

- 1 功能概述
- 2 核心类介绍
 - 2.1 PLVLCChatroomViewModel
 - 2.1.1 生命周期
 - 2.1.2 获取聊天记录
 - 2.1.3 发送消息
 - 2.1.4 消息数组
 - 2.1.5 监听与回调

1 功能概述

聊天室模块包括发言、点赞、历史记录、欢迎语等功能。聊天室模块在UI、交互、功能上，相较于其他模块，都会更复杂更庞大。因此设计、搭建、维护一个聊天室，也是一件较为费时费力的事情。我们推荐直接使用保利威封装好的聊天室模块，该部分代码完全开源，支持直接使用，以及二次开发。

2 核心类介绍

2.1 PLVLCChatroomViewModel

`PLVLCChatroomViewModel` 是云课堂场景 Scene 层聊天室的核心类，由于一个应用只允许存在一个聊天室，所以 `PLVLCChatroomViewModel` 采用的是单例的模式，负责创建、持有、销毁 Common 层聊天室核心类 `PLVChatroomPresenter` 的实例对象以及与 Common 层聊天室模块进行通信，核心类 `PLVLCChatroomViewModel` 提供功能如下：

1. 对 View 层提供发送消息的接口；
2. 管理 Common 层返回的消息模型；
3. 在 View 层需要刷新 UI、更新列表数据时，通过回调通知 View 层。

2.1.1 生命周期

`PLVLCChatroomViewModel` 是单例类，进入直播间启动聊天室的时候调用 - `setup` 方法，离开直播间时调用 - `clear` 方法，代码如下：

```
// 使用新的直播间数据启动聊天室管理器
[[PLVLCChatroomViewModel sharedViewModel] setup];
// 退出前调用，用于资源释放、状态位清零
[[PLVLCChatroomViewModel sharedViewModel] clear];
```

代码注释提到的“直播间数据”指的是 `PLVRoomDataManager` 单例持有的 `roomData`。

2.1.2 获取聊天记录

Common 层聊天室核心类 `PLVChatroomPresenter` 在初始化完毕之后，会自动获取第一页的聊天记录，获取更多聊天记录代码如下：

```
[[PLVLCChatroomViewModel sharedViewModel] loadHistory];
```

`-loadHistory` 使用在 2.1.1 的 `-setup` 方法里初始化的 presenter 对象，调用 Common 层聊天室模块获取聊天室记录。

2.1.3 发送消息

`PLVLCChatroomViewModel` 类提供带货场景所需的发送消息的API，具体接口定义如下：

```
/// 发送私聊提问消息
/// @param content 消息文本
/// @return YES表示数据将有更新，可等待收到回调后刷新列表；NO表示socket未登录或房间关闭，可进行toast提示
- (BOOL)sendQuesstionMessage:(NSString *)content;

/// 发送文本消息
/// @param content 消息文本
/// @return YES表示数据将有更新，可等待收到回调后刷新列表；NO表示socket未登录或房间关闭，可进行toast提示
- (BOOL)sendSpeakMessage:(NSString *)content;

/// 发送图片消息
/// @param image 图片
/// @return YES表示数据将有更新，可等待收到回调后刷新列表；NO表示socket未登录或房间关闭，可进行toast提示
- (BOOL)sendImageMessage:(UIImage *)image;

/// 发送点赞消息
/// 点赞数的实时更新通过监听roomData的likeCount获得
- (void)sendLike;
```

2.1.4 消息数组

`PLVLCChatroomViewModel` 类提供以下消息列表数组，作为聊天室 View 层的数据源：

```
/// 公聊消息数组
@property (nonatomic, strong, readonly) NSMutableArray <PLVChatModel *> *chatArray;
/// 私聊消息数组
@property (nonatomic, strong, readonly) NSMutableArray <PLVChatModel *> *privateChatArray;
```

2.1.5 监听与回调

`PLVLCChatroomViewModel` 允许设置多代理监听，提供了以下接口增加监听和移除监听：

```
/// 增加PLVLCChatroomViewModelProtocol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<PLVLCChatroomViewModelProtocol>)delegate delegateQueue:(dispatch_queue_t)delegateQueue;

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

上面两个接口提到的协议 `PLVLCChatroomViewModelProtocol` 提供了以下这些代理方法，这些方法均是用于在 View 层需要刷新 UI、更新列表数据时，通过回调通知 View 层：

```

@protocol PLVLCChatroomViewModelProtocol <NSObject>

@optional

#pragma mark 私聊

/// 本地发送了新的私聊消息
/// 用于刷新列表
- (void)chatroomManager_didSendQuestionMessage;

/// 通知socket接收到新的私聊（教师回答）消息，每次1条
/// 用于刷新列表、显示新消息提示
- (void)chatroomManager_didReceiveAnswerMessage;

#pragma mark 公聊

/// 返回本地发送的公聊消息（包含禁言的情况）
/// 用于刷新列表、发送弹幕
/// @param model 消息模型，不为空
- (void)chatroomManager_didSendMessage:(PLVChatModel *)model;

/// 返回socket接收到的公聊消息
/// 用于刷新列表、发送弹幕、显示新消息提示
/// @param modelArray 消息队列，不为空
- (void)chatroomManager_didReceiveMessages:(NSArray <PLVChatModel *> *)modelArray;

/// socket通知有消息被删除（1条或多条）
/// 用于刷新列表
- (void)chatroomManager_didMessageDeleted;

/// 获取历史聊天记录成功时触发
/// 用于刷新列表，停止【下拉加载更多】控件的动画
/// @param noMore 是否还有更多历史消息，YES表示已加载完，此时可隐藏【下拉加载更多】控件
/// @param first 是否是初次加载历史消息，初次加载需滚动列表到底部
- (void)chatroomManager_loadHistorySuccess:(BOOL)noMore firstTime:(BOOL)first;

/// 获取历史聊天消息失败时触发
/// 用于停止【下拉加载更多】控件的动画
- (void)chatroomManager_loadHistoryFailure;

/// 如果4秒内有登录聊天室的用户（包括自己），间隔4秒触发一次
/// @param userArray 4秒内登录聊天室的用户数组，如果为nil，表示当前时间段内当前用户有登录事件
- (void)chatroomManager_loginUsers:(NSArray <PLVChatUser *> * _Nullable )userArray;

/// 上报管理员发布的消息文本，间隔8秒触发一次
/// @param content 管理员消息文本
- (void)chatroomManager_managerMessage:(NSString * )content;

/// 上报需插入弹幕的文本，间隔1秒触发一次
/// @param content 弹幕文本
- (void)chatroomManager_danmu:(NSString * )content;

@end

```