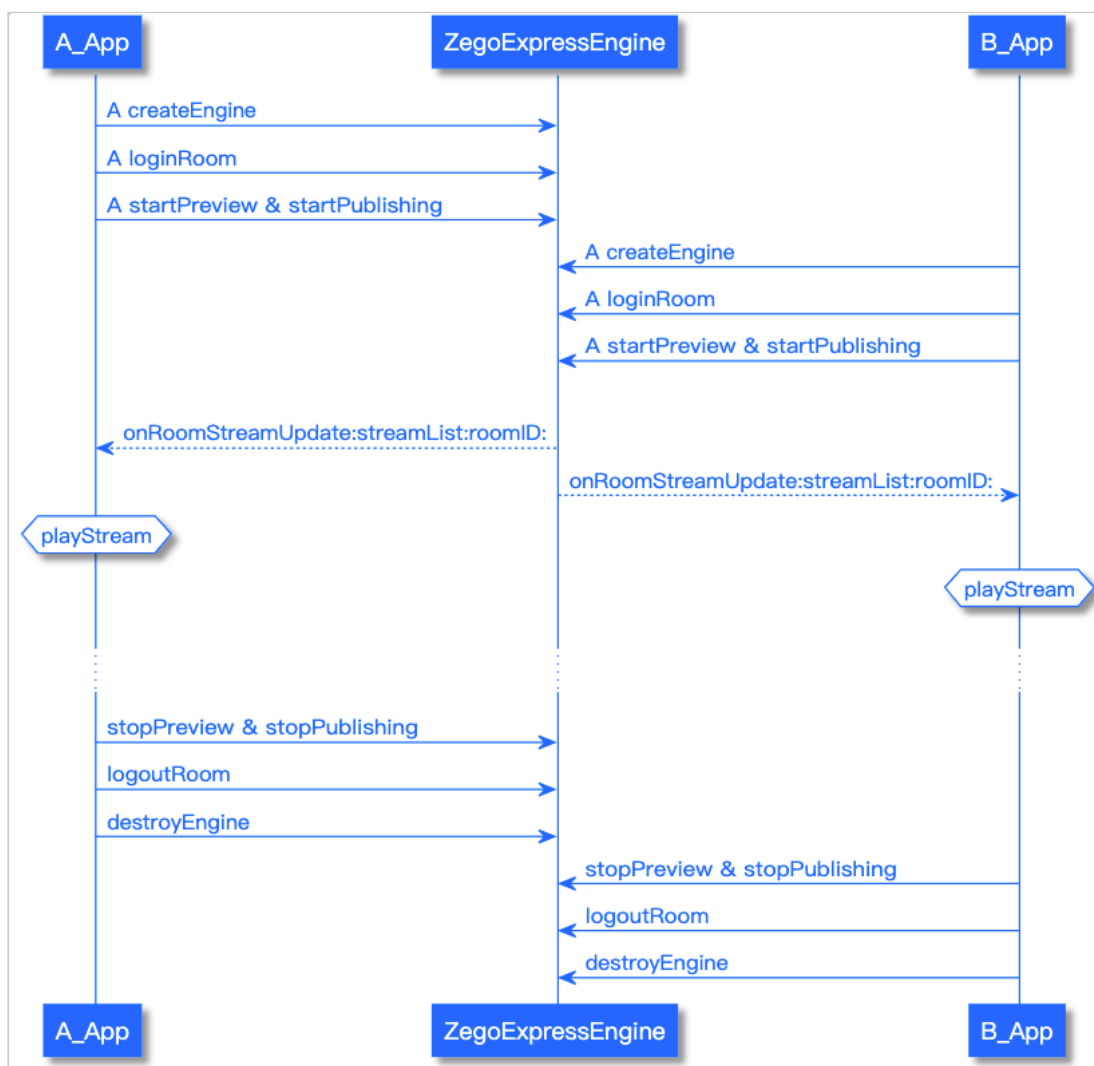


功能实现流程

实时视频场景的典型使用之一是同一会话中的多用户进行视频实时通话。即多个人一起互相推拉流，例如：



以2人实时视频通话的场景为例，Express Video SDK 的详细 API 调用时序图如下：



请注意：

1. 上面流程中以 2 名房间成员间的实时视频为例，实际上 **Express Video SDK** 支持多人实时视频。建议开发者参考上述流程设计自己的多人实时视频通话的场景。
2. 为了便于开发者更快理解 [示例专题](#) 中 [视频通话](#) 模块中的逻辑，下述每节会将功能核心源码片段挑出来并加以讲解。开发者亦可直接参考 [Demo](#)。

1 初始化 SDK

在使用 **Express Video SDK** 进行视频通话之前，需要初始化**Express Video SDK**。由于初始化操作 SDK 内部处理的操作较多，建议开发者在 App 启动的时候进行。

视频通话 模块中初始化相关源码片段演示如下，如下样例代码与 **视频通话** 模块中的源码不完全一致，仅供参考：

```
/** 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"0123456789012345678901234567890123456789012345678901234567890123"

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

2 登录房间并推拉流，监听各种回调，开始视频通话

用户间进行实时视频对话前，需要先登录到同一个房间，在收到登录房间成功的回调之后可以直接调用**Express Video SDK** 的 API 接口进行推拉流操作。

已经在房间内进行视频通话的用户需要获取后来进入房间进行视频通话的用户推流的通知，并且也需要获取中途退出视频通话的用户停止推流的通知。在获取这些通知之后，先前在房间内进行视频通话的用户应拉取房间内新增的推流、停止拉取房间内停推的流，并做相应的 UI 展示。

视频通话 模块中初始化相关源码片段演示如下，如下样例代码与 **视频通话** 模块中的源码不完全一致，仅供参考：

```
/** 创建用户 */
ZegoUser user = new ZegoUser("user1");

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

```
/** 设置并监听回调 */
engine.setEventHandler(new IZegoEventHandler() {

    /** 以下为常用的房间相关回调 */
```

```

@Override
public void onRoomStateUpdate(String roomID, ZegoRoomState state, int
errorCode, JSONObject extendedData) {
    /** 房间状态更新回调, 登陆房间后, 当房间连接状态发生变更 (如出现房间断开, 登
陆认证失败等情况), 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, 登陆房间后, 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, 登陆房间后, 当房间内有用户新推送或删除音视频流时, SDK会通过
该回调通知 */
    if(updateType == ZegoUpdateType.ADD){
        for(ZegoStream zegoStream: streamList){
            AppLogger.getInstance().i("onRoomStreamUpdate: ZegoU
pdateType.ADD streamId:"+zegoStream.streamID);
            TextureView addTextureView=new TextureView(ZGVideoTa
lKUI.this);
            int row=streamIdList.size()/2;
            int column=streamIdList.size()%2;
            addToGridLayout(row,column,addTextureView);
            viewMap.put(zegoStream.streamID,addTextureView);
            streamIdList.add(zegoStream.streamID);
            mSDKEngine.startPlayingStream(zegoStream.streamID, n
ew ZegoCanvas(addTextureView));
        }
    }else if(updateType == ZegoUpdateType.DELETE){// callback in
UIThread
        for(ZegoStream zegoStream: streamList){
            AppLogger.getInstance().i("onRoomStreamUpdate: Zego
UpdateType.DELETE streamId:"+zegoStream.streamID);
            mSDKEngine.stopPlayingStream(zegoStream.streamID);
            streamIdList.remove(zegoStream.streamID);
            notifyGridLayout();
            viewMap.remove(zegoStream.streamID);
        }
    }
}
//....
});

```

在onRoomStreamUpdate回调中，我们可以获取流的增加/移除信息，流增加时进行拉流播放 `startPlayingStream`，流移除时停止拉流 `stopPlayingStream`

登录房间流程请参考：[快速开始 – 实现流程](#) 的“2.2 登录房间”。

推流流程请参考：[快速开始 – 实现流程](#) 的“2.3 推流”。

注意：

1. `onPublisherStateUpdate` 回调可监听推流状态。若推流不成功，一般为网络问题，SDK内部会做重试工作，开发者也可根据情况做有限次数的推流重试，或给出友好的交互提示。

拉流流程请参考：[快速开始 – 实现流程](#) 的“2.4 拉流”。

注意：

1. `onPlayerStateUpdate` 可监听拉流状态。若拉流不成功，一般为网络问题，SDK内部会做重试工作，开发者也可根据情况做有限次数的拉流重试，或给出友好的交互提示。

3 停止视频通话

用户在视频通话过程中退出视频通话，应该停止推流、停止拉流、退出房间，释放对应的 UI 对象资源等。

`视频通话` 模块中初始化相关源码片段演示如下，如下样例代码与 `视频通话` 模块中的源码不完全一致，仅供参考：

```
/** 停止推流 */
engine.stopPublishingStream();
/** 停止本地预览 */
engine.stopPreview();

for(String playStreamid : playStreamids){
    // 将所拉的多条流停拉，这里开发者在业务层应自己维护所拉的流的列表，例如这里的 playStreamids
    // 停止拉流
    engine.stopPlayingStream(streamID);
    // 将相应拉流渲染的View设置为null、释放资源等
```

```
    ...;  
}  
/** 登出房间 */  
engine.logoutRoom("room1");
```