

- [1 功能概述](#)
- [2 接口介绍](#)
- [3 实现介绍](#)
 - [3.1 初始化页面参数](#)
 - [3.2 初始化直播间数据管理器](#)
 - [3.3 初始化页面UI](#)
 - [3.4 设置布局回调](#)

1 功能概述

手机开播场景的界面实现为 **PLVLSLiveStreamerActivity**

手机开播场景支持的功能有推流、连麦、聊天、文档。

2 接口介绍

```
/**
 * 启动手机开播页
 *
 * @param activity      上下文Activity
 * @param channelId     频道号
 * @param viewerId      开播者ID
 * @param viewerName    开播者昵称
 * @param avatarUrl     开播者头像url
 * @param actor         开播者头衔
 * @param isOpenMic     是否打开麦克风
 * @param isOpenCamera  是否打开相机
 * @param isFrontCamera 是否使用前置摄像头
 * @return PLVLaunchResult.isSuccess=true表示启动成功，PLVLaunchResult.isSuccess=false表示启动失败
 */
public static PLVLaunchResult launchStreamer(@NonNull Activity activity,
                                             @NonNull String channelId,
                                             @NonNull String viewerId,
                                             @NonNull String viewerName,
                                             @NonNull String avatarUrl,
                                             @NonNull String actor,
                                             boolean isOpenMic,
                                             boolean isOpenCamera,
                                             boolean isFrontCamera);
```

3 实现介绍

手机开播将每个功能模块封装成一个布局Layout类，在Activity中通过简单地布局组合即可完成集成。

3.1 初始化页面参数

```
private void initParams() {  
    // 获取输入数据  
    Intent intent = getIntent();  
    String channelId = intent.getStringExtra(EXTRA_CHANNEL_ID);  
    String viewerId = intent.getStringExtra(EXTRA_VIEWER_ID);  
    String viewerName = intent.getStringExtra(EXTRA_VIEWER_NAME);  
    String avatarUrl = intent.getStringExtra(EXTRA_AVATAR_URL);  
    String actor = intent.getStringExtra(EXTRA_ACTOR);  
    // 设置Config数据  
    PLVLiveChannelConfigFiller.setupUser(viewerId, viewerName, avatarUrl,  
    PLVSocketUserConstant.USER_TYPE_TEACHER, actor);  
    PLVLiveChannelConfigFiller.setupChannelId(channelId);  
}
```

3.2 初始化直播间数据管理器

```
private void initLiveRoomManager() {  
    // 使用PLVLiveChannelConfigFiller配置好直播参数后，用其创建直播间数据管理器实例  
    liveRoomDataManager = new PLVLiveRoomDataManager(PLVLiveChannelConfigFiller.generateNewChannelConfig());  
    // 进行网络请求，获取直播详情数据  
    liveRoomDataManager.requestChannelDetail();  
}
```

3.3 初始化页面UI

```
private void initView() {
    // 初始化推流和连麦布局
    plvlsStreamerLy.init(liveRoomDataManager);
    // 初始化推流的媒体配置
    boolean isOpenMic = getIntent().getBooleanExtra(EXTRA_IS_OPEN_MIC, true);
    boolean isOpenCamera = getIntent().getBooleanExtra(EXTRA_IS_OPEN_CAMERA, true);
    boolean isFrontCamera = getIntent().getBooleanExtra(EXTRA_IS_FRONT_CAMERA, true);
    plvlsStreamerLy.enableRecordingAudioVolume(isOpenMic);
    plvlsStreamerLy.enableLocalVideo(isOpenCamera);
    plvlsStreamerLy.setCameraDirection(isFrontCamera);
    // 初始化状态栏布局
    plvlsStatusBarLy.init(liveRoomDataManager);
    // 注册成员列表中的streamerView，并请求成员列表接口
    plvlsStreamerLy.getStreamerPresenter().registerView(plvlsStatusBarLy.getMemberLayoutStreamerView());
    plvlsStreamerLy.getStreamerPresenter().requestMemberList();
    // 初始化聊天室布局
    plvlsChatroomLy.init(liveRoomDataManager);
    // 初始化文档布局
    plvlsDocumentLy.init(liveRoomDataManager);
    // 进入横屏模式
    PLVScreenUtils.enterLandscape(this);
}
```

Demo中组合了推流、状态栏、成员列表、聊天室、文档布局。

各个功能模块的布局接口如下：

推流连麦布局接口：IPLVLSStramerLayout

状态栏布局接口：IPLVLSStatusBarLayout

聊天室布局接口：IPLVLSChatroomLayout

文档布局接口：IPLVLSDocumentLayout

3.4 设置布局回调

由于各个功能之间需要进行通信，例如状态栏的点击文档按钮需要文档列表的显示响应，因此每个布局都提供了外部可以使用的回调，方便各个功能模块进行通信。

详细接口介绍请参考各个功能模块的具体介绍文章。

回调设置和处理请参考Activity中的[设置布局回调方法块](#)。