

- [1 功能概述](#)
- [2 使用演示](#)
- [3 接口介绍](#)
- [4 实现介绍](#)
 - [4.1 初始化视图](#)
 - [4.2 初始化数据](#)

1 功能概述

聊天室模块包括发言、回复、历史记录等功能。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 使用演示

互动学堂中聊天室是一个弹层布局，调用聊天室布局的构造函数、设置回调监听，并调用其初始化方法以完成初始化。在需要打开聊天室的时候，调用其 `show()` 方法显示即可。

```
// 聊天室布局构造、设置回调监听
private void initChatroomLayout() {
    chatroomLayout = new PLVHCChatroomLayout(getContext());
    chatroomLayout.setOnViewActionListener(new IPLVHCChatroomLayout.OnViewActionListener() {
        // 聊天室布局的回调监听 ...
    });
}

// 初始化数据
chatroomLayout.init(liveRoomDataManager);

// 需要显示的时候，调用show
chatroomLayout.show(getWidth(), smallScreenHeight, smallScreenLocation);
```

详细的调用代码以及和其他模块的通信，请参考 `PLVHCToolBarLayout` 类中的代码调用。

3 接口介绍

互动学堂聊天室布局的接口定义为 `IPLVHCChatroomLayout`，接口定义了可供外部直接调用的功能方法，以及需要外部响应的事件监听器。

1. 提供给外部的可直接调用功能方法

```
/**
 * 初始化
 *
 * @param liveRoomDataManager 直播间数据管理器
 */
void init(IPLVLiveRoomDataManager liveRoomDataManager);

/**
 * 设置view交互事件监听器
 *
 * @param listener 监听器
 */
void setOnViewActionListener(OnViewActionListener listener);

/**
 * 获取聊天室presenter
 *
 * @return 聊天室presenter
 */
IPLVChatroomContract.IChatroomPresenter getChatroomPresenter();

/**
 * 课节准备中
 */
void onLessonPreparing(long serverTime, long lessonStartTime);

/**
 * 课节开始
 */
void onLessonStarted();

/**
 * 课节结束
 */
void onLessonEnd(long inClassTime);

/**
 * 处理图片选择结果
 *
 * @param data 数据
 */
void handleImgSelectResult(Intent data);

/**
 * 显示布局
 */
void show(int viewWidth, int viewHeight, int[] viewLocation);

/**
 * 隐藏
 */
void hide();

/**
 * 是否显示状态
```

```

*
* @return true: 显示, false: 不显示
*/
boolean isShown();

/**
* 是否拦截返回事件
*
* @return true: 拦截, false: 不拦截
*/
boolean onBackPressed();

/**
* 销毁, 释放资源
*/
void destroy();

```

2. 需要外部响应的事件监听器

```

/**
* view交互事件监听器
*/
interface OnViewActionListener {
    /**
    * 可见性改变回调
    *
    * @param isVisible true: 显示, false: 隐藏
    */
    void onVisibilityChanged(boolean isVisible);

    /**
    * 未读信息数改变
    *
    * @param currentUnreadCount 未读信息数
    */
    void onUnreadMsgCountChanged(int currentUnreadCount);
}

```

4 实现介绍

`IPLVHCChatroomLayout` 的实现类是 `PLVHCChatroomLayout`，包括聊天消息列表、发送信息输入框、表情布局、点赞布局等子布局。

4.1 初始化视图

初始化视图在聊天室布局构造时执行，包括聊天室子视图的创建、回调监听设置、消息列表初始化，以及表情列表初始化等。

```

private void initView() {
    LayoutInflater.from(getContext()).inflate(R.layout.plvhc_chatroom_layout, this);

    // findViewById ...

    // setOnClickListener ...

    plvhcChatroomCallInputEt.setOnTouchListener(new OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            if (event.getAction() == MotionEvent.ACTION_UP) {
                v.callOnClick();
            }
            return true;
        }
    });
    plvhcChatroomChatMsgRv.setOnTouchListener(new OnTouchListener() {
        @Override
        public boolean onTouch(View v, MotionEvent event) {
            if (event.getAction() == MotionEvent.ACTION_DOWN) {
                closeEmojiSmallLayout();
            }
            return false;
        }
    });

    // 初始化聊天室列表
    PLVMessageRecyclerView.setLayoutManager(plvhcChatroomChatMsgRv).setStackFromEnd(false);
    plvhcChatroomChatMsgRv.addItemDecoration(new
    PLVMessageRecyclerView.SpacesItemDecoration(ConvertUtils.dp2px(12), 0));
    chatMessageAdapter = new PLVHCMessageAdapter();
    chatMessageAdapter.setOnViewActionListener(new PLVHCMessageAdapter.OnViewActionListener() {
        // 聊天消息列表回调监听 ...
    });
    plvhcChatroomChatMsgRv.setAdapter(chatMessageAdapter);
    plvhcChatroomChatMsgRv.addUnreadView(plvhcChatroomMoreMsgTv);
    plvhcChatroomChatMsgRv.addOnUnreadCountChangeListener(new
    PLVMessageRecyclerView.OnUnreadCountChangeListener() {
        @Override
        public void onChange(int currentUnreadCount) {
            plvhcChatroomMoreMsgTv.setText(currentUnreadCount + "条新信息");
        }
    });

    // 初始化下拉加载历史记录控件
    plvhcChatroomSwipeLoadView.setColorSchemeResources(android.R.color.holo_blue_light,
    android.R.color.holo_red_light,
        android.R.color.holo_orange_light, android.R.color.holo_green_light);
    plvhcChatroomSwipeLoadView.setOnRefreshListener(new SwipeRefreshLayout.OnRefreshListener() {
        @Override
        public void onRefresh() {
            chatroomPresenter.requestChatHistory(chatroomPresenter.getViewIndex(chatroomView));
        }
    });
}

```

```

// 初始化表情列表
PLVHCChatroomUtils.initEmojiList(plvhcChatroomEmojiRv, plvhcChatroomCallInputEt);

// 默认不可用状态
disableChatroom();
}

```

4.2 初始化数据

初始化数据包括数据管理器 `IPLVLiveRoomDataManager` 的接收保存、Presenter `IChatroomPresenter` 的注册和初始化、历史聊天数据的请求，以及聊天室 socket 的登录。

```

@Override
public void init(IPLVLiveRoomDataManager liveRoomDataManager) {
    this.liveRoomDataManager = liveRoomDataManager;
    String userType = liveRoomDataManager.getConfig().getUser().getViewerType();
    isTeacherType = PLVSocketUserConstant.USER_TYPE_TEACHER.equals(userType);
    plvhcChatroomCloseRoomIv.setVisibility(isTeacherType ? View.VISIBLE : View.GONE);

    this.chatroomPresenter = new PLVChatroomPresenter(liveRoomDataManager);
    this.chatroomPresenter.registerView(chatroomView);
    this.chatroomPresenter.init();
    //请求一次历史记录
    chatroomPresenter.requestChatHistory(chatroomPresenter.getViewIndex(chatroomView));

    initSocketLoginManager();
}

```

聊天室以 mvp 模式实现，具体交互逻辑在 `chatroomView` 内实现，详情可以参考代码块 **聊天室 - MVP模式的view层实现**。