

- 1 功能概述
- 2 核心类介绍
 - 2.1 PLVChatroomPresenter
 - 2.1.1 初始化
 - 2.1.2 设置监听器
 - 2.1.3 获取聊天记录
 - 2.1.4 发送消息
 - 2.1.5 销毁

1 功能概述

该模块位于文件夹 PolyvLiveCommonModule/Modules/Chatroom 下，采用 MVP 模式设计，将 View 层对 SDK 层的聊天室 `chatroomManager` 的直接调用隔离开，并将多个场景中对 `chatroomManager` 的共用代码抽离封装起来，使 View 层的逻辑更简洁。

2 核心类介绍

2.1 PLVChatroomPresenter

`PLVChatroomPresenter` 类主要负责 socket 的登录、socket 消息的接收与发送，同时还提供以下功能：

1. 对直播间数据的在线人数、观看热度、点赞数进行实时更新；
2. 提供发送各类消息的API，并把消息封装成数据模型返回；
3. 加载历史聊天记录，并把加载结果与数据封装成数据模型通过回调通知 Scene 层；
4. 监听 socket 关于聊天消息的接收与删除，并把新消息封装成数据模型，然后通过回调通知 Scene 层。

2.1.1 初始化

在 Scene 层对 `PLVChatroomPresenter` 进行初始化，初始化方法定义如下：

```
/// 初始化方法
/// @param count 每次调用接口获取的聊天消息条数，不得小于1
/// @param allow 是否允许使用分房间功能
- (instancetype)initWithLoadingHistoryCount:(NSInteger)count childRoomAllow:(BOOL)allow;
```

也可使用 `-init` 方法初始化，此时每次调用接口获取的聊天消息条数为默认值 20，默认不允许使用分房间功能。

2.1.2 设置监听器

`PLVChatroomPresenter` 提供了属性 `delegate` 用于设置回调监听，初始化 `presenter` 并设置监听的示例代码如下（更多细节参考 Demo 的 `PLVLCChatroomViewModel` 类）：

```
// 初始化聊天室Presenter并设置delegate
self.presenter = [[PLVChatroomPresenter alloc] initWithLoadingHistoryCount:10 childRoomAllow:YES];
self.presenter.delegate = self;
```

属性 `delegate` 遵循协议 `PLVChatroomPresenterProtocol`，用于通知 Scene 层聊天室模块的相关 socket 消息和 Http 接口返回的聊天室相关数据（详见 2.1.3）。协议 `PLVChatroomPresenterProtocol` 的接口定义如下：

```
/* PLVChatroomPresenter的协议 */
@protocol PLVChatroomPresenterProtocol <NSObject>

@optional

/// 获取历史聊天消息成功时触发
/// @param modelArray 聊天消息队列
/// @param noMore 是否还有更多历史消息，YES表示已加载完
- (void)chatroomPresenter_loadHistorySuccess:(NSArray <PLVChatModel *> *)modelArray noMore:(BOOL)noMore;

/// 获取历史聊天消息失败时触发
- (void)chatroomPresenter_loadHistoryFailure;

/// 返回socket接收到的消息
/// @param modelArray 消息队列，不为空
- (void)chatroomPresenter_didReceiveChatModels:(NSArray <PLVChatModel *> *)modelArray;

/// 返回socket接收到的教师回答消息
/// @param model 教师回答消息
- (void)chatroomPresenter_didReceiveAnswerChatModel:(PLVChatModel *)model;

/// socket通知已删除某条消息
/// @param msgId 被删除消息ID
- (void)chatroomPresenter_didMessageDeleted:(NSString *)msgId;

/// socket通知所有聊天消息被清空
- (void)chatroomPresenter_didAllMessageDeleted;

@end
```

2.1.3 获取聊天记录

`PLVChatroomPresenter` 提供了加载历史聊天记录，加载历史聊天记录方法定义如下：

```
/// 加载历史聊天记录  
- (void)loadHistory;
```

每次加载的聊天消息条数为初始化时传入的参数 `count` 的值，或者默认值 20（使用 `-init` 方法初始化时）。每次调用 `Http` 接口成功之后，会记录当前成功加载的页数，下次调用方法 `-loadHistory` 会自动加载下一页数据。

加载聊天记录成功或失败都会通过 2.1.2 所提到的 `delegate` 方法通知 `Scene` 层。

注意，初始化 `PLVChatroomPresenter` 实例对象时，即自动获取了第一页的聊天记录，并触发获取历史聊天记录相关回调。

2.1.4 发送消息

`PLVChatroomPresenter` 类提供发送各类消息的API，并把消息封装成数据模型返回，具体接口定义如下：

```
/// 发送私聊提问消息  
/// @param content 消息文本  
/// @return 消息数据模型  
- (PLVChatModel * _Nullable)sendQuestionMessage:(NSString *)content;  
  
/// 发送文本消息  
/// @param content 消息文本  
/// @return 消息数据模型  
- (PLVChatModel * _Nullable)sendSpeakMessage:(NSString *)content;  
  
/// 发送图片消息  
/// @param image 图片  
/// @return 消息数据模型  
- (PLVChatModel * _Nullable)sendImageMessage:(UIImage *)image;  
  
/// 发送自定义消息  
/// @param event 自定义消息event字段  
/// @param data 自定义消息data字段  
/// @param tip 自定义消息tip字段  
/// @param emitMode 自定义消息emitMode字段  
/// @return 是否成功发送的布尔值  
- (BOOL)sendCustomMessageWithEvent:(NSString *)event  
                                data:(NSDictionary *)data  
                                tip:(NSString * _Nullable)tip  
                                emitMode:(int)emitMode;  
  
/// 发送点赞消息  
- (void)sendLike;
```

发送消息成功时接口返回消息数据模型，失败时返回 `nil`。

2.1.5 禁言和踢人

在讲师端，还支持踢人与禁言的操作，在初始化 `PLVChatroomPresenter` 的实例对象时，可设置属性 `specialRole` 为 `YES`，表示当前登录用户是特殊身份（讲师），此时登录用户将不受聊天室是否关闭、禁言的影响，且可以发送对其他在线成员进行禁言、踢人操作。

```
self.presenter.specialRole = YES;
```

Scene 层可直接调用 SDK 的 `PLVChatroomManager` 的以下方法发送禁言/取消禁言、踢人消息：

```
/// 发送禁言消息，讲师端专用接口
/// @param banned YES-禁言 NO-取消禁言
/// @param userId 被禁言/取消禁言的用户ID
- (BOOL)sendBanMessage:(BOOL)banned bannedUserId:(NSString *)userId;

/// 发送踢人消息，讲师端专用接口
/// @param userId 被踢出的用户ID
- (BOOL)sendKickMessageWithUserId:(NSString *)userId;
```

2.1.5 销毁

Scene 层在退出当前场景时，对 `presenter` 实例进行销毁操作。销毁方法定义如下：

```
/// 销毁方法
/// 退出前调用，用于资源释放、状态位清零
- (void)destroy;
```