



登录
上午好! 2017年3月30日 星期四

阳和移动开发

关注移动互联网和移动APP开发工具、开发框架、测试工具、微信开发、Android源码、Android开源类库以及各种开源组件的IT科技网站

现在的位置: [首页](#) > [Android开发经验](#) > 正文
[RSS](#)
[上篇](#) [下篇](#)

Android自定义控件之仿知乎详情页

2015年12月26日 / [Android开发经验](#) / 共 11635字 / 字号 [小](#) [中](#) [大](#) / 评论关闭

效果图





由，结社自由的原则，虽然该原则已被违反了70年。

因为这段历史，麦德维杰夫被西方开玩笑地称为“红色罗伊”，他当了30年持不同政见者，最后却成了苏联唯一站起来维护共产主义的人。

包含的技术点

这个知乎的详情页面所包含的几个技术点：

1. support.v7包下的ToolBar的使用
2. ScrollView实现滑动顶部停靠
3. 监听手势滑动方向来显示和隐藏底部视图

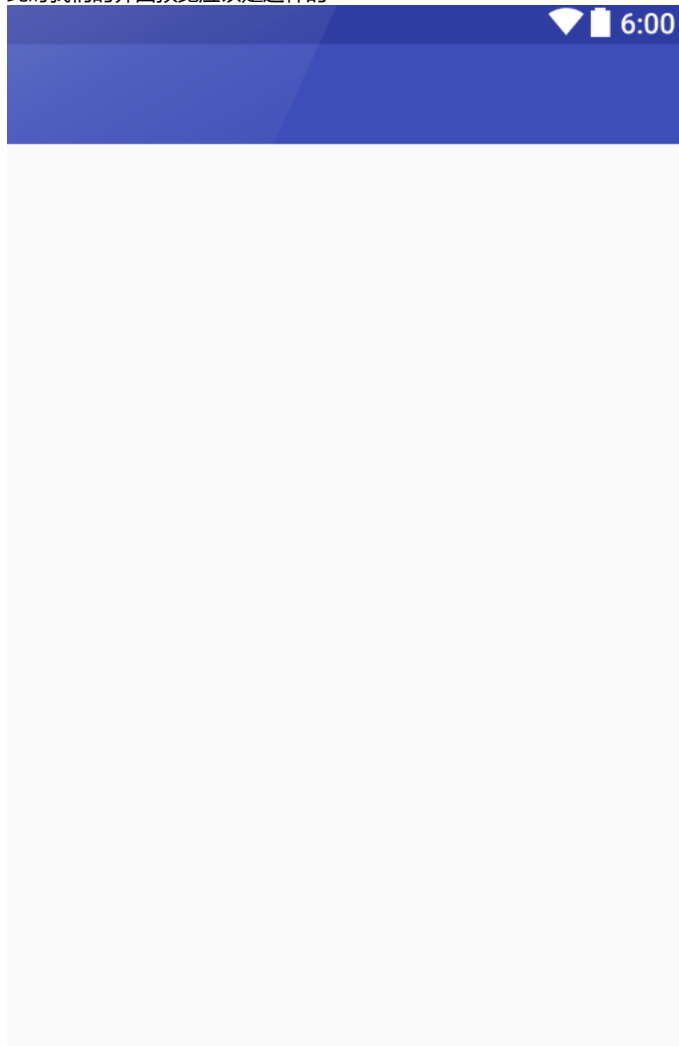
ToolBar的使用

知乎的Material Design版本顶部的导航是一个ToolBar控件，ToolBar是support.v7包下的一个控件，ToolBar的使用非常简单，首先我们现在layout文件夹中新建一个ToolBar.xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.Toolbar
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:local="http://schemas.android.com/apk/res-auto"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/toolbar"
    android:background="?attr/colorPrimary"
    local:theme="@style/Base.ThemeOverlay.AppCompat.Dark.ActionBar"
    android:popupTheme="@style/ThemeOverlay.AppCompat.Light"
    >
</android.support.v7.widget.Toolbar>
```



此时我们的界面预览应该是这样的

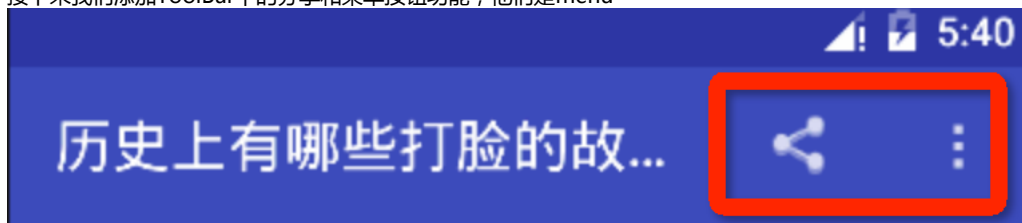


接下来我们在MainActivity中进行一下设置

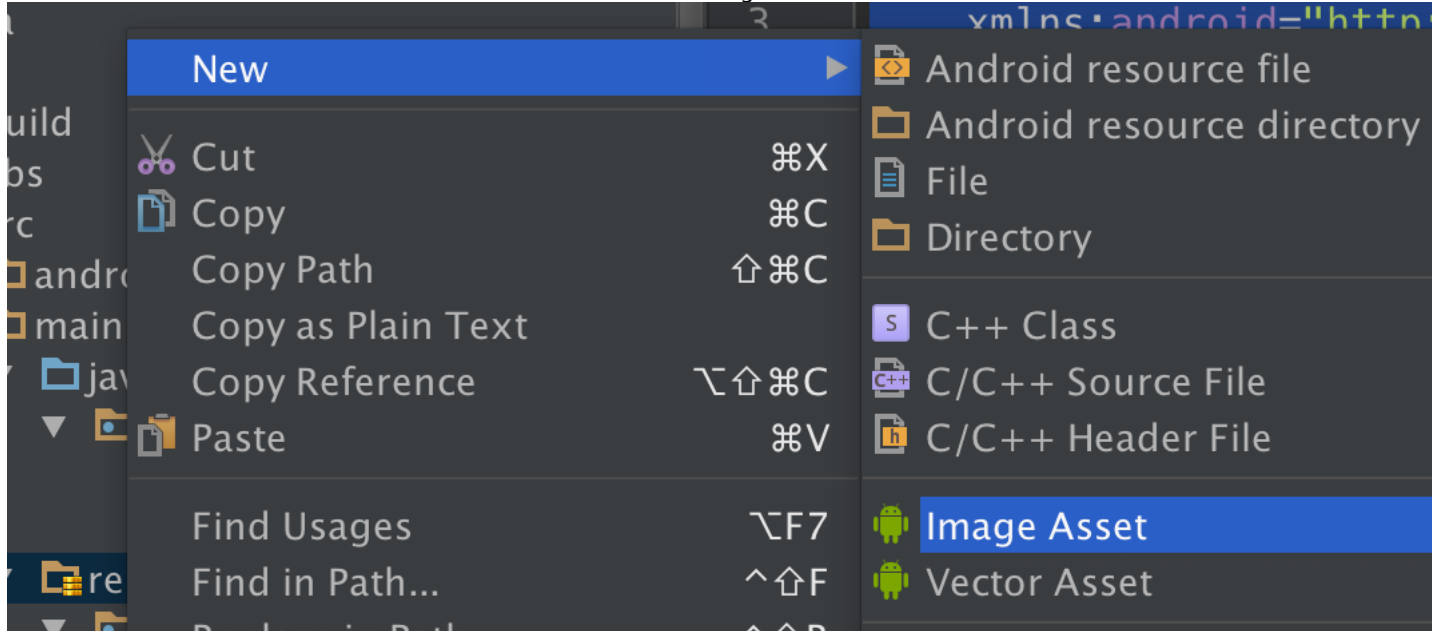
```
public class MainActivity extends AppCompatActivity{
    private Toolbar mToolbar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mToolbar = (Toolbar) findViewById(R.id.toolbar);
        mToolbar.setTitle("历史上有哪些打脸的故事?");
        setSupportActionBar(mToolbar);
    }
}
```

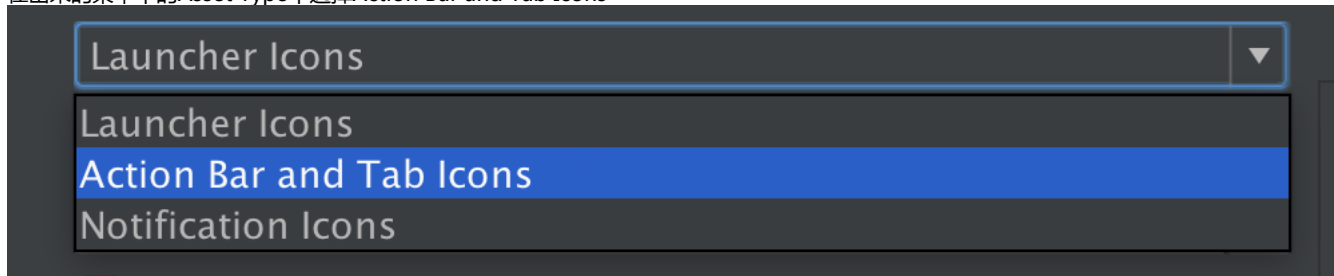
接下来我们添加ToolBar中的分享和菜单按钮功能，他们是menu



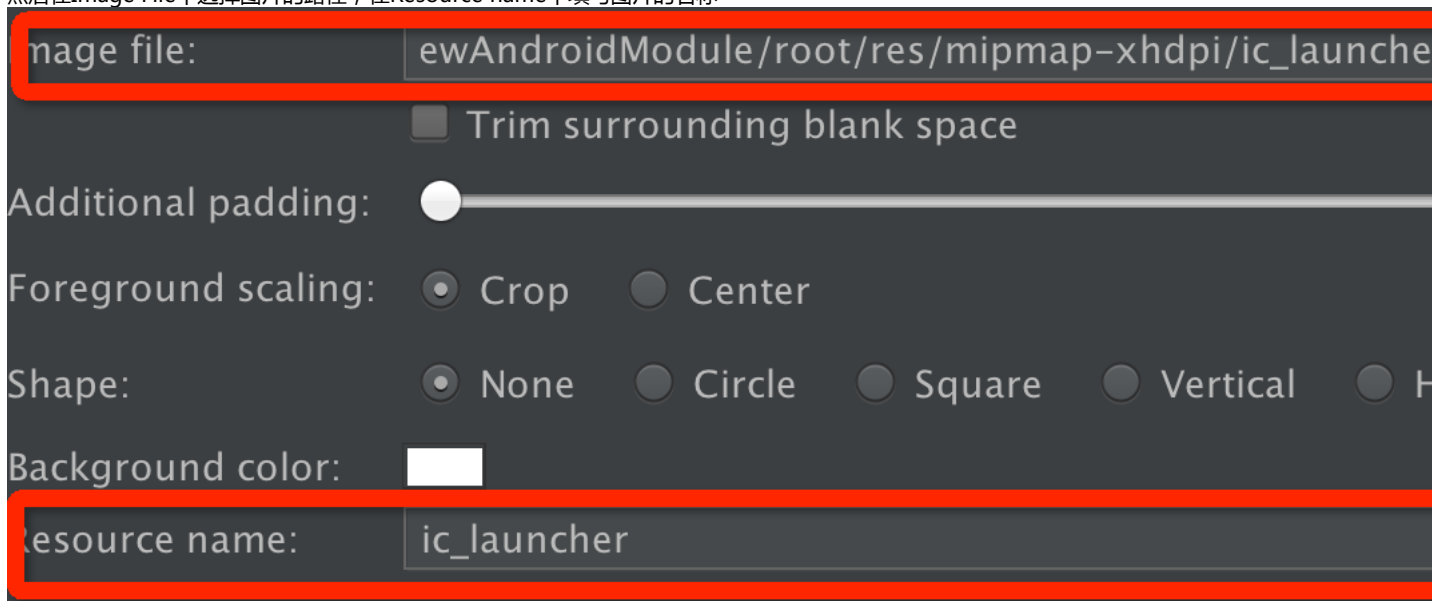
我们在我们的工程目录下的res文件夹上面单击右键，点击New，然后点击Image Asset



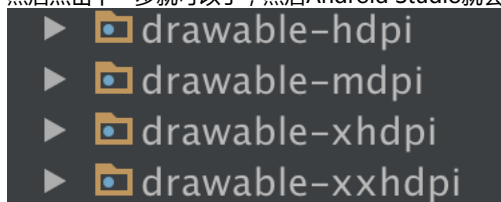
在出来的菜单中的Asset Type中选择Action Bar and Tab Icons



然后在Image File中选择图片的路径，在Resource name中填写图片的名称



然后点击下一步就可以了，然后Android Studio会自动帮我们创建下面这几个文件夹，并且图片也被添加进去了



接下来我们在res下创建一个menu文件夹，在menu文件夹内创建一个menu_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      >
    <item
```

```
        android:id="@+id/action_share"
        android:title="分享"
        android:icon="@drawable/ic_action_share"
        app:showAsAction="always"
    />
    <item
        android:id="@+id/action_menu"
        android:title="菜单"
        android:icon="@drawable/ic_action_menu"
        app:showAsAction="always"
    />
</menu>
```

接下来在MainActivity中重写onCreateOptionsMenu和onOptionsItemSelected方法，这段代码非常简单，而且我们新建项目的时候可以自动帮我们生成，我们只需要稍微修改下就可以了，我就不——解释了

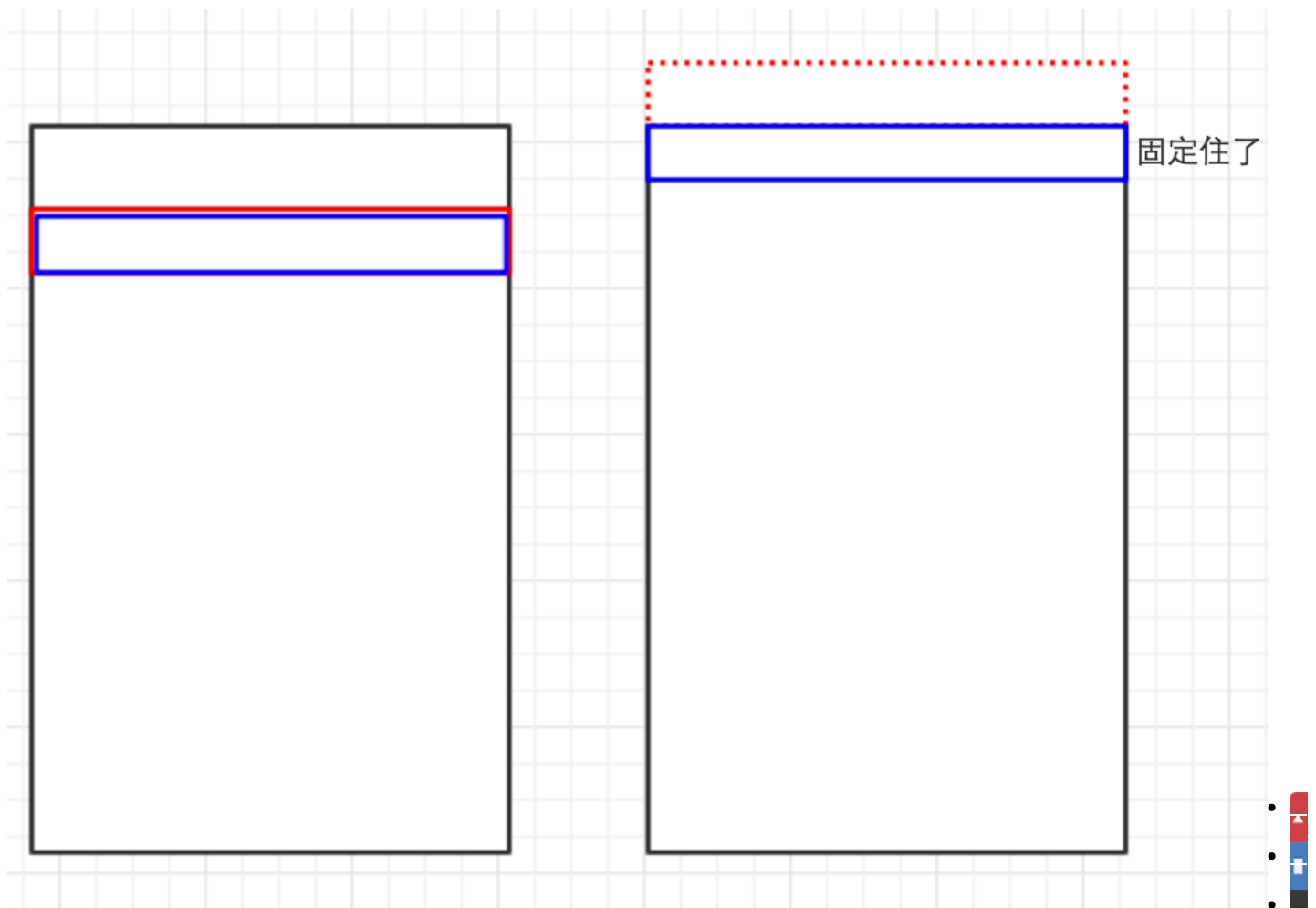
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menu_main, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if (id == R.id.action_share){
        Toast.makeText(MainActivity.this, "分享", Toast.LENGTH_SHORT).show();
        return true;
    }
    if (id == R.id.action_menu){
        Toast.makeText(MainActivity.this, "菜单", Toast.LENGTH_SHORT).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

好了，ToolBar已经准备完毕了，我们可以运行一下项目看一下效果，是不是和知乎的一样！

ScrollView实现滑动顶部停靠

这个功能我是参考[Android 仿美团网,大众点评购买框悬浮效果之修改版](#)这篇文章来实现的，他的实现思路非常牛x，我最开始的想法是监听要停留在顶部的View的滑动位置，当它滑动到顶部的时候再创建一个和他一模一样的View显示在顶部的位置，但是我发现这样做非常的麻烦，但是这篇文章的作者使用了另一种思路，他将两个View都创建出来，只不过他们一开始是重合的，我们看上去就像只有一个View一样，当View滑动到顶部时，上面覆盖的那个View就固定住了，从而我们视觉上感觉是这个View停靠在了顶部。



我们创建两个一模一样的布局，一个是蓝色覆盖在上面，一个是红色在下面，当我们向上滑动的时候，红色的顺着滑出去，而上面的蓝色的就停留在顶部了，这样就形成了滑动顶部停靠的效果。

首先我们先自定义一个ScrollView，由于ScrollView没有onScrollListener，所以我们必须要自己写一个onScrollListener

```
public class MyScrollView extends ScrollView{
    private OnScrollListener mListener;

    public interface OnScrollListener{
        void onScroll(int scrollY);
    }

    public MyScrollView(Context context) {
        super(context);
    }

    public MyScrollView(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    public MyScrollView(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
    }

    public void setOnScrollListener(OnScrollListener listener){
        mListener = listener;
    }

    @Override
    protected int computeVerticalScrollRange() {
        return super.computeVerticalScrollRange();
    }

    @Override
    protected void onScrollChanged(int l, int t, int oldl, int oldt) {
        super.onScrollChanged(l, t, oldl, oldt);
        if (mListener!=null){
            mListener.onScroll(t);
        }
    }
}
```

我们在onScrollChanged中可以获得当前ScrollView的滑动位置，我们回调调用mListener的onScroll方法并且将当前ScrollView的滑动位置传给MainActivity

在MainActivity中我们实现MyScrollView.OnScrollListener接口，并且重写onScroll方法，在onScroll方法中设置蓝色View的位置为和红色View重合
先来看一下Activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:id="@+id/container"
    tools:context="com.zhangqi.zhihudetail.MainActivity">

    <include
        android:id="@+id/toolbar"
        layout="@layout/toolbar" />

    <com.zhangqi.zhihudetail.MyScrollView
        android:id="@+id/myscrollview"
        android:layout_below="@id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
    >

        <FrameLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
        >

            <LinearLayout
                android:layout_width="match_parent"
                android:orientation="vertical"
                android:layout_height="match_parent">
                <TextView
                    android:id="@+id/tv_title"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content"
                    android:background="#efefef"
                    android:paddingBottom="10dp"
                    android:paddingLeft="5dp"
                    android:paddingTop="10dp"
                    android:text="历史有哪些打脸的故事?"
                    android:textColor="#ababab"
                    android:textSize="18sp" />

                <include
                    android:id="@+id/user_detail"
                    layout="@layout/user_detail_view"
                />

                <WebView
                    android:id="@+id/webview"
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content">
                </WebView>
            </LinearLayout>
            <include
                android:id="@+id/top_user_detail"
                layout="@layout/user_detail_view"
            />

        </FrameLayout>
    </com.zhangqi.zhihudetail.MyScrollView>
</RelativeLayout>
```

其中我将顶部停靠的View的布局抽取出来了，因为要重用，所以抽取出来使用include重用即可

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/rl_user_detail"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#ffffff"
    android:padding="10dp">

    <ImageView
        android:id="@+id/iv_avatar"
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:src="@drawable/sso_zhihu_logo" />

    <TextView
        android:id="@+id/tv_nickname"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_toRightOf="@id/iv_avatar"
        android:paddingLeft="5dp"
        android:text="神灯"
        android:textColor="#ff000000"
```



```

        android:textSize="16sp" />

<TextView
    android:id="@+id/tv_detail"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@id/tv_nickname"
    android:layout_below="@id/tv_nickname"
    android:paddingLeft="5dp"
    android:paddingTop="5dp"
    android:text="阿拉灯神灯"
    android:textColor="#bcbcbc" />

<TextView
    android:id="@+id/tv_like_num"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:drawableLeft="@drawable/ic_vote_normal"
    android:drawablePadding="10dp"
    android:gravity="center_vertical"
    android:padding="5dp"
    android:text="3028" />

<View
    android:layout_width="1dp"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@id/tv_like_num"
    android:layout_alignTop="@id/tv_like_num"
    android:layout_centerVertical="true"
    android:layout_marginBottom="2dp"
    android:layout_marginRight="5dp"
    android:layout_marginTop="2dp"
    android:layout_toLeftOf="@id/tv_like_num"
    android:background="#CCC" />
</RelativeLayout>

```

接下来看MainActivity

```

public class MainActivity extends AppCompatActivity implements MyScrollView.OnScrollListener{

    //自定义的ScrollView
    private MyScrollView mScrollView;
    //随着ScrollView滑走的View
    private RelativeLayout mUserDetail;
    //固定在顶部的View
    private RelativeLayout mTopUserDetail;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        initView();
    }

    private void initView() {
        mUserDetail = (RelativeLayout) findViewById(R.id.user_detail);
        mTopUserDetail = (RelativeLayout) findViewById(R.id.top_user_detail);
        mScrollView = (MyScrollView) findViewById(R.id.myscrollview);
        mScrollView.setOnScrollListener(this);
        //当布局中所有的View都测量完后回调的方法，我们在这个方法中可以拿到View的宽和高
        //在这个方法中调用onScroll是为什么？
        //因为我们要在onScroll中获得mUserDetail距顶部的高度
        //只有在所有的View都测量完后我们才能拿到这个高度值，否则我们拿到的是0
        //所以在onGlobalLayout中调用一下onScroll方法，我们一定可以拿到mUserDetail这个View
        //距离屏幕顶部的距离，从而设置给我们的mTopUserDetail这个View，实现两个View的重合
        findViewById(R.id.container).getViewTreeObserver().addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener() {
            @Override
            public void onGlobalLayout() {
                onScroll(mScrollView.getScrollY());
            }
        });
    }

    @Override
    public void onScroll(int scrollY) {
        //在最开始mUserDetail距离屏幕顶部是有一段距离的，而最开始scrollY=0，
        //所以在最开始的时候我们取两者的最大值就可以使两个View重合起来
        //因为我们是所有的View都测量完毕后调用过onScroll方法的，
        //所以mUserDetail.getTop()得到的值是正确的值
        int userDetailView2Top = Math.max(scrollY, mUserDetail.getTop());
        //调用mTopUserDetail的layout方法，设置其在屏幕上的位置
        mTopUserDetail.layout(0, userDetailView2Top, mTopUserDetail.getWidth(), userDetailView2Top + mTopUserDetail.getHeight());
    }
}

```

现在我们已经可以实现滑动停靠的功能了，接下来我们再来实现屏幕底部的View随着滑动方向显示和隐藏的功能

屏幕底部View随滑动方向显示和隐藏功能

我们看到屏幕底部有一个布局，当我们手指向上滑动的时候，底部的View是隐藏的，为了给我们更好地阅读体验，当我们手指向下滑动的时候，底部的View是显示出来的，提供给我们一些功能。

那么我们就修改刚才自定义的ScrollView，给onScrollListener添加两个方法，一个是向上滑动，一个是向下滑动

```
public interface OnScrollListener{
    void onScroll(int scrollY);
    void onScrollToTop();
    void onScrollToBottom();
}
```

那么我们怎么来判断用户是向上滑动还是向下滑动的呢？我们只需要重写ScrollView的onTouchEvent方法

```
@Override
public boolean onTouchEvent(MotionEvent ev) {
    if (mListener!=null) {
        switch (ev.getAction()) {
            case MotionEvent.ACTION_DOWN:
                //记录按下时的Y坐标
                downY = (int) ev.getY();
                break;
            case MotionEvent.ACTION_MOVE:
                //记录滑动时的Y坐标
                int moveY = (int) ev.getY();
                //计算出一个差值
                offsetY = moveY - downY;
                downY = moveY;
                break;
            case MotionEvent.ACTION_UP:
                //当手指抬起时判断差值的大小
                if (offsetY < 0) { //如果小于0，则说明用户手指向上滑动
                    mListener.onScrollToBottom();
                } else { //如果大于0，则说明用户手指向下滑动
                    mListener.onScrollToTop();
                }
                break;
        }
    }
    return super.onTouchEvent(ev);
}
```

接下来我们要在MainActivity中重写这两个方法

```
@Override
public void onScrollToTop() {
    if (!ll_bottom.isShown()) {
        ll_bottom.clearAnimation();
        ll_bottom.startAnimation(showAnim);
        ll_bottom.setVisibility(View.VISIBLE);
    }
}

@Override
public void onScrollToBottom() {
    if (ll_bottom.isShown()) {
        ll_bottom.clearAnimation();
        ll_bottom.startAnimation(dismissAnim);
        ll_bottom.setVisibility(View.GONE);
    }
}
```

其中ll_bottom就是我们底部的布局，他的xml如下

```
<LinearLayout
    android:id="@+id/ll_bottom"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingTop="10dp"
    android:paddingBottom="10dp"
    android:layout_alignParentBottom="true"
    android:background="#ffffff"
    android:orientation="horizontal">

    <TextView
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:drawableTop="@drawable/ic_nohelp"
        android:text="没有帮助"
        android:textColor="#ababab"
        android:drawablePadding="10dp"
        android:gravity="center_horizontal"
    />
```

```
<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:drawableTop="@drawable/ic_thank"
    android:text="感谢"
    android:textColor="#ababab"
    android:drawablePadding="10dp"
    android:gravity="center_horizontal"/>

<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:drawableTop="@drawable/ic_collect"
    android:text="收藏"
    android:textColor="#ababab"
    android:drawablePadding="10dp"
    android:gravity="center_horizontal"/>

<TextView
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:drawableTop="@drawable/ic_comment"
    android:text="评论 184"
    android:textColor="#ababab"
    android:drawablePadding="10dp"
    android:gravity="center_horizontal"/>
</LinearLayout>
```

其中的showAnim和dismissAnim是

```
showAnim = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.bottom_show);
dismissAnim = AnimationUtils.loadAnimation(getApplicationContext(),R.anim.bottom_dismiss);

<?xml version="1.0" encoding="utf-8"?>
<!--bottom_show.xml-->
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="200"
        android:fromYDelta="10%p"
        android:toYDelta="0"
    />
</set>

<?xml version="1.0" encoding="utf-8"?>
<!--bottom_dismiss.xml-->
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <translate
        android:duration="1000"
        android:fromYDelta="0"
        android:toYDelta="100%p"
    />
</set>
```

完整代码

好了现在所有的功能都已经实现了，博客将各个功能分开写了，是为了让大家清晰了解每个功能的实现方式，但是这样确实对于项目的完整性有一定的影响，我将代码提交到了我的GitHub中，大家可以到[我的GitHub](#)上下载完整代码，然后再配合博客中各个功能模块的讲解，希望对大家有所帮助

0

[返回](#)

- 【上篇】[微信：JSSDK开发](#)
- 【下篇】[微信支付开发的基本步骤](#)



作者: [summer](#)

- 该日志由 summer 于1年前发表在[Android开发经验](#)分类下，最后更新于 2015年12月26日.
- 转载请注明: [Android自定义控件之仿知乎详情页 | 阳和移动开发](#) +[复制链接](#)
-

抱歉!评论已关闭.

栏目导航

-

资讯

- [手机资讯](#)
- [移动开发资讯](#)

•

开发工具相关

- [开发工具和框架](#)
- [UI设计与开发](#)
- [集成IDE](#)
- [SDK](#)
- [游戏开发](#)
- [代码转换](#)
- [在线开发工具](#)

•

数据库相关

- [数据库](#)
- [NoSQL](#)

•

组件类库

- [ORM](#)
- [REST](#)
- [图片处理](#)
- [广告](#)
- [JSON](#)
- [类库管理](#)
- [网络通信](#)
- [通用框架](#)
- [任务管理](#)
- [评级\(Ratings\)](#)
- [PDF](#)
- [工具\(Toolkits\)](#)
- [实用工具类\(Utills\)](#)
- [XMPP](#)
- [日志\(Logging\)](#)
- [消息推送\(Push\)](#)
- [发布平台](#)
- [Bug报告\(Crash Reports\)](#)
- [缓存\(Caching\)](#)
- [切面\(Aspects\)](#)
- [音频\(Audio\)](#)
- [条形码\(Barcodes\)](#)
- [Blocks](#)
- [特效\(Specially Effect\)](#)
- [浏览器组件](#)
- [并发\(Concurrency\)](#)
- [云\(Cloud\)](#)
- [文件系统\(File system\)](#)
- [地图\(Maps\)](#)
- [图表\(Chart\)](#)
- [内容提供\(ContentProvider\)](#)
- [应用锁\(AppLock\)](#)
- [Parcelables](#)
- [RSS](#)
- [动画\(Animations\)](#)
- [框架架构\(Architecture\)](#)
- [SVG](#)
- [Gradle插件\(Gradle Plugins\)](#)
- [安全\(Security\)](#)
- [依赖注入\(Dependency Injections\)](#)
- [字体\(Fonts\)](#)
- [视频\(Video\)](#)
- [本地化\(Localization\)](#)
- [后台处理\(Background Processing\)](#)
- [事件总线\(Event Buses\)](#)



- [Preferences](#)
- [选择器 \(Picker \)](#)
- [日期选择器\(Date Pickers\)](#)
- [图片选择器\(Image Pickers\)](#)
- [颜色选择器\(Color Pickers\)](#)
- [视图效果 \(View Effects \)](#)
- [APT](#)
- [蓝牙\(Bluetooth\)](#)
- [变更日志\(ChangeLog\)](#)
- [视图适配器\(View Adapters\)](#)
- [翻页效果\(Curl/Flip Effects\)](#)
- [键盘](#)
- [视图切换\(View Transition\)](#)
- [图标\(Icons\)](#)
- [相机\(Camera\)](#)
- [XML](#)
- [ANR\(应用无响应\)](#)
- [波纹效果\(Ripple Effects\)](#)
- [通知\(Notifications\)](#)
- [分析 \(Analytics \)](#)
- [任务调度\(Job Schedulers\)](#)
- [OpenGL](#)
- [FRP](#)
- [USB](#)
- [Intent \(意图 \)](#)
- [Drawables](#)
- [定位\(Location\)](#)
- [SOAP](#)
- [Purchases](#)
- [天气\(Weather\)](#)

•

UI组件

- [滑动面板\(Sliding Panels\)](#)
- [表格视图\(Grid Views\)](#)
- [对话框\(Dialogs\)](#)
- [进度条](#)
- [UI布局\(Layouts\)](#)
- [按钮\(Buttons\)](#)
- [日历\(Calendars\)](#)
- [滚动视图\(ScrollView\)](#)
- [文本输入\(Textfield\)](#)
- [操作栏\(Action Bars\)](#)
- [列表视图\(List Views\)](#)
- [下拉刷新\(Pull Refresh\)](#)
- [文字视图\(Text Views\)](#)
- [Toasts](#)
- [标签栏\(Tab Bars\)](#)
- [菜单 \(Menu \)](#)
- [图标提醒\(Badges\)](#)
- [Table Views](#)
- [状态栏\(Status Bars\)](#)
- [Range Bars](#)
- [View Pagers](#)
- [模板引擎\(Template Engines\)](#)
- [引导页 \(Intro&Guide View \)](#)
- [指示器\(ActivityIndicator\)](#)
- [弹出视图 \(Popup View \)](#)
- [开关 \(Switch \)](#)
- [标签\(Label\)](#)
- [手势交互\(Gesture\)](#)
- [滑杆\(Slider\)](#)
- [Cards](#)
- [导航栏\(Navigation Bar\)](#)
- [透明指示层\(HUD\)](#)
- [视图\(Views\)](#)
- [Recycler Views](#)
- [Carousels](#)
- [校验\(Validation\)](#)

•



测试工具

-

调试工具

-

社交分享组件

-

插件

-

开发语言

-

其它

- [关于本站](#)

返回首页



Copyright © 2014-2017 阳和移动开发 保留所有权利. Theme by [Robin](#) 备案号：琼ICP备13000548号-2 [网站地图](#) 