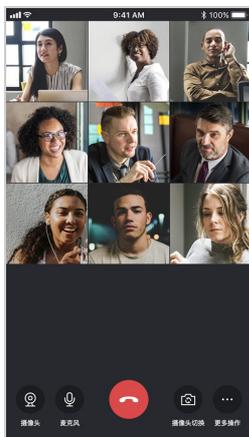
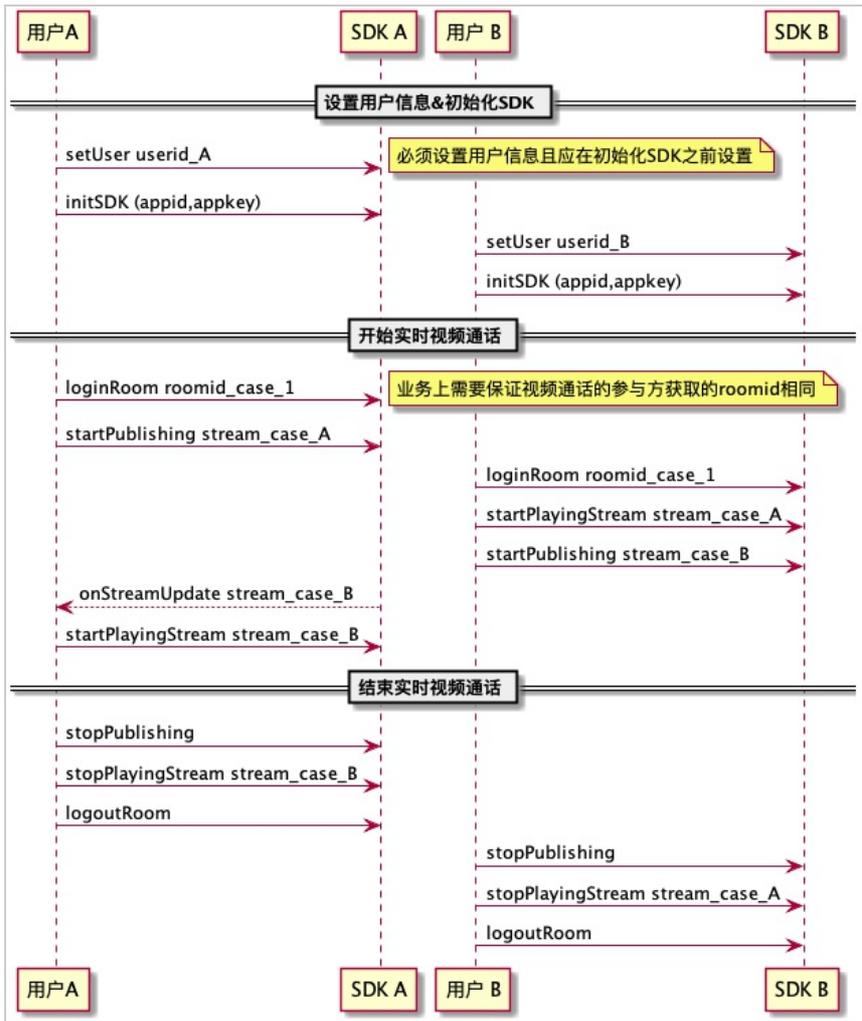


功能实现流程

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，ZEGO 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 **ZEGO 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 [视频通话模块的源码 \(https://github.com/zegodev/liveroom-topics-android/tree/master/src/LiveRoomPlayground/videoCommunica](https://github.com/zegodev/liveroom-topics-android/tree/master/src/LiveRoomPlayground/videoCommunica)

1 初始化 SDK

在使用 ZEGO SDK 进行视频通话之前，需要初始化 ZEGO 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：

```
// zegoLiveRoom 为 ZegoLiveRoom 的单例对象
// appid 与 appSign 为 ZEGO 分配的标识，是使用 ZEGO SDK 的必要条件，可通过 <a>https://console.zego.im/acount/login</a> 申请 appID 与 appSign
zegoLiveRoom.initSDK(appID, appSign, new IZegoInitSDKCompletionCallback(
) {
    @Override
    public void onInitSDK(int i) {

        // 初始化完成
        if (i == 0) {
            // 成功初始化，开发者可在此设置一些需要在推拉流之前设置的音视频参数或者在此设置相关的代理

        } else {

            // 初始化失败的处理，一般为客户端网络问题导致，SDK内部会做重试工作，开发者也可在此做有限次数初始化重试，或给出友好的交互提示

            // 当初始化失败时释放SDK，避免下次再次初始化SDK会收不到回调
            zegoLiveRoom.unInitSDK();
        }
    }
});
```

2 设置监听房间内实时通话用户推流变化的代理

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：

```
ZGVideoCommunicationHelper.java

/**
 * 通过设置sdk房间代理，可以收到房间内的一些信息回调。
 * 开发者可以按自己的需求在回调里实现自己的业务
 */
zegoLiveRoom.setZegoRoomCallback(new IZegoRoomCallback() {

    ...

    /**
     * 房间内推流变化的回调，开发者应监听此回调来在视频通话场景中拉别人的流或停拉别人的流
     *
     * @param type 流增加还是流减少的类型
     * @param listStream 变化的流列表
     * @param roomID 房间id
     */
    @Override
    public void onStreamUpdated(int type, ZegoStreamInfo[] listStream, String roomID) {

        // 当登陆房间成功后，如果房间内中途有人推流或停止推流。房间内其他人就能通过该回调收到流更新通知。
        for (ZegoStreamInfo streamInfo : listStream) {
            if (type == ZegoConstants.StreamUpdateType.Added) {
                // playRenderView 为开发者业务层传入的用于拉流渲染的 TextureView
                zegoLiveRoom.startPlaying(streamInfo.streamID, playRenderView);
            }
        }
    }
});
```

```

        // ZegoVideoViewMode.ScaleAspectFill 为渲染模式的一种，
        具体可以参考 https://doc.zego.im/API/HideDoc/ViewMode/ViewMode.html
        zegoLiveRoom.setPlayViewMode(ZegoVideoViewMode.ScaleAspectFill, streamInfo.streamID);
        // 开发者可在此做其他业务相关的UI操作
        ...;

    } else if (type == ZegoConstants.StreamUpdateType.Deleted) {
        // 停止拉对应的流
        zegoLiveRoom.stopPlaying(streamInfo.streamID);
        // 开发者可在此做其他业务相关的UI操作，停止拉流之后应该
        将对应渲染的View置为null，以防止内存泄漏
        ...;

    }

}

}

...
});

```

3 登录房间并推拉流，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 **ZEGO SDK** 的 **API** 接口进行推拉流操作。

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：

```

ZGVideoCommunicationHelper.java

// zegoLiveRoom 为 ZegoLiveRoom 的单例对象

```

```

zegoLiveRoom.loginRoom(roomID, ZegoConstants.RoomRole.Anchor, new IZegoLoginCompletionCallback() {
    @Override
    public void onLoginCompletion(int i, ZegoStreamInfo[] zegoStreamInfos) {
        // zegoStreamInfos, 内部封装了 userID、userName、streamID 和 extraInfo。
        // 登录房间成功后, 开发者可通过 zegoStreamInfos 获取到当前房间推流信息。便于后续的拉流操作
        // 当 zegoStreamInfos 为 null 时说明当前房间没有人推流

        if(0 == i) {

            // 在视频通话场景中, 登录成功之后可以直接调用SDK的接口直接进行推流操作, 并做UI展示。

            // ZegoVideoViewMode.ScaleAspectFill 为渲染模式的一种, 具体可以参考 https://doc.zego.im/API/HideDoc/ViewMode/ViewMode.html
            zegoLiveRoom.setPreviewViewMode(ZegoVideoViewMode.ScaleAspectFill);
            // localPreviewView 为开发者业务层传入的用于本地推流渲染的 TextureView
            zegoLiveRoom.startPreview(localPreviewView);
            // publishStreamid 为业务定义的推流的流id, 开发者需保证流id 唯一, 否则会造成重复流id推流失败的问题
            // 在视频通话场景中, 推流的flag需为 ZegoConstants.PublishFlag.JoinPublish
            zegoLiveRoom.startPublishing(publishStreamid, publishStreamid + "-title", ZegoConstants.PublishFlag.JoinPublish);
            // 开发者可在此做其他业务相关的UI操作
            ...;

            // 在视频通话场景中, 登录成功之后可以直接调用SDK的接口直接进行拉流操作, 并做UI展示。

            for(ZegoStreamInfo zegoStreamInfo : zegoStreamInfos) {
                // playRenderView 为开发者业务层传入的用于拉流渲染的 TextureView
                zegoLiveRoom.startPlayingStream(zegoStreamInfo.streamID, playRenderView);
                // ZegoVideoViewMode.ScaleAspectFill 为渲染模式的一种,

```

```
具体可以参考 https://doc.zego.im/API/HideDoc/ViewMode/ViewMode.html
        zegoLiveRoom.setPlayViewMode(ZegoVideoViewMode.ScaleAspectFill, zegoStreamInfo.streamID);
        // 开发者可在此做其他业务相关的UI操作
        ...;
    }

    }else {
        // 登录失败的处理，一般为客户端网络问题导致，SDK内部会做重试工作，开发者也可在此做有限次数的登录重试，或给出友好的交互提示，提示用户重新登录房间
    }
}
});
```

如上在登录房间的回调中获取的流信息的数组为当前所登录的房间里正在推的流，因此在获取这些流id之后应直接对其进行拉流操作；通过 [2 设置监听房间内实时通话用户变化的代理 \(/398.html#2\)](#) 获取的流信息为之后加入房间的用户所推的流信息或中途退出房间的用户停推的流信息，因此应在 `onStreamUpdate` 的回调中拉流或者停拉对应的流。

登录房间流程请参考：[快速集成-登录房间](#)

推流流程请参考：[快速集成-推流](#)

注意：

1. 在推流之前应该调用 `zegoLiveRoom.setZegoLivePublisherCallback(...)` 设置推流的回调，以监听推流是否成功。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：[快速集成-拉流](#)

注意：

1. 在拉流之前应该调用 `zegoLiveRoom.setZegoLivePlayerCallback(...)` 设置拉流的回调，以监听拉流是否成功。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。

4 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：

```
ZGVideoCommunicationHelper.java

// 停止推流
zegoLiveRoom.stopPublishing();
// 停止推流本地渲染
zegoLiveRoom.stopPreviewView();
// 将相应本地渲染的View设置为null、释放资源等
...;
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
    zegoLiveRoom.stopPlaying(playStreamid);
    // 将相应拉流渲染的View设置为null、释放资源等
    ...;
}
zegoLiveRoom.logoutRoom();
```