前段时间做了一个关于商城的分享介绍卡片式ViewPager，效果看起来还是蛮炫丽的，整体有如下效果特点：

1. 无限轮播
2. 两种卡片布局（中间与两边的不同）
3. 指示灯
4. 滚动到下一个卡片会在Y轴进行偏移
5. 可显示前后卡片的一部分
6. 一开始进入从中间开始
7. 卡片圆角，背景可颜色或图片

说的这么多估计大家还是有点懵吧，那我们就先看下效果图跟项目结构图，再一一跟大家解释其效果实现与文件作用。

实现这个不一样的卡片Viewpager注意的地方:

> 1、对ViewPager设置setPageMargin(int marginPixels)
> 2、布局中引用自定义的ViewPager进行设置clipToPadding="false"，paddingLeft，paddingRight
> 3、对ViewPager设置setPageTransformer()切换时的动画效果

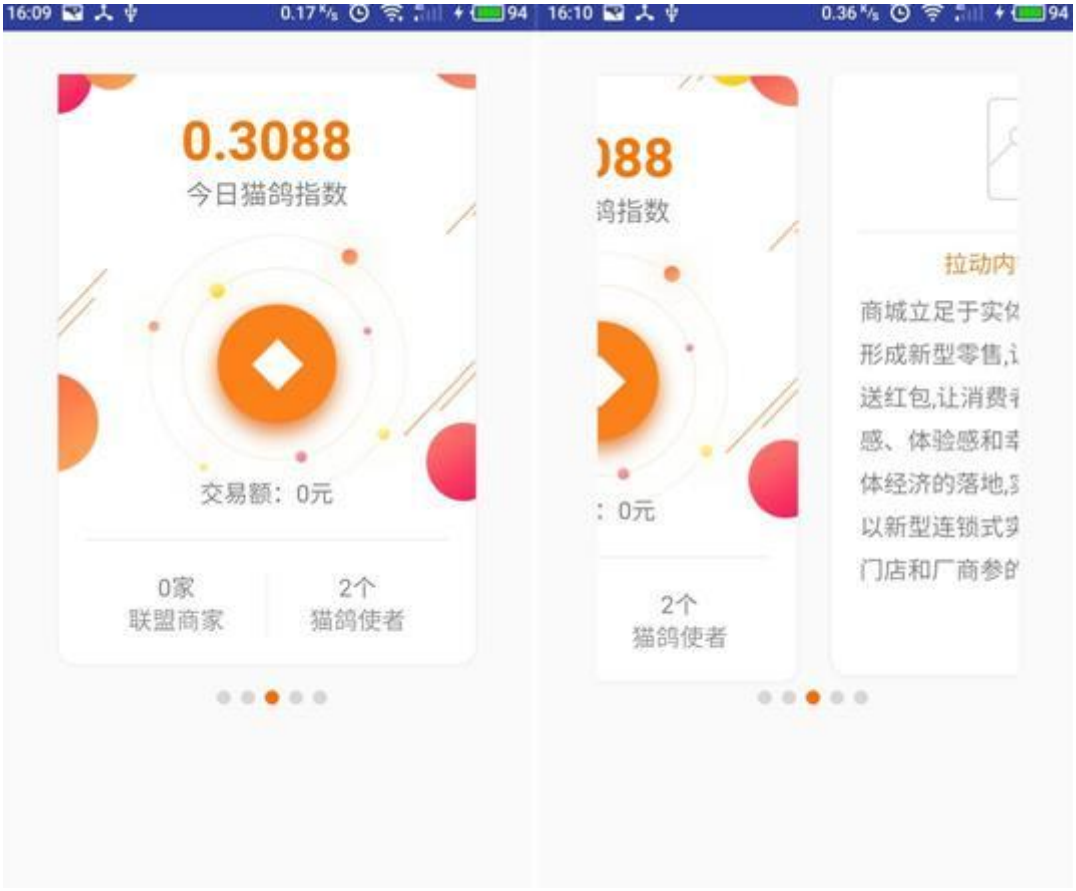
解析以上三个设置特点：

1、setPageMargin：一般ViewPager都是页与页在切换时都是紧贴在一起的。它的设置主要是使页与页有一定的margin。设置之后的效果：

2、clipToPadding: 这个属性一般都是viewgrounp对象才会用到, 他的意思就是对于padding 所占的尺寸大小也绘制其他的item的view。他必须与padding一起使用才会有作用。

clipToPadding=“true”或没有设置时的效果：

看到第一张，是不是明显的左右两张卡片没有显示出预览，再看到第二张才知道原来是左右两边有边距遮盖了，所以只有设置false才可以看到我们要的那种效果。



一、类包文件的介绍:

1、bean->ShareCardItem：是通过用来解析数据的一个模型

2、util->StreamUtils：是用来读取文件res->raw文件的数据。

因为这里的卡片数据我这里就不做数据网络请求，防止项目域名变化，大家就大概理解data.json就是通过网络获取来的数据就行了。而该工具类就是通过get(Context context, int id)读取data.json中的数据json字符串。

```
public class StreamUtils {

    public static String get(Context context, int id) {
        InputStream stream = context.getResources().openRawResource(id);
        return read(stream);
    }

    public static String read(InputStream stream) {
        return read(stream, "utf-8");
    }

    public static String read(InputStream is, String encode) {
        if (is != null) {
            try {
                BufferedReader reader = new BufferedReader(new
InputStreamReader(is, encode));
                StringBuilder sb = new StringBuilder();
                String line = null;
                while ((line = reader.readLine()) != null) {
                    sb.append(line + "\n");
                }
```

```
                is.close();
                return sb.toString();
            } catch (UnsupportedEncodingException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        return "";
    }

    public static String getBase64(byte[] base) {
        String base64 = null;
        try {
            base64 = Base64.encodeToString(base, Base64.NO_WRAP);
        } catch (Exception e) {
            e.printStackTrace();
        }
        return base64;
    }
}12345678910111213141516171819202122232425262728293031323334353637383940 41
```

3、view->LoopPagerAdapterWrapper/LoopViewPager/ViewPager

本文的卡片无限循环滚动就是通过这上个文件实现的，代码量太多了，这里就不做粘贴，可以下载源码查看。

4、view->ScaleTransformer：是ViewPager中的卡片在Y轴上的移动动画。

注释掉的那些是切换卡片缩放跟透明度的变化，这里没做具体计算，如果要用的话，可以自行计算。

```
public class ScaleTransformer implements ViewPager.PageTransformer {
    private static final float MIN_SCALE = 0.7f;//缩放的比例
    private static final float MIN_ALPHA = 0.5f;//透明度的比例
    private static final float MOVE_Y = 40;//设置y轴移动的基数

    @Override
    public void transformPage(View page, float position) {
        if (position < -1 || position > 1) {
//            page.setAlpha(MIN_ALPHA);
//            page.setScaleX(MIN_SCALE);
//            page.setScaleY(MIN_SCALE);
            page.setTranslationY(0);
        } else if (position <= 1) { // [-1,1]
//            float scaleFactor = Math.max(MIN_SCALE, 1 - Math.abs(position));
            if (position <= 0) {
//                float scaleX = 1 + 0.3f * position;
//                page.setScaleX(scaleX);
//                page.setScaleY(scaleX);
                page.setTranslationY(0);
            } else {
                page.setTranslationY(-MOVE_Y * (1 - Math.abs(position)));
//                float scaleX = 1 - 0.3f * position;
//                page.setScaleX(scaleX);
//                page.setScaleY(scaleX);
            }
//            page.setAlpha(MIN_ALPHA + (scaleFactor - MIN_SCALE) / (1 -
MIN_SCALE) * (1 - MIN_ALPHA));
```

```
        }
    }
}
12345678910111213141516171819202122232425262728293O
```

5、view->ShareCardView：重点来了，这就是所说的卡片ViewPager

这里面用到了两种卡片，所以这里只是为了给大家举例，如果要实现其他效果，大致也是一样的。

```
public class ShareCardView extends FrameLayout implements
ViewPager.OnPageChangeListener {
    private static final int MSG_NEXT = 1;
    private Context mContext;
    private ViewPager mViewPager; //自定义的无限循环ViewPager
    private ViewGroup mViewGroup;
    private CardAdapter mAdapter;
    private int mFocusImageId;
    private int mUnfocusImageId;
    private Handler mHandler;
    private TimerTask mTimerTask;
    private Timer mTimer;
    private int centerPos; //中间卡片位置
    private int pageCount; //所有卡片的个数

    public ShareCardView(Context context) {
        this(context, null);
    }

    public ShareCardView(Context context, AttributeSet set) {
        super(context, set);
        init(context);
    }

    private void init(Context context) {
        mContext = context;
        View container =
LayoutInflater.from(mContext).inflate(R.layout.layout_share_cardview, null);
        addView(container);
        mViewPager = (ViewPager) (container.findViewById(R.id.slide_viewPager));
        mViewGroup = (ViewGroup) (container.findViewById(R.id.slide_viewGroup));

        mFocusImageId = R.mipmap.card_point_focused;
        mUnfocusImageId = R.mipmap.card_point_unfocused;

        mViewPager.setPageTransformer(false, new ScaleTransformer());//设置卡片切换
动画
        mViewPager.setPageMargin(60);//卡片与卡片间的距离
        mViewPager.setOnPageChangeListener(this);
        mHandler = new Handler() {
            @Override
            public void handleMessage(Message msg) {
                switch (msg.what) {
                    case MSG_NEXT:
                        next();
                        break;
                }
                super.handleMessage(msg);
```

```
            }
        };

        mViewPager.setOnTouchListener(new OnTouchListener() {
            @Override
            public boolean onTouch(View v, MotionEvent event) {
                switch (event.getAction()) {
                    case MotionEvent.ACTION_DOWN:
                    case MotionEvent.ACTION_MOVE:
                        stopTimer();
                        break;
                    case MotionEvent.ACTION_UP:
                        startTimer();
                        break;
                }
                return false;
            }
        });
    }

    //设置数据
    public void setCardData(ShareCardItem cardItem) {
        pageCount = cardItem.getDataList().size() + 1;
        centerPos = pageCount / 2;//中间卡片的位置
        mAdapter = new CardAdapter(cardItem);
        mViewPager.setAdapter(mAdapter);
        mAdapter.select(centerPos);//默认从中间卡片开始
        mViewPager.setCurrentItem(centerPos);
        mViewPager.setOffscreenPageLimit(pageCount);//预加载所有卡片
        startTimer();
    }

    //启动动画
    public void startTimer() {
        stopTimer();
        mTimerTask = new TimerTask() {
            @Override
            public void run() {
                mHandler.sendEmptyMessage(MSG_NEXT);
            }
        };
        mTimer = new Timer(true);
        mTimer.schedule(mTimerTask, 3000, 3000);
    }

    //停止动画
    public void stopTimer() {
        mHandler.removeMessages(MSG_NEXT);
        if (mTimer != null) {
            mTimer.cancel();
            mTimer = null;
        }

        if (mTimerTask != null) {
            mTimerTask.cancel();
            mTimerTask = null;
        }
    }
```

```
    //选择下一个卡片
    public void next() {
        int pos = mViewPager.getCurrentItem();
        pos += 1;
        mViewPager.setCurrentItem(pos);
    }

    //判断是否显隐控件
    public void refresh() {
        if (getCount() <= 0) {
            this.setVisibility(View.GONE);
        } else {
            this.setVisibility(View.VISIBLE);
        }
    }

    public int getCount() {
        if (mAdapter != null) {
            return mAdapter.getCount();
        }
        return 0;
    }

    public class CardAdapter extends PagerAdapter {

        private LayoutInflater inflater;
        private ArrayList<ImageView> mPoints;
        private ZSCardItem zsCardItem = new ZSCardItem();
        private List<LRCardItem> lrCardItems = new ArrayList<>();
        private List<Integer> iconLists = Arrays.asList(R.mipmap.share_card1,
R.mipmap.share_card2,
                R.mipmap.share_card3, R.mipmap.share_card4);

        public CardAdapter(ShareCardItem cardItem) {
            inflater = LayoutInflater.from(mContext);
            mPoints = new ArrayList<>();
            zsCardItem = cardItem.getData();
            lrCardItems = cardItem.getDataList();
            setItems();
        }

        @Override
        public int getCount() {
            return pageCount;
        }

        private ImageView newPoint() {
            ImageView imageView = new ImageView(mContext);
            LinearLayout.LayoutParams params =
                    new LinearLayout.LayoutParams(LayoutParams.WRAP_CONTENT,
LayoutParams.WRAP_CONTENT);
            params.leftMargin = 20;
            imageView.setLayoutParams(params);
            imageView.setBackgroundResource(mUnfocusImageId);
            return imageView;
        }
```

```
        //中间卡片
        public void setZSCard(View view, ZSCardItem item) {
            TextView zsTv = (TextView) view.findViewById(R.id.zs_tv);
            TextView jyeTv = (TextView) view.findViewById(R.id.jye_tv);
            TextView lmTv = (TextView) view.findViewById(R.id.lm_tv);
            TextView sbTv = (TextView) view.findViewById(R.id.sb_tv);
            zsTv.setText(String.valueOf(item.getTodayIndex()));
            jyeTv.setText(String.format("交易额：%s元", item.getAmount()));
            lmTv.setText(String.format("%s家", item.getShopNum()));
            sbTv.setText(String.format("%s个", item.getMemberNum()));
        }

        //其他卡片
        public void setLRCard(View view, int lrCardItemPos) {
            LRCardItem item = lrCardItems.size() > lrCardItemPos ?
                    lrCardItems.get(lrCardItemPos) : new LRCardItem();
            ImageView iconIv = (ImageView) view.findViewById(R.id.icon_iv);
            TextView titleTv = (TextView) view.findViewById(R.id.title_tv);
            TextView contentTv = (TextView) view.findViewById(R.id.content_tv);
            if (lrCardItemPos < iconLists.size()) {
                iconIv.setImageResource(iconLists.get(lrCardItemPos));
            }
            titleTv.setText(item.getTitle());
            contentTv.setText(item.getContent());
            contentTv.setMovementMethod(ScrollingMovementMethod.getInstance());
        }

        public void setItems() {
            while (mPoints.size() < pageCount) mPoints.add(newPoint());
            while (mPoints.size() > pageCount) mPoints.remove(0);
            mViewGroup.removeAllViews();
            for (ImageView view : mPoints) {
                mViewGroup.addView(view);
            }
            mViewPager.setCurrentItem(0);
            select(0);
        }

        public void select(int pos) {
            if (mPoints.size() > 0) {
                pos = pos % mPoints.size();
                for (int i = 0; i < mPoints.size(); i++) {
                    if (i == pos) {
                        mPoints.get(i).setBackgroundResource(mFocusImageId);
                    } else {
                        mPoints.get(i).setBackgroundResource(mUnfocusImageId);
                    }
                }
            }
            refresh();
        }

        @Override
        public boolean isViewFromObject(View view, Object object) {
            return view == object;
        }

        @Override
```

```
        public Object instantiateItem(ViewGroup container, int position) {
            View view = null;
            if (position == centerPos) {
                view = inflater.inflate(R.layout.layout_share_zscard, container,
false);
                setZSCard(view, zsCardItem);
            } else if (position < centerPos) {
                view = inflater.inflate(R.layout.layout_share_lrcard, container,
false);
                setLRCard(view, position);
            } else {
                view = inflater.inflate(R.layout.layout_share_lrcard, container,
false);
                setLRCard(view, position - 1);
            }
            container.addView(view);
            return view;
        }

        @Override
        public void destroyItem(ViewGroup container, int position, Object object)
{
            container.removeView((View) object);
        }
    }

    @Override
    public void onPageScrolled(int position, float positionOffset, int
positionOffsetPixels) {
    }

    @Override
    public void onPageSelected(int position) {
        mAdapter.select(position);
    }

    @Override
    public void onPageScrollStateChanged(int state) {

    }
}12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012112212312412512612712812913013113213313413513613713813914014114214314414514614714814915015115215315415515615715815916016116216316416516616716817017017117217317417517617717817918018118218318418518618718818919019119219319419519619719819920020120220320420520620720820921021121221321421521621721821922022122223224225226227228229230231232233234235236237238239240241242243244245246247248249250251252253
```

二、布局文件介绍：

1、卡片viewpager+指示灯：layout_share_cardview.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <soban.cardloopviewpager.view.ViewPager
        android:id="@+id/slide_viewPager"
        android:layout_width="match_parent"
        android:layout_height="@dimen/share_card_height"
        android:clipToPadding="false"
        android:paddingBottom="5dp"
        android:paddingLeft="40dp"
        android:paddingRight="40dp"
        android:paddingTop="30dp" />

    <LinearLayout
        android:id="@+id/slide_viewGroup"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/slide_viewPager"
        android:gravity="center_horizontal"
        android:orientation="horizontal" />
</RelativeLayout>
12345678910111213141516171819202122232 4
```

2、中间卡片+圆角+背景图：layout_share_zscard.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/subview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_share_card">

    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_margin="1dip"
        android:background="@mipmap/share_card_zs" />

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:padding="@dimen/share_card_space">

        <TextView
            android:id="@+id/zs_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="10dip"
```

```
            android:text="0"
            android:textColor="@color/theme_color"
            android:textSize="36sp"
            android:textStyle="bold" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/zs_tv"
            android:layout_centerHorizontal="true"
            android:text="今日猫鸽指数"
            android:textSize="18sp" />

        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerInParent="true"
            android:src="@mipmap/share_card_c" />

        <LinearLayout
            android:id="@+id/bottom_layout"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:gravity="center_horizontal"
            android:orientation="vertical">

            <TextView
                android:id="@+id/jye_tv"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_marginBottom="@dimen/share_card_space"
                android:text="交易额：0元"
                android:textSize="@dimen/share_card_text" />

            <View
                android:layout_width="match_parent"
                android:layout_height="1dip"
                android:background="@color/theme_border" />

            <LinearLayout
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:orientation="horizontal"
                android:paddingTop="@dimen/share_card_space">

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:gravity="center"
                    android:orientation="vertical">

                    <TextView
                        android:id="@+id/lm_tv"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="0家"
```

```
                        android:textSize="@dimen/share_card_text" />

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="联盟商家"
                        android:textColor="@color/theme_text"
                        android:textSize="@dimen/share_card_text" />
                </LinearLayout>

                <View
                    android:layout_width="1dip"
                    android:layout_height="match_parent"
                    android:background="@color/theme_border" />

                <LinearLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:layout_weight="1"
                    android:gravity="center"
                    android:orientation="vertical">

                    <TextView
                        android:id="@+id/sb_tv"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="0个"
                        android:textSize="@dimen/share_card_text" />

                    <TextView
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:text="猫鸽使者"
                        android:textColor="@color/theme_text"
                        android:textSize="@dimen/share_card_text" />
                </LinearLayout>
            </LinearLayout>
        </LinearLayout>
    </RelativeLayout>
</RelativeLayout>12345678910111213141516171819202122232425262728293031323334353637383940414243444546474849505152535455565758596061626364656667686970717273747576777879808182838485868788899091929394959697989910010110210310410510610710810911011111211311411511611711811912012122
```

3、左右卡片+圆角+背景色：layout_share_lrcard.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/subview"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/bg_share_card"
    android:gravity="center_horizontal"
    android:orientation="vertical"
    android:padding="@dimen/share_card_space">

    <ImageView
```

```
        android:id="@+id/icon_iv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@mipmap/share_card1" />

    <View
        android:layout_width="match_parent"
        android:layout_height="1dip"
        android:layout_marginTop="@dimen/share_card_space"
        android:background="@color/theme_border" />

    <TextView
        android:id="@+id/title_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="10dip"
        android:layout_marginTop="10dip"
        android:singleLine="true"
        android:textColor="@color/theme_color"
        android:textSize="@dimen/share_card_text" />

    <TextView
        android:id="@+id/content_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:lineSpacingMultiplier="1.5"
        android:maxLines="9"
        android:scrollbars="none"
        android:textSize="@dimen/share_card_text" />
</LinearLayout>
12345678910111213141516171819202122232425262728293031323334353637383940414 2
```

三、注意build.gradle引用的库:

```
compile 'com.jakewharton:butterknife:7.0.1'
compile 'com.google.code.gson:gson:2.6.2'12
```

butterknife：控件对象自动创建出来
gson：生成javabean需要