

- [1 功能概述](#)
- [2 使用演示](#)
- [3 实现介绍](#)
 - [3.1 PLVLSDocumentLayout](#)
 - [3.2 PLVLSDocumentControllerLayout](#)

1 功能概述

文档（PPT&白板）模块包括白板、PPT展示、画笔标注等功能。文档模块的整个界面被封装在 `PLVLSDocumentLayout` 中。

2 使用演示

使用文档模块只需要在Activity的布局中添加进 `PLVLSDocumentLayout` 布局，并调用其初始化方法即可。

```
// PLVLSLiveStreamerActivity.java
private void initView() {
    // 文档布局
    plvlsDocumentLy = (IPLVLSDocumentLayout) findViewById(R.id.plvls_document_ly);
    // 初始化文档布局，传入初始化所需的参数
    plvlsDocumentLy.init(liveRoomDataManager);
}
```

同时，也可以设置布局的监听回调，以便在不同模块布局之间进行通信。

```
// IPLVLSDocumentLayout 设置监听器方法定义
/**
 * 设置标注工具栏 折起/展开 回调
 */
void setMarkToolOnFoldExpandListener(PLVLSDocumentControllerExpandMenu.OnFoldExpandListener
onFoldExpandListener);

/**
 * 设置文档布局区域 全屏/正常 回调
 */
void setOnSwitchFullScreenListener(PLVLSDocumentControllerLayout.OnSwitchFullScreenListener
onSwitchFullScreenListener);
```

3 实现介绍

3.1 PLVLSDocumentLayout

文档布局是一个ViewGroup，它主要由一个负责显示白板PPT的WebView - `PLVSDocumentWebView`，和一个叠加在WebView上面负责控制标注画笔等功能的控制栏 - `PLVLSDocumentControllerLayout` 构成。

```
@Override
public void init(IPLVLiveRoomDataManager liveRoomDataManager) {
    // 初始化文档WebView
    initDocumentWebView();
    // 初始化 MVP - Presenter
    initPresenter();
    // 初始化文档控制栏
    initDocumentController();
}
```

3.2 PLVLSDocumentControllerLayout

控制栏是一个皮肤布局，包括标注工具、切换文档全屏、添加新的白板、切换文档上下页功能。它不直接操纵白板PPT的WebView，而是通过调用Presenter提供的相关方法，由Presenter再对文档WebView状态进行修改。下面代码展示当使用者点击箭头标注工具，是如何告诉WebView切换到箭头标注状态的。

```

// PLVLSDocumentControllerLayout.java
// 箭头工具点击监听
plvlsDocumentMarkToolArrowIv.setOnClickListener(new OnClickListener() {
    @Override
    public void onClick(View v) {
        keepControllerVisible();
        setColorSelectorVisibility(VISIBLE);
        changeMarkToolSelectedType(v);
        // 通过回调让PLVLSDocumentLayout来执行切换
        callbackMarkToolType(PLVMarkToolType.ARROW);
        plvlsDocumentMarkMenu.requestLayout();
    }
});

// PLVLSDocumentLayout.java
// 文档布局中设置，控制栏选择标注工具时回调监听
plvlsDocumentControllerLayout.setOnChangeMarkToolListener(new
PLVLSDocumentControllerLayout.OnChangeMarkToolListener() {
    @Override
    public void onChangeMarkTool(@PLVMarkToolType.Range final String markToolType) {
        if (PLVMarkToolType.CLEAR.equals(markToolType)) {
            // 清屏时的处理逻辑
        } else {
            // 其他标注工具的处理逻辑，让Presenter通知WebView变更标注工具
            documentPresenter.changeMarkToolType(markToolType);
        }
    }
});

```