

- [1 功能概述](#)
- [2 使用演示](#)
- [3 实现介绍](#)
 - [3.1 PLVLinkMicPresenter](#)
 - [3.2 PLVLCLinkMicControlBar](#)
 - [3.2.1 初始化方法](#)
 - [3.2.2 PLVLCLinkMicControlBarDelegate](#)
 - [3.3 PLVLCLinkMicWindowsView](#)

1 功能概述

保利威云课堂sdk支持连麦功能，举手连麦、音频连麦、视频连麦可自由选择，并支持多人连麦，丰富课堂内容。

连麦流程：讲师端发起连麦->学员举手申请连麦->讲师端允许学员连麦->学员正式加入连麦。

2 使用演示

在demo项目中 PLVLCCloudClassViewController 演示了如何创建连麦布局，以及如何调用初始化方法。

```
self.linkMicAreaView = [[PLVLCLinkMicAreaView alloc] init];
self.linkMicAreaView.delegate = self;
```

其中 PLVLCLinkMicAreaView 负责管理 PLVLinkMicPresenter 连麦管理器， PLVLCLinkMicControlBar 连麦悬浮控制栏以及 PLVLCLinkMicWindowsView 连麦窗口列表视图。因此可 PLVLCLinkMicAreaView 作为连麦模块集成使用。

同时 PLVLCCloudClassViewController 需遵循协议 PLVLCLinkMicAreaViewDelegate 。该协议提供第一画面切换回调、连麦状态变更回调等相关的回调，用于视图的更新和切换。该协议提供以下代理方法：

```

@protocol PLVLCLinkMicAreaViewDelegate <NSObject>

/// RTC画面窗口 需外部展示 ‘第一画面连麦窗口’
///
/// @param linkMicAreaView 连麦区域视图
/// @param canvasView 第一画面连麦窗口视图（需外部进行添加展示）
- (void)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView showFirstSiteCanvasViewOnExternal:(UIView
*)canvasView;

/// RTC画面窗口被点击（表示用户希望视图位置交换）
///
/// @note 当接收到此回调时，表示用户希望 某个Rtc画布视图(即canvasView) 在外部位置中显示，可对 canvasView 作相应布局处理
///
/// @param linkMicAreaView 连麦区域视图
/// @param canvasView Rtc画布视图
- (UIView *)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView rtcWindowDidClickedCanvasView:(UIView
*)canvasView;

/// 恢复外部视图位置
///
/// @note 当外部视图需要回归原位时，此回调将被触发；接收到此回调后，可将 externalView 重新布局在原位上
///
/// @param linkMicAreaView 连麦区域视图
/// @param externalView 被添加在连麦区域视图上的外部视图
- (void)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView rollbackExternalView:(UIView
*)externalView;

/// ‘是否在RTC房间中’ 状态值发生改变
///
/// @param linkMicAreaView 连麦区域视图
/// @param inRTCRoom 当前 是否在RTC房间中
- (void)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView inRTCRoomChanged:(BOOL)inRTCRoom;

/// ‘RTC房间在线用户数’ 发生改变
///
/// @param linkMicAreaView 连麦区域视图
/// @param currentRTCRoomUserCount 当前 RTC房间在线用户数
- (void)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView currentRTCRoomUserCountChanged:
(NSInteger)currentRTCRoomUserCount;

/// ‘是否正在连麦’ 状态值改变
///
/// @param linkMicAreaView 连麦区域视图
/// @param inLinkMic 当前是否正在连麦（YES:正在连麦中 NO:不在连麦中）
- (void)plvLCLinkMicAreaView:(PLVLCLinkMicAreaView *)linkMicAreaView inLinkMicChanged:(BOOL)inLinkMic;

/// 需获知 ‘当前频道是否直播中’
///
/// @note 此回调不保证在主线程触发
///
/// @param linkMicAreaView 连麦区域视图
- (BOOL)plvLCLinkMicAreaViewGetChannelInLive:(PLVLCLinkMicAreaView *)linkMicAreaView;

/// 需获知 ‘主讲的PPT 当前是否在主屏’
///

```

```
/// @note 此回调不保证在主线程触发
///
/// @param linkMicAreaView 连麦区域视图
- (BOOL)plvLCLinkMicAreaViewGetMainSpeakerPPTOnMain:(PLVLinkMicAreaView *)linkMicAreaView;

@end
```

并在 `-viewWillLayoutSubviews` 方法中，对 `PLVLinkMicAreaView` 进行布局，具体可参考demo项目中 `PLVLinkMicAreaView` 的实现。

3 实现介绍

3.1 PLVLinkMicPresenter

`PLVLinkMicPresenter` 连麦管理器在demo项目中已由 `PLVLinkMicAreaView` 进行管理，可查看以下说明，了解如何自行使用 `PLVLinkMicPresenter`。

连麦功能模块使用了mvp模式，因此在实现时需要先初始化Presenter。

```
self.presenter = [[PLVLinkMicPresenter alloc] init];

self.presenter.delegate = self;

self.presenter.preRenderContainer = self;
```

同时需要配置默认值，以连麦悬浮控制栏中声明的为准，详见demo项目中 `PLVLinkMicControlBarProtocol.h`。若业务改变时，可直接修改 `PLVLinkMicControlBarProtocol.h` 的默认值，则连麦管理器 中同时生效。

```
self.presenter.micDefaultOpen = PLVLinkMicControlBarMicDefaultOpen;

self.presenter.cameraDefaultOpen = PLVLinkMicControlBarCameraDefaultOpen;

self.presenter.cameraDefaultFront = PLVLinkMicControlBarSwitchCameraDefaultFront;
```

同时使用 `PLVLinkMicPresenter` 可遵循协议 `PLVLinkMicPresenterDelegate`，并按需求实现其中代理方法。更多内容可以参考demo项目中 `PLVLinkMicAreaView` 的演示。

3.2 PLVLinkMicControlBar

`PLVLinkMicControlBar` 连麦悬浮控制栏在demo项目中已由 `PLVLinkMicAreaView` 进行管理，可查看以下说明，了解如何自行使用 `PLVLinkMicControlBar` 对其进行轻度修改或模仿重写实现业务场景。

PLVLCLinkMicControlBar 在 PLVLCLinkMicAreaView 中初始化，由连麦悬浮控制条的点击事件以及刷新连麦控制条的UI状态。主要的逻辑集中在动画、点击事件响应、和横竖屏两套UI保持同步。

3.2.1 初始化方法

PLVLCLinkMicLandscapeControlBar 和 PLVLCLinkMicPortraitControlBar 分别是连麦悬浮控制条 PLVLCLinkMicControlBar 的竖屏和横屏的实现。

```
_portraitControlBar = [[PLVLCLinkMicPortraitControlBar alloc] init];
_portraitControlBar.delegate = self;
_landscapeControlBar = [[PLVLCLinkMicLandscapeControlBar alloc] init];
_landscapeControlBar.delegate = self;
```

PLVLCLinkMicControlBarProtocol 是连麦控制栏对象抽象协议，在 PLVLCLinkMicAreaView 中用于控制当前连麦悬浮控制栏currentControlBar。实现[setCurrentControlBar]可以参考demo项目中 PLVLCLinkMicAreaView 的演示。

```
self.currentControlBar = self.landscapeControlBar; //设置当前悬浮控制条为竖屏样式
```

在 PLVLCCloudClassViewController 中合适时机中需要插入currentControlBar，具体实现可以参考demo项目中 PLVLCCloudClassViewController 的演示。

3.2.2 PLVLCLinkMicControlBarDelegate

PLVLCLinkMicAreaView 需遵循协议 PLVLCLinkMicControlBarDelegate，并实现其中代理方法。以连麦开关按钮被点击为例，通过不同的状态调用PLVLinkMicPresenter的申请连麦、取消申请连麦、退出连麦：

```

- (void)plvLCLinkMicControlBar:(id<PLVLCLinkMicControlBarProtocol>)bar onOffButtonClickedCurrentStatus:
(PLVLCLinkMicControlBarStatus)status{
    __weak typeof(self) weakSelf = self;
    if (status == PLVLCLinkMicControlBarStatus_Open) {
        // Bar 处于显示 ‘申请连麦’, 点击表示希望申请连麦
        [self.presenter requestJoinLinkMic];
    }else if (status == PLVLCLinkMicControlBarStatus_Waiting){
        // Bar 处于显示 ‘请求中...’, 点击表示希望取消申请连麦
        [PLVFDUtil showAlertWithTitle:@"确认取消申请连麦吗? " message:nil viewController:[PLVFDUtil
getCurrentViewController] cancelButtonTitle:@"按错了" cancelActionStyle:UIAlertActionStyleDefault
cancelActionBlock:nil confirmActionTitle:@"取消申请连麦" confirmActionStyle:UIAlertActionStyleDestructive
confirmActionBlock:^(UIAlertAction * _Nonnull action) {
            [weakSelf.presenter cancelRequestJoinLinkMic];
        }];
    }else if (status == PLVLCLinkMicControlBarStatus_Joined){
        // Bar 处于已连麦, 点击表示希望取消申请连麦
        [PLVFDUtil showAlertWithTitle:@"确认挂断连麦吗? " message:nil viewController:[PLVFDUtil
getCurrentViewController] cancelButtonTitle:@"按错了" cancelActionStyle:UIAlertActionStyleDefault
cancelActionBlock:nil confirmActionTitle:@"挂断" confirmActionStyle:UIAlertActionStyleDestructive
confirmActionBlock:^(UIAlertAction * _Nonnull action) {
            [weakSelf.presenter leaveLinkMic];
        }];
    }
}
}

```

更多内容可以参考demo项目中 `PLVLCLinkMicAreaView` 的演示。

3.3 PLVLCLinkMicWindowsView

`PLVLCLinkMicWindowsView` 是连麦功能模块的主体，负责展示多个连麦成员RTC画面窗口，该视图支持左右滑动浏览。初始化方法参考如下：

```

_windowView = [[PLVLCLinkMicWindowsView alloc] init];
_windowView.delegate = self;

```

同时 `PLVLCLinkMicAreaView` 需遵循协议 `PLVLCLinkMicWindowsViewDelegate`，并实现其中代理方法，与 `PLVLinkMicPresenter` 连麦管理器实现数据交换。示例如下：

```

#pragma mark PLVLCLinkMicWindowsViewDelegate
/// 连麦窗口列表视图 需获知 ‘当前频道连麦场景类型’
- (PLVChannelLinkMicSceneType)plvLCLinkMicWindowsViewGetCurrentLinkMicSceneType:(PLVLCLinkMicWindowsView *)windowsView{
    return self.presenter.linkMicSceneType;//获取连麦管理当前频道连麦场景类型
}
/// 连麦窗口被点击事件（表示用户希望某个窗口成为‘第一画面’）
- (void)plvLCLinkMicWindowsView:(PLVLCLinkMicWindowsView *)windowsView linkMicUserWantToBecomeFirstSite:(NSInteger)index{
    [self.presenter changeMainSpeakerInLocalWithLinkMicUserIndex:index];//连麦管理器请求变更‘第一画面’
}

/// 连麦窗口列表视图 需要获取当前用户数组
- (NSArray *)plvLCLinkMicWindowsViewGetCurrentUserModelArray:(PLVLCLinkMicWindowsView *)windowsView{
    return self.presenter.onlineUserArray;//获取连麦管理器当前RTC房间在线用户数组
}

```

更多内容可以参考demo项目中 `PLVLCLinkMicAreaView` 的演示。