1 功能概述

带货场景的页面实现为 **PLVECWatchRoomViewController**

带货场景下定义的 直播模式、回放模式 的 共用界面。 直播支持的功能有：播放器、聊天室、商品、打赏、互动应用。 回放支持的功能有：播放器。

2 使用介绍

由于频道信息、登录用户账号信息等带货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，进入带货场景的页面只需要以下代码：

```
PLVECWatchRoomViewController *vctrl = [[PLVECWatchRoomViewController alloc] init];
[self.navigationController pushViewController:vctrl animated:YES];
```

离开带货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.navigationController popViewControllerAnimated:YES];
```

3 页面介绍

3.1 页面 UI

带货场景页面，不管是直播还是回放，都由以下页面 UI 控件组成：

```
@property (nonatomic, strong) PLVECPlayerViewController *playerVC; // 页面底部播放控制器
@property (nonatomic, strong) UIScrollView *scrollView; // 播放器顶部横行透明滚动视图
@property (nonatomic, strong) PLVECLiveDetailPageView *liveDetailPageView; // 滚动视图第一屏
@property (nonatomic, strong) PLVECHomePageView *homePageView; // 滚动视图第二屏
@property (nonatomic, strong) UIButton *closeButton; // 右上角关闭按钮
```

其中，`scrollView` 有三屏，第一屏 `liveDetailPageView` 是显示公告信息、直播介绍，第二屏 `homePageView` 如果是直播，显示聊天室、点赞按钮、商品按钮、购物车按钮、礼物按钮等互动控件，如果是回放，显示回放视频进度条和切换视频播放速率等按钮等互动控件，第三屏为空白，用于用户希望无干扰观看时使用。进入页面默认显示第二屏。播放器位于 `scrollView` 底部，关闭按钮位于 `scrollView` 上方。

视图结构如下：

```
.
└── PLVECWatchRoomViewController
    ├── PLVECPlayerViewController
    ├──  UIScrollView
    │    ├── PLVECLiveDetailPageView
    │    └── PLVECHomePageView
    └── UIButton
```

其次，带货场景页面不支持横屏，只支持竖屏，因此需对父类 `UIViewController` 的以下方法进行覆写：

```
- (BOOL)shouldAutorotate {
    return NO;
}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation {
    return UIInterfaceOrientationPortrait;
}

- (UIInterfaceOrientationMask)supportedInterfaceOrientations {
    return UIInterfaceOrientationMaskPortrait;
}
```

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 `PLVECWatchRoomViewController` 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》。

带货场景页面需要监听的 delegate 方法如下：

```
#pragma mark PLVRoomDataManager Protocol

/// 在线人数 onlineCount 更新
- (void)roomDataManager_didOnlineCountChanged:(NSUInteger)onlineCount {

}

/// 点赞数 likeCount 更新
- (void)roomDataManager_didLikeCountChanged:(NSUInteger)likeCount {

}

/// 播放状态 playing 更新
- (void)roomDataManager_didPlayingStatusChanged:(BOOL)playing {

}

/// 菜单信息 menuInfo 更新
- (void)roomDataManager_didMenuInfoChanged:(PLVLiveVideoChannelMenuInfo *)menuInfo {

}

/// 直播状态 liveState 更新
- (void)roomDataManager_didLiveStateChanged:(PLVChannelLiveStreamState)liveState {

}
```

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。