

- [1 功能概述](#)
- [2 使用演示](#)
- [3 实现介绍](#)
 - [3.1 文档展示部分](#)
 - [3.2 文档上传删除部分](#)

1 功能概述

文档管理模块包括选择PPT文档、上传&删除PPT文档、查看PPT每页详情等功能。文档管理是一个弹层页面，整个界面被封装在 `PLVLSPPtListLayout` 中。

2 使用演示

文档管理是一个弹层页面，跟一般直接写进xml的布局不同，使用弹层页面只需要创建弹层对象，并调用 `open()` 方法即可显示弹层。

```
// PLVLSStatusBarLayout.java
private void initView() {
    // 创建了文档管理弹层页面
    pptListLayout = new PLVLSPPtListLayout(getContext());
}

private void processSelectDocument() {
    if (pptListLayout != null) {
        // 点击状态栏上面的文档图标时，调用open打开文档管理弹层
        pptListLayout.open();
    }
}
```

3 实现介绍

3.1 文档展示部分

文档展示包括两种列表的展示，分别是显示PPT封面的所有文档列表 `PLVLSPPtViewType.COVER`，显示PPT每页详情的文档详情列表 `PLVLSPPtViewType.PAGE`。这两种列表使用同一个RecyclerView和Adapter展示，通过给Adapter设置不同的viewType来区分。

下面代码展示如何获得所有PPT文档列表的数据，并设置给Adapter进行列表显示。

```

// PLVLSptListLayout.java
public void open() {
    // 打开弹层时，初始化弹层视图
    initViewByShowType();
}
private void initViewByShowType() {
    // 请求更新数据
    requestUpdateData();
}

/**
 * 根据显示样式，向Presenter请求更新列表数据
 */
private void requestUpdateData() {
    if (showViewType == PLVLSptViewType.COVER) {
        // 向Presenter请求更新PPT文档列表数据
        PLVDocumentPresenter.getInstance().requestGetPptCoverList();
    } else if (showViewType == PLVLSptViewType.PAGE) {
        PLVDocumentPresenter.getInstance().requestGetPptPageList(currentAutoId);
    }
}
}

```

在异步请求获取到文档列表数据后，Presenter会将文档列表数据回调给视图。

```

// PLVLSptListLayout.java
// MVP - View 回调实现
mvpView = new PLVAbsDocumentView() {
    @Override
    public void onPptCoverList(@Nullable PLVSPPTInfo pptInfo) {
        processPptCoverList(pptInfo);
    }
}
}

```

在方法 `processPptCoverList()` 中，会对文档数据进行解析，并转换为视图对象 `PLVLSptV0`，将视图对象列表传给Adapter即可展示文档列表。

3.2 文档上传删除部分

文档删除通过调用Presenter的 `deleteDocument()` 方法实现。

```

// PLVLSptListLayout.java
// 设置列表项长按点击监听
pptListAdapter.setOnPptItemLongClickListener(new PLVLSptListViewHolder.OnPptItemLongClickListener() {
    @Override
    public void onLongClick(View view, final int id, final String fileId) {
        if (pptListAdapter.getRealViewType() != PLVLSptViewType.COVER) {
            return;
        }
        if (documentDeleteArrow != null) {
            // 设置删除确认箭头的点击回调
            documentDeleteArrow.setOnClickListener(new OnClickListener() {
                @Override
                public void onClick(View v) {
                    // 点击删除箭头后 弹窗提示是否删除
                    documentDeleteConfirmDialog
                        .setRightBtnListener(new OnClickListener() {
                            @Override
                            public void onClick(View v) {
                                // 点击确认后, 调用Presenter删除文档
                                PLVDocumentPresenter.getInstance().deleteDocument(fileId);
                                documentDeleteConfirmDialog.hide();
                            }
                        })
                        .show();
                }
            });
            // 显示删除确认箭头
            documentDeleteArrow.showAtLocation(view);
        }
    }
});

```

文档上传部分较为复杂，涉及到9种文档上传状态（见枚举 `PLVPptUploadStatus`）的监听，本地文档上传状态缓存，文档上传失败状态的处理等。具体的文档上传实现细节见 `PLVLSptListLayout` 类的代码块 [初始化方法 - 文档上传状态相关回调设置](#)。