

- [1 功能概述](#)
- [2 使用介绍](#)
- [3 页面介绍](#)
 - [3.1 页面 UI](#)
 - [3.2 直播间数据监听](#)

1 功能概述

手机开播场景的页面实现为 **PLVLSSreamerViewController**

手机开播场景支持的功能有推流、连麦、聊天、PPT&白板、文档管理。

2 使用介绍

由于频道信息、登录用户账号信息等手机开播页面所需要的初始化参数，在登录时已经通过类 **PLVRoomLoginClient** 进行了配置，并在 **PLVRoomLoginClient** 内部使用 **PLVRoomDataManager** 类的单例对这些直播间数据信息进行了管理，进入手机开播场景的页面只需要以下代码：

```
PLVLSSreamerViewController *vctrl = [[PLVLSSreamerViewController alloc] init];
vctrl.modalPresentationStyle = UIModalPresentationFullScreen;
[self presentViewController:vctrl animated:YES completion:nil];
```

离开手机开播页面除了把页面从页面堆栈移出，务必调用 **PLVRoomLoginClient** 的登出接口，移除 **PLVRoomDataManager** 单例对当前直播间数据的持有和监听：

```
[PLVRoomLoginClient logout];
[self dismissViewControllerAnimated:YES completion:nil];
```

3 页面介绍

3.1 页面 UI

手机开播场景页面布局，由以下页面 UI 控件组成：

```
@property (nonatomic, strong) PLVLSStatusAreaView *statusAreaView;    // 顶部状态栏区域
@property (nonatomic, strong) UIView *streamerAreaView;              // 右侧推流&连麦区域
@property (nonatomic, strong) PLVLSDocumentAreaView *documentAreaView; // 左侧白板&PPT区域
@property (nonatomic, strong) PLVLSChatroomAreaView *chatroomAreaView; // 左下角聊天室区域
```

其次，手机开播页面还包含以下（不长驻）UI 控件：

```
@property (nonatomic, strong) PLVLSChannelInfoSheet *channelInfoSheet; // 频道信息弹层
@property (nonatomic, strong) PLVLSSettingSheet *settingSheet;           // 设置弹层
@property (nonatomic, strong) PLVLSMemberSheet *memberSheet;           // 成员管理弹层
@property (nonatomic, strong) PLVLSCountDownView *countDownView; // 开始上课时的倒数蒙层
```

最后，手机开播页面不支持竖屏，只支持横屏，因此需对父类 `UIViewController` 的以下方法进行覆写：

```
- (BOOL)shouldAutorotate {
    return YES;
}

- (UIInterfaceOrientationMask)supportedInterfaceOrientations {
    return UIInterfaceOrientationMaskLandscapeLeft | UIInterfaceOrientationMaskLandscapeRight;
}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {
    return UIInterfaceOrientationLandscapeLeft;
}
```

3.2 直播间数据监听

页面需要对直播间数据 `PLVRoomData` 进行实时监听并作出相应的 UI 更新，首先，需要导入头文件 `#import "PLVRoomDataManager.h"`，并让 `PLVLSStreamerViewController` 遵循协议 `PLVRoomDataManagerProtocol`，然后添加监听，代码示例如下：

```
[[PLVRoomDataManager sharedManager] addDelegate:self delegateQueue:dispatch_get_main_queue()];
```

代码示例中，`delegateQueue` 参数传入 `dispatch_get_main_queue()` 表示希望回调方法从主线程执行，这是由于当前页面相关回调里执行的操作都是更新 UI。

关于 `PLVRoomDataManager` 的更多介绍，详见文档《6_2 核心common-数据》。

手机开播页面需要监听的 `delegate` 方法如下：

```
#pragma mark - PLVRoomDataManager Protocol

/// 菜单信息 menuInfo 更新
- (void)roomDataManager_didMenuInfoChanged:(PLVLiveVideoChannelMenuInfo *)menuInfo {
    // 此时更新频道信息弹层数据不能使用self，会过早触发弹层初始化，导致弹层高度计算错误
    [_channelInfoSheet updateChannelInfoWithData:menuInfo];
}
```

监听的移除在 2 使用介绍 里面提到的 `PLVRoomLoginClient` 的登出方法 `+logout` 内部进行，所以无需额外做移除监听的操作。