1 功能概述

商品模块包括商品推送布局`PLVECCommodityPushLayout`、商品弹层`PLVECCommodityPopupView`，在直播带货中可以根据直播实时情况，由后台配置商品并推送到观看端上。

2 使用演示

商品模块的实时更新依赖于聊天室的事件监听，因此在聊天室mvp-view的构建中需要实现商品相关的接口

商品控制事件示例代码如下：

```
// PLVECLiveHomeFragment.java
private IPLVChatroomContract.IChatroomView chatroomView = new PLVAbsChatroomView() {
    // 商品控制事件监听
    @Override
    public void onProductControlEvent(@NonNull PLVProductControlEvent productControlEvent) {
        super.onProductControlEvent(productControlEvent);
        acceptProductControlEvent(productControlEvent);
    }

    // 其它接口实现...
}

// 商品控制事件的处理
private void acceptProductControlEvent(final PLVProductControlEvent productControlEvent) {
    if (!isOpenCommodityMenu) {
        return;
    }
    handler.post(new Runnable() {
        @Override
        public void run() {
            final PLVProductContentBean contentBean = productControlEvent.getContent();
            if (productControlEvent.getContent() == null) {
                return;
            }
            if (productControlEvent.isPush()) {//商品推送
                commodityPushLayout.setViewActionListener(new PLVECCommodityPushLayout.ViewActionListener() {
                    @Override
                    public void onEnterClick() {
                        acceptBuyCommodityClick(contentBean);
                    }
                });
                // 更新推送布局的商品内容
                commodityPushLayout.updateView(contentBean.getProductId(), contentBean.getShowId(),
contentBean.getCover(), contentBean.getName(), contentBean.getRealPrice(), contentBean.getPrice());
                // 显示推送布局
                commodityPushLayout.show();
            } else if (productControlEvent.isNewly()) {//新增
                commodityPopupView.add(contentBean, true);
            } else if (productControlEvent.isRedact()) {//编辑
                commodityPopupView.update(contentBean);
            } else if (productControlEvent.isPutOnShelves()) {//上架
                commodityPopupView.add(contentBean, false);
            }
        }
    });
}
```

3 实现介绍

3.1 商品推送布局 - PLVECCommodityPushLayout

商品推送布局在接受到后台的商品控制推送事件时显示，支持显示一件商品，用于带货时推送一件指定商品。

商品推送布局的用例如下：

```
// PLVECLiveHomeFragment.java
// 设置进入商品按钮点击回调
commodityPushLayout.setViewActionListener(new PLVECCommodityPushLayout.ViewActionListener() {
    @Override
    public void onEnterClick() {
        // 在点击进入商品按钮后，跳转到指定的购物页面
        acceptBuyCommodityClick(contentBean);
    }
});
// 更新推送布局显示的商品
commodityPushLayout.updateView(contentBean.getProductId(), contentBean.getShowId(), contentBean.getCover(),
contentBean.getName(), contentBean.getRealPrice(), contentBean.getPrice());
// 显示推送布局，会在5秒后自动隐藏
commodityPushLayout.show();
```

3.2 商品弹层 - PLVECCommodityPopupView

商品弹层在观众点击首页的购物车按钮时显示，使用列表显示多个商品

商品弹层的用例如下：

```
// PLVECLiveHomeFragment.java
private void showCommodityLayout(View v) {
    // 清空旧数据
    commodityPopupView.setCommodityVO(null);
    // 每次弹出都调用一次接口获取商品信息
    liveRoomDataManager.requestProductList();
    // 显示商品弹层
    commodityPopupView.showCommodityLayout(v, new PLVECCommodityAdapter.OnViewActionListener() {
        @Override
        public void onBuyCommodityClick(View view, PLVProductContentBean contentsBean) {
            acceptBuyCommodityClick(contentsBean);
        }

        @Override
        public void onLoadMoreData(int rank) {
            liveRoomDataManager.requestProductList(rank);
        }
    });
}

private void acceptProductControlEvent(final PLVProductControlEvent productControlEvent) {
    // 其它部分代码...
    if (productControlEvent.isPush()) {
        // 商品推送...
    } else if (productControlEvent.isNewly()) {
        // 商品弹层 新增商品
        commodityPopupView.add(contentBean, true);
    } else if (productControlEvent.isRedact()) {
        // 商品弹层 更新编辑商品
        commodityPopupView.update(contentBean);
    } else if (productControlEvent.isPutOnShelves()) {
        // 商品弹层 上架商品
        commodityPopupView.add(contentBean, false);
    }
}
```