

- [1 概述](#)
- [2 核心类介绍](#)
 - [2.1 PLVRoomData](#)
 - [2.2 PLVRoomDataManager](#)

1 概述

该模块位于文件夹 PolyvLiveCommonModule/Modules/Room 下，管理的数据包括：配置数据、http数据、业务模块数据等。这些数据合并在一起，使用数据模型类 `PLVRoomData` 进行数据获取或存储，并使用 `PLVRoomDataManager` 作为对外的数据路由器，当数据发生变化，通知各个已注册监听的业务模块。

2 核心类介绍

2.1 PLVRoomData

数据模型类 `PLVRoomData` 对外提供了以下几个接口，接口返回的数据由 `PLVRoomData` 持有：

```
/// 设置 roomUser
- (void)setupRoomUser:(PLVRoomUser *)roomUser;

/// 获取频道菜单信息
- (void)requestChannelDetail:(void(^)(PLVLiveVideoChannelMenuInfo *channelMenuInfo))completion;

/// 上报观看热度
- (void)reportViewerIncrease;

/// 获取功能开关
- (void)requestChannelFunctionSwitch;
```

除了接口 `-requestChannelFunctionSwitch` 在初始化聊天室时（可选）进行请求，其他接口（必选）均在登录时进行调用，使用实例详见登录管理器 `PLVRoomLoginClient`。

2.2 PLVRoomDataManager

数据路由器 `PLVRoomDataManager` 是单例，负责持有每个直播间只允许存在一个实例的 `PLVRoomData` 对象，并对持有的 `roomData` 进行 KVO 监听，当 `roomData` 管理的数据发生变化，通知各个已注册监听的业务模块。

数据路由器 `PLVRoomDataManager` 中，`roomData` 的属性相关接口定义如下：

```

/// 当前活跃的直播间数据
@property (nonatomic, strong, readonly) PLVRoomData *roomData;

/// 单例方法
+ (instancetype)sharedManager;

/// 配置当前的直播间数据，进入直播间时调用
- (void)configRoomData:(PLVRoomData *)roomData;

/// 移除当前的直播间数据，离开直播间时调用
- (void)removeRoomData;

```

使用示例如下，接口调用上下文详见登录管理器 `PLVRoomLoginClient`：

```

// 登录时调用
[[PLVRoomDataManager sharedManager] configRoomData:roomData];
// 退出登录时调用
[[PLVRoomDataManager sharedManager] removeRoomData];

```

数据监听注册、移除监听相关接口如下：

```

/// 增加PLVRoomDataManagerProtocol协议的监听者
/// @param delegate 待增加的监听者
/// @param delegateQueue 执行回调的队列
- (void)addDelegate:(id<PLVRoomDataManagerProtocol>)delegate delegateQueue:(dispatch_queue_t)delegateQueue;

/// 移除PLVRoomDataManagerProtocol协议的监听者
/// @param delegate 待移除的监听者
- (void)removeDelegate:(id<PLVRoomDataManagerProtocol>)delegate;

/// 移除PLVRoomDataManagerProtocol协议的所有监听者
- (void)removeAllDelegates;

```

接口 `-removeAllDelegates` 在数据路由器 `PLVRoomDataManager` 内部已在调用 `-removeAllDelegates` 移除当前持有的直播间数据时进行了调用，只需要在各个需要监听 `roomData` 数据的业务模块、功能页面调用 `-addDelegate:delegateQueue:` 接口，在不需要监听的时候（譬如页面退出、模块注销时）调用 `-removeDelegate:` 接口。