

- [1 功能概述](#)
- [2 使用演示](#)
- [3 接口介绍](#)
- [4、实现介绍](#)
 - [4.1 PLVLSChatroomLayout](#)
 - [4.1.1 初始化view方法](#)
 - [4.1.2 初始化数据方法](#)

1 功能概述

聊天室模块包括发言、麦克风控制、摄像头控制等功能。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

聊天室在手机开播场景中对应的实现布局为 `PLVLSChatroomLayout`，以下示例了如何创建聊天室布局以及初始化，代码如下：

```
// 聊天室布局
private IPLVLSChatroomLayout plvlsChatroomLy;
// findViewById
plvlsChatroomLy = findViewById(R.id.plvls_chatroom_ly);
// 初始化聊天室布局
plvlsChatroomLy.init(liveRoomDataManager);
```

上述方法的使用用例可以在polyv demo项目中的 `PLVLSLiveStreamerActivity` 找到。

3 接口介绍

手机开播场景下，针对聊天室布局进行封装的接口。定义了：

- 1、外部直接调用的方法
- 2、需要外部响应的事件监听器

```
public interface IPLVLSChatroomLayout {

    // <editor-fold defaultstate="collapsed" desc="1、外部直接调用的方法">
    /**
     * 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);

    /**
     * 添加聊天室在线人数变化监听器
     *
     * @param listener 监听器
     */
    void addOnOnlineCountListener(IPLVOnDataChangedListener<Integer> listener);

    /**
     * 设置麦克风开关按钮状态
     *
     * @param isOpen true: 打开, false: 关闭
     */
    void setOpenMicViewStatus(boolean isOpen);

    /**
     * 设置摄像头开关按钮状态
     *
     * @param isOpen true: 打开, false: 关闭
     */
    void setOpenCameraViewStatus(boolean isOpen);

    /**
     * 设置前置摄像头按钮状态
     *
     * @param isFront true: 前置, false: 后置
     */
    void setFrontCameraViewStatus(boolean isFront);

    /**
     * 改变聊天室布局可见性
     *
     * @param visibility 可见性
     */
    void setVisibility(int visibility);

    /**
     * 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();
    // </editor-fold>

    // <editor-fold defaultstate="collapsed" desc="2、需要外部响应的事件监听器">
    /**
     * view交互事件监听器
     */
    interface OnViewActionListener {
        /**
         * 麦克风控制
         *
         * @param isMute true: 禁用, false: 启用
         */
        boolean onMicControl(boolean isMute);

        /**
         * 摄像头控制
         *
         * @param isMute true: 禁用, false: 启用
         */
        boolean onCameraControl(boolean isMute);

        /**
         * 前置摄像头控制
         *
         * @param isFront true: 前置, false: 后置
         */
        boolean onFrontCameraControl(boolean isFront);
    }
    // </editor-fold>
}

```

4、实现介绍

4.1 PLVLSChatroomLayout

该类是手机开播场景下的聊天室布局, 实现 `IPLVLSChatroomLayout` 接口。

该布局包含聊天信息列表、发送信息输入框、聊天室工具栏布局等相关的UI。

下面会列举介绍该布局中涉及到的主要的方法。

4.1.1 初始化view方法

`PLVLSChatroomLayout` 继承于 `FrameLayout`, 在构造器中使用 `initView` 方法对 `view` 进行了初始化处理。

```

public PLVLSChatroomLayout(@NonNull Context context, @Nullable AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    initView();
}

private void initView() {
    //填充布局到该view中
    LayoutInflater.from(getContext()).inflate(R.layout.plvls_chatroom_layout, this);

    //findView
    plvlsChatroomChatMsgLy = findViewById(R.id.plvls_chatroom_chat_msg_ly);
    plvlsChatroomSwipeLoadView = findViewById(R.id.plvls_chatroom_swipe_load_view);
    plvlsChatroomChatMsgRv = findViewById(R.id.plvls_chatroom_chat_msg_rv);
    plvlsChatroomUnreadMsgTv = findViewById(R.id.plvls_chatroom_unread_msg_tv);
    plvlsChatroomControlIv = findViewById(R.id.plvls_chatroom_control_iv);
    plvlsChatroomToolbarMicControlIv = findViewById(R.id.plvls_chatroom_toolbar_mic_control_iv);
    plvlsChatroomToolbarCameraControlIv = findViewById(R.id.plvls_chatroom_toolbar_camera_control_iv);
    plvlsChatroomToolbarFrontCameraControlIv =
    findViewById(R.id.plvls_chatroom_toolbar_front_camera_control_iv);
    plvlsChatroomToolbarOpenInputWindowTv = findViewById(R.id.plvls_chatroom_toolbar_open_input_window_tv);

    //初始化按钮点击事件
    plvlsChatroomControlIv.setOnClickListener(this);
    plvlsChatroomToolbarMicControlIv.setOnClickListener(this);
    plvlsChatroomToolbarCameraControlIv.setOnClickListener(this);
    plvlsChatroomToolbarFrontCameraControlIv.setOnClickListener(this);
    plvlsChatroomToolbarOpenInputWindowTv.setOnClickListener(this);

    //初始化控件...
}

```

4.1.2 初始化数据方法

PLVLSChatroomLayout 的 init 方法是对外API，需要外部传入 IPLVLiveRoomDataManager 后进行初始化。

```

//聊天室presenter
private IPLVChatroomContract.IChatroomPresenter chatroomPresenter;

@Override
public void init(IPLVLiveRoomDataManager liveRoomDataManager) {
    this.liveRoomDataManager = liveRoomDataManager;
    //创建聊天室mvp-presenter
    this.chatroomPresenter = new PLVChatroomPresenter(liveRoomDataManager);
    //注册聊天室mvp-view
    this.chatroomPresenter.registerView(chatroomView);
    //初始化聊天室mvp-presenter
    this.chatroomPresenter.init();
    //请求一次历史记录
    chatroomPresenter.requestChatHistory(chatroomPresenter.getViewIndex(chatroomView));
    //初始化socket登录管理器
    initSocketLoginManager();
}

//创建聊天室的mvp-view
private PLVAbsChatroomView chatroomView = new PLVAbsChatroomView() {
    //...
}

```

这里的 `PLVChatroomPresenter` 是使用 `mvp` 模式封装的聊天室 `presenter`，用 `presenter` 将 `view` 和 `mode` 隔离开来，一切业务逻辑都是通过 `presenter` 来进行操作，也就是说 `presenter` 是视图的数据的桥梁，视图和数据相隔两端。