

- 1 功能概述
- 2 实现介绍
 - 2.1 PLVStreamerPresenter
 - 2.1.1 初始化方法
 - 2.1.2 设置默认配置
 - 2.1.3 推流画面的预览
 - 2.1.4 开始上课方法
 - 2.1.5 结束上课方法
 - 2.1.6 允许用户上麦
 - 2.1.7 挂断用户连麦
 - 2.2 PLVLSLinkMicAreaView

1 功能概述

推流和连麦模块是手机开播场景中的基础功能，支持摄像头开关、摄像头前置后置切换、麦克风开关功能，推流同时支持清晰度设置和摄像头镜像的功能。推流可以用于讲师上课，连麦支持音频连麦、视频连麦可自由选择，并支持多人连麦，丰富课堂内容。

推流流程：初始化推流引擎->设置推流配置参数->开始推流。

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 实现介绍

2.1 PLVStreamerPresenter

该类是手机开播场景下，针对推流和连麦进行封装的推流管理器，支持旁路CDN推流、支持课程管理、支持RTC连麦、连麦用户管理。

2.1.1 初始化方法

在demo项目 `PLVLSStreamerViewController` 中需对管理器实现初始化处理：

```
self.streamerPresenter = [[PLVStreamerPresenter alloc] init];
self.streamerPresenter.delegate = self;
self.streamerPresenter.previewType = PLVStreamerPresenterPreviewType_UserArray;
```

同时 `PLVLSStreamerViewController` 需遵循协议 `PLVStreamerPresenterDelegate`，并实现其中代理方法。更多内容可以参考demo项目中 `PLVLSStreamerViewController` 的演示。

2.1.2 设置默认配置

根据具体的业务场景可以进行麦克风、摄像头、流属性、混流布局模式以及其他属性的默认配置：

```
// 设置 麦克风、摄像头默认配置
// 仅在[prepareLocalPreviewCompletion:]方法调用前设置有效
self.streamerPresenter.micDefaultOpen = YES;// YES:麦克风默认开启 NO:麦克风默认关闭; 默认值 NO
self.streamerPresenter.cameraDefaultOpen = YES;// YES:摄像头默认开启 NO:摄像头默认关闭; 默认值 NO
self.streamerPresenter.cameraDefaultFront = YES;// YES:摄像头默认前置 NO:摄像头默认后置; 默认值 后置

// 设置 流属性配置
[self.streamerPresenter setupStreamScale:PLVLinkMicStreamScale16_9];
[self.streamerPresenter setupStreamQuality:PLVLinkMicStreamQuality720P];

// 设置 混流配置
[self.streamerPresenter setupMixLayoutType:PLVRTCStreamerMixLayoutType_MainSpeaker];
```

2.1.3 推流画面的预览

以demo项目中 PLVLSStreamerViewController 演示为例，socket 登录成功回调后加入RTC频道：

```
#pragma mark - PLVSocketManager Protocol

- (void)socketManager_didLoginSuccess:(NSString *)ackString { // 登录成功
    // 登录成功
    [self.streamerPresenter joinRTCChannel];
}
```

此时房间状态发生改变，在下方示例的代理方法中设置本地画面预览的载体视图：

```
#pragma mark - PLVStreamerPresenterDelegate

/// ‘房间加入状态’ 发生改变

- (void)plvStreamerPresenter:(PLVStreamerPresenter *)presenter currentRtcRoomJoinStatus:
(PLVStreamerPresenterRoomJoinStatus)currentRtcRoomJoinStatus inRTCRoomChanged:(BOOL)inRTCRoomChanged inRTCRoom:
(BOOL)inRTCRoom{
    if (inRTCRoomChanged) {
        if (inRTCRoom) {
            [self preapareStartClass];
        }
    }
}

- (void)preapareStartClass {
    __weak typeof(self) weakSelf = self;
    [self.streamerPresenter prepareLocalPreviewCompletion:^(BOOL granted, BOOL prepareSuccess) {
        if (prepareSuccess) {
            [weakSelf.streamerPresenter setupLocalPreviewWithCanvaView:nil];
        }
        if (!granted) {
            NSString *msg = [NSString stringWithFormat:@"需要获取您的音视频权限，请前往设置"];
            [PLVAuthorizationManager showAlertWithTitle:@"提示" message:msg viewController:weakSelf];
        }
    }];
}
```

2.1.4 开始上课方法

PLVStreamerPresenter 的 startClass 方法是用于开始上课的方法，内部调用 startPushStream 方法。startPushStream 方法仅负责推流，而 startClass 方法将开始推流，且发送请求修改频道状态为“开始上课”。建议业务场景使用 startClass 方法。

在demo项目中 PLVLSStreamerViewController 演示了倒计时结束时开始上课:

```

    __weak typeof(self) weakSelf = self;
    _coutBackView.countDownCompletedBlock = ^{
        [weakSelf.streamerPresenter startClass];
    };

```

2.1.5 结束上课方法

PLVStreamerPresenter 的 finishClass 方法是用于结束上课的方法，内部调用 stopPushStream 方法。startPushStream 方法仅停止推流，而 finishClass 方法停止推流，且发送请求修改频道状态为“结束上课”。建议业务场景使用 finishClass 方法：

```
[self.streamerPresenter finishClass];
```

2.1.6 允许用户上麦

`PLVStreamerPresenter` 的 `allowRemoteUserJoinLinkMic` 方法是用于允许用户上麦的方法：

```
/// 允许 某位远端用户 上麦
///
/// @param waitUser 远端用户等待连麦模型
/// @param emitCompleteBlock ‘允许 某位远端用户 上麦’的请求发送结果Block
- (void)allowRemoteUserJoinLinkMic:(PLVLinkMicWaitUser *)waitUser emitCompleteBlock:(nullable void (^)(BOOL emitSuccess))emitCompleteBlock;
```

在 `PLVStreamerPresenter` 的 `addLinkMicWaitUserWithDict` 内部方法中实现对用户上麦。

当RTC房间在线用户数组‘发生改变时，会调用 `PLVLSLinkMicAreaView` 的 `reloadLinkMicUserWindows` 方法刷新连麦用户的画面：

```
/// ‘RTC房间在线用户数组’ 发生改变
- (void)plvStreamerPresenter:(PLVStreamerPresenter *)presenter linkMicOnlineUserListRefresh:(NSArray<PLVLinkMicOnlineUser *>*)onlineUserArray{
    [self.linkMicAreaView reloadLinkMicUserWindows];
}
```

2.1.7 挂断用户连麦

`PLVStreamerPresenter` 的 `closeAllLinkMicUser` 方法是用于挂断全部连麦用户的方法：

```
[self.streamerPresenter closeAllLinkMicUser];
```

`PLVStreamerPresenter` 的 `closeRemoteUserLinkMic` 方法是用于挂断某位远端用户的连麦的方法，在 `PLVStreamerPresenter` 内部使用，`closeAllLinkMicUser` 方法应用 `closeRemoteUserLinkMic` 方法实现挂断全部连麦用户的方法。

更多实现可以参考demo项目中 `PLVLSStreamerViewController`。

2.2 PLVLSLinkMicAreaView

`PLVLSLinkMicAreaView` 用于承载 `PLVStreamerPresenter` 推流管理器提供的数据。

在demo项目中 `PLVLSStreamerViewController` 演示了如何添加 `PLVLSLinkMicAreaView` 控件：

```
self.linkMicAreaView = [[PLVLSLinkMicAreaView alloc] init];
self.linkMicAreaView.delegate = self;
[self.view addSubview:self.linkMicAreaView];
```

同时 `PLVLSStreamerViewController` 需遵循协议 `PLVLSLinkMicAreaViewDelegate`，并实现以下代理方法：

```
@protocol PLVLSLinkMicAreaViewDelegate <NSObject>

/// 连麦窗口列表视图 需要获取当前用户数组
///
/// @param linkMicAreaView 连麦窗口列表视图
- (NSArray *)plvLSLinkMicWindowsViewGetCurrentUserModelArray:(PLVLSLinkMicAreaView *)linkMicAreaView;

/// 连麦窗口列表视图 需要查询某个条件用户的下标值
///
/// @param linkMicAreaView 连麦窗口列表视图
/// @param filtrateBlockBlock 筛选条件Block
- (NSInteger)plvLSLinkMicWindowsView:(PLVLSLinkMicAreaView *)linkMicAreaView
findUserModelIndexWithFiltrateBlock:(BOOL(^)(PLVLinkMicOnlineUser * enumerateUser))filtrateBlockBlock;

/// 连麦窗口列表视图 需要根据下标值获取对应用户
///
/// @param linkMicAreaView 连麦窗口列表视图
/// @param targetIndex 目标下标值
- (PLVLinkMicOnlineUser *)plvLSLinkMicWindowsView:(PLVLSLinkMicAreaView *)linkMicAreaView
getUserModelFromOnlineUserArrayWithIndex:(NSInteger)targetIndex;

@end
```

并在 `-viewWillLayoutSubviews` 方法中，对 `PLVLSLinkMicAreaView` 进行布局：

```
self.linkMicAreaView.frame = CGRectMake(CGRectGetMaxX(self.documentAreaView.frame) + linkMicAreaViewLeftPadding,
CGRectGetMaxY(self.statusAreaView.frame), linkMicAreaViewWidth, documentViewHeight);
```

上述内容的具体实现可以在demo项目中 `PLVLSStreamerViewController` 找到。