**P2P金融项目**

## 项目实现

### 功能一：页面的基本架构

#### 实现效果：



#### MainActivity对应的布局：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"** >   <**FrameLayout  android:id="@+id/fl\_main\_content"  android:layout\_width="match\_parent"  android:layout\_height="0dp"  android:layout\_weight="1"**/>   <**include layout="@layout/main\_bottom"**></**include**> </**LinearLayout**> |

##### 其中，main\_bottom.xml定义为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="55dp"  android:orientation="vertical"**>  <**TextView  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/home\_back\_selected"** />  <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="54dp"  android:orientation="horizontal"** >  <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/ll\_main\_home"  android:orientation="vertical"  android:gravity="center"** >  <**ImageView  android:layout\_marginTop="2dp"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:src="@drawable/bottom01"  android:id="@+id/iv\_main\_home"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/tv\_main\_home"  android:text="首页"  android:textSize="18sp"  android:layout\_marginTop="1dp"  android:textColor="@color/home\_back\_unselected"** />  </**LinearLayout**>  <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/ll\_main\_invest"  android:orientation="vertical"  android:gravity="center"** >  <**ImageView  android:layout\_marginTop="2dp"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:src="@drawable/bottom03"  android:id="@+id/iv\_main\_invest"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/tv\_main\_invest"  android:text="投资"  android:textSize="18sp"  android:layout\_marginTop="1dp"  android:textColor="@color/home\_back\_unselected"** />  </**LinearLayout**>  <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/ll\_main\_me"  android:orientation="vertical"  android:gravity="center"** >  <**ImageView  android:layout\_marginTop="2dp"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:src="@drawable/bottom05"  android:id="@+id/iv\_main\_me"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/tv\_main\_me"  android:text="我的资产"  android:textSize="18sp"  android:layout\_marginTop="1dp"  android:textColor="@color/home\_back\_unselected"** />  </**LinearLayout**>  <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="match\_parent"  android:layout\_weight="1"  android:id="@+id/ll\_main\_more"  android:orientation="vertical"  android:gravity="center"** >  <**ImageView  android:layout\_marginTop="2dp"  android:layout\_width="30dp"  android:layout\_height="30dp"  android:src="@drawable/bottom07"  android:id="@+id/iv\_main\_more"** />  <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:id="@+id/tv\_main\_more"  android:text="更多"  android:textSize="18sp"  android:layout\_marginTop="1dp"  android:textColor="@color/home\_back\_unselected"** />  </**LinearLayout**>  </**LinearLayout**> </**LinearLayout**> |

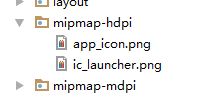
##### 布局中涉及到colors.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**resources**>  <**color name="colorPrimary"**>#3F51B5</**color**>  <**color name="colorPrimaryDark"**>#303F9F</**color**>  <**color name="colorAccent"**>#FF4081</**color**>   <**color name="home\_back\_unselected"**>#676767</**color**>  <**color name="home\_back\_selected"**>#3883D1</**color**>  <**color name="home\_back\_selected01"**>#E24242</**color**>  <**color name="title\_back"**>#2F83C7</**color**>  <**color name="product\_detail\_common"**>#666666</**color**>  <**color name="text\_progress"**>#18b4ed</**color**>  <**color name="title\_text"**>#26B8EE</**color**>  <**color name="divider\_line"**>#e1e1e1</**color**>  <**color name="product\_red\_common"**>#e84768</**color**>  <**color name="round\_red\_common"**>#DC143C</**color**>  <**color name="my\_title\_bg"**>#28baed</**color**>  <**color name="more\_text"**>#5A6E9E</**color**>  <**color name="white"**>#ffffff</**color**> </**resources**> |

##### 提供应用的图标

|  |
| --- |
| **android:icon="@mipmap/app\_icon"** |

其中，使用mipmap下app\_icon图标



##### 设置应用的名称

在strings.xml中：

|  |
| --- |
| <**string name="app\_name"**>硅谷金融</**string**> |

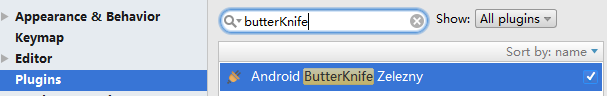
#### 回到MainActivity中：

|  |
| --- |
| @Override **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_main***);  ButterKnife.*bind*(**this**);  } |

#### 这里使用第三方框架ButterKnife。Gradle中配置如下：

|  |
| --- |
| compile **'com.jakewharton:butterknife:7.0.1'** |

同时，提供插件：



#### 提供点击的回调：

|  |
| --- |
| **private** HomeFragment **homeFragment**; **private** InvestFragment **investFragment**; **private** MeFragment **meFragment**; **private** MoreFragment **moreFragment**; **private** FragmentTransaction **transaction**;  @OnClick({R.id.***ll\_item\_home***, R.id.***ll\_item\_invest***, R.id.***ll\_item\_me***, R.id.***ll\_item\_more***}) **public void** changeTab(View v) {  Log.*e*(**"TAG"**, **"11111"**);  FragmentManager supportFragmentManager = **this**.getSupportFragmentManager();  **transaction** = supportFragmentManager.beginTransaction();   hideFragment();  **switch** (v.getId()) {  **case** R.id.***ll\_item\_home***:  **if** (**homeFragment** == **null**) {  **homeFragment** = **new** HomeFragment();  **transaction**.add(R.id.***fl\_main\_content***, **homeFragment**);  }  **transaction**.show(**homeFragment**);  **break**;  **case** R.id.***ll\_item\_invest***:  **if** (**investFragment** == **null**) {  **investFragment** = **new** InvestFragment();  **transaction**.add(R.id.***fl\_main\_content***, **investFragment**);  }  **transaction**.show(**investFragment**);  **break**;  **case** R.id.***ll\_item\_me***:  **if** (**meFragment** == **null**) {  **meFragment** = **new** MeFragment();  **transaction**.add(R.id.***fl\_main\_content***, **meFragment**);  }  **transaction**.show(**meFragment**);  **break**;  **case** R.id.***ll\_item\_more***:  **if** (**moreFragment** == **null**) {  **moreFragment** = **new** MoreFragment();  **transaction**.add(R.id.***fl\_main\_content***, **moreFragment**);  }  **transaction**.show(**moreFragment**);  **break**;   }  **transaction**.commit(); } |

##### 其中，hideFragment()方法如下：

|  |
| --- |
| **private void** hideFragment() {  **if**(**homeFragment** != **null**){  **transaction**.hide(**homeFragment**);  }  **if**(**investFragment** != **null**){  **transaction**.hide(**investFragment**);  }  **if**(**meFragment** != **null**){  **transaction**.hide(**meFragment**);  }  **if**(**moreFragment** != **null**){  **transaction**.hide(**moreFragment**);  } } |

#### 页面中提供的HomeFragment如下：

|  |
| --- |
| **public class** HomeFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getContext(), R.layout.***fragment\_home***, **null**);   **return** view;  } } |

##### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="首页"  android:textSize="40sp"  android:textColor="@android:color/holo\_blue\_bright"** />  </**LinearLayout**> |

#### 页面中提供的InvestFragment如下：

|  |
| --- |
| **public class** InvestFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getContext(), R.layout.***fragment\_invest***, **null**);  **return** view;  } } |

##### 对应的布局如下：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="投资"  android:textSize="40sp"  android:textColor="@android:color/holo\_blue\_bright"** />  </**LinearLayout**> |

#### 页面中提供的MeFragment如下：

|  |
| --- |
| **public class** MeFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getContext(), R.layout.***fragment\_me***, **null**);  **return** view;  } } |

##### 对应的布局如下：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="我的资产"  android:textSize="40sp"  android:textColor="@android:color/holo\_blue\_bright"** />  </**LinearLayout**> |

#### 页面中提供的MoreFragment如下：

|  |
| --- |
| **public class** MoreFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getContext(), R.layout.***fragment\_more***, **null**);   **return** view;  } } |

##### 对应的布局如下：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="更多"  android:textSize="40sp"  android:textColor="@android:color/holo\_blue\_bright"** />  </**LinearLayout**> |

实现切换时，颜色的改变

|  |
| --- |
| *//重置现有的视图的颜色* reSetTab(); **switch** (v.getId()) {  **case** R.id.***ll\_item\_home***:  **if** (**homeFragment** == **null**) {  **homeFragment** = **new** HomeFragment();  **transaction**.add(R.id.***fl\_main\_content***, **homeFragment**);  }  *//设置为选中* **ivItemHome**.setImageResource(R.drawable.***bid01***); **tvItemHome**.setTextColor(getResources().getColor(R.color.***home\_back\_selected***));  **transaction**.show(**homeFragment**);  **break**;  **case** R.id.***ll\_item\_invest***:  **if** (**investFragment** == **null**) {  **investFragment** = **new** InvestFragment();  **transaction**.add(R.id.***fl\_main\_content***, **investFragment**);  }  **ivItemInvest**.setImageResource(R.drawable.***bid03***); **tvItemInvest**.setTextColor(getResources().getColor(R.color.***home\_back\_selected***));  **transaction**.show(**investFragment**);  **break**;  **case** R.id.***ll\_item\_me***:  **if** (**meFragment** == **null**) {  **meFragment** = **new** MeFragment();  **transaction**.add(R.id.***fl\_main\_content***, **meFragment**);  }  **ivItemMe**.setImageResource(R.drawable.***bid05***); **tvItemMe**.setTextColor(getResources().getColor(R.color.***home\_back\_selected***));  **transaction**.show(**meFragment**);  **break**;  **case** R.id.***ll\_item\_more***:  **if** (**moreFragment** == **null**) {  **moreFragment** = **new** MoreFragment();  **transaction**.add(R.id.***fl\_main\_content***, **moreFragment**);  }  **ivItemMore**.setImageResource(R.drawable.***bid07***); **tvItemMore**.setTextColor(getResources().getColor(R.color.***home\_back\_selected***));  **transaction**.show(**moreFragment**);  **break**;  } |

其中，reSetTab()定义如下：

|  |
| --- |
| **private void** reSetTab() {  **ivItemHome**.setImageResource(R.drawable.***bid02***);  **ivItemInvest**.setImageResource(R.drawable.***bid04***);  **ivItemMe**.setImageResource(R.drawable.***bid06***);  **ivItemMore**.setImageResource(R.drawable.***bid08***);  **tvItemHome**.setTextColor(getResources().getColor(R.color.***home\_back\_unselected***));  **tvItemInvest**.setTextColor(getResources().getColor(R.color.***home\_back\_unselected***));  **tvItemMe**.setTextColor(getResources().getColor(R.color.***home\_back\_unselected***));  **tvItemMore**.setTextColor(getResources().getColor(R.color.***home\_back\_unselected***));  } |

设置默认状态下，显示首页：

在onCreate()中调用initData()方法，此方法定义如下：

|  |
| --- |
| **private void** initData() {  FragmentManager supportFragmentManager = **this**.getSupportFragmentManager();  **transaction** = supportFragmentManager.beginTransaction();  **if** (**homeFragment** == **null**) {  **homeFragment** = **new** HomeFragment();  **transaction**.add(R.id.***fl\_main\_content***, **homeFragment**);  }  *//设置为选中* **ivItemHome**.setImageResource(R.drawable.***bid01***);  **tvItemHome**.setTextColor(getResources().getColor(R.color.***home\_back\_selected***));  **transaction**.show(**homeFragment**);  **transaction**.commit(); } |

#### MainActivity中连续点击两次退出当前应用

|  |
| --- |
| **private boolean flag** = **true**;   **private** Handler **handler** = **new** Handler(){  **public void** handleMessage(Message msg){  **switch** (msg.**what**) {  **case** 1 :  **flag** = **true**;  **break**;  }  }  };  @Override  **public boolean** onKeyUp(**int** keyCode, KeyEvent event) {  **if**(keyCode == KeyEvent.***KEYCODE\_BACK***){  **if**(**flag**){  **flag** = **false**;  Toast.*makeText*(MainActivity.**this**, **"再点击一次，退出硅谷金融"**, Toast.***LENGTH\_SHORT***).show();  *//发送一个延迟消息* **handler**.sendEmptyMessageDelayed(1,2000);   **return true**;  }   }   **return super**.onKeyUp(keyCode,event);  }   @Override  **protected void** onDestroy() {  Log.*e*(**"TAG"**, **"onDestroy()"**);  *//方式一：* **handler**.removeMessages(1);  *//方式二： // handler.removeCallbacksAndMessages(null);* **super**.onDestroy();  } |

### 功能二：提供欢迎界面

|  |
| --- |
| **public class** WelcomeActivity **extends** BaseActivity {    @Bind(R.id.***round\_welcome***)  RoundProgress **roundWelcome**;  @Bind(R.id.***iv\_welcome\_icon***)  ImageView **ivWelcomeIcon**;   @Override  **protected void** initTitle() {   }   @Override  **protected void** initData() {   UIUtils.*getHandler*().postDelayed(**new** Runnable() {  @Override  **public void** run() {  startActivity(**new** Intent(WelcomeActivity.**this**,MainActivity.**class**));  finish();  }  }, 3000);  *//启动动画* showAnimation();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_welcome***;  }  }  */\*\*  \* 显示动画:  \* 透明度: 0--1 持续2s  \*/* **private void** showAnimation() {  *//透明度* AlphaAnimation alphaAnimation = **new** AlphaAnimation(0, 1);  alphaAnimation.setDuration(2000);   *//启动动画* **rlWelcome**.startAnimation(alphaAnimation); }  @Override **protected void** onDestroy() {  **super**.onDestroy();  UIUtils.*getHandler*().removeCallbacksAndMessages(**null**); } |

对应的布局：

|  |
| --- |
| *<?***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="match\_parent"  android:background="@drawable/start\_background"**>   <**ImageView  android:id="@+id/iv\_welcome\_icon"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="100dp"  android:background="@drawable/app\_icon"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/iv\_welcome\_icon"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="20dp"  android:text="硅谷金融"  android:textColor="@color/white"  android:textSize="40sp"** />   <**LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentBottom="true"  android:layout\_centerHorizontal="true"  android:layout\_marginBottom="100dp"  android:orientation="horizontal"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="当前版本:"  android:textColor="@color/white"  android:textSize="15sp"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="v 1.0"  android:textColor="@color/white"  android:textSize="15sp"** />   </**LinearLayout**>  </**RelativeLayout**> |

##### 设置为全屏显示：

|  |
| --- |
| * 方式一：动态编码的方式:   // 去掉窗口标题  requestWindowFeature(Window.*FEATURE\_NO\_TITLE*);  // 隐藏顶部的状态栏  getWindow().addFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);  setContentView();   * *方式二：*   android:theme=*"@android:style/Theme.Black.NoTitleBar"*  android:theme=*"@android:style/Theme.Black.NoTitleBar.Fullscreen"*   * *方式三*   <style name=*"AppTheme"* parent=*"AppBaseTheme"*>  <item name=*"android:windowNoTitle"*>true</item><!-- 没有标题 -->  </style> |

### 功能三：显示页面头

#### 显示效果：







#### 提供common\_title.xml文件如下：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="45dp"  android:orientation="vertical"**  **android:background="@drawable/lock\_title\_bg"**  >   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="44dp"  android:gravity="center\_vertical"  android:orientation="horizontal"**>   <**ImageView  android:id="@+id/iv\_title\_back"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="15dp"  android:src="@drawable/left"** />   <**TextView  android:id="@+id/tv\_title"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="首页"  android:gravity="center"  android:textColor="@color/title\_back"  android:textSize="25sp"** />   <**ImageView  android:id="@+id/iv\_title\_setting"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:src="@drawable/my\_setting\_icon"  android:layout\_marginRight="15dp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/title\_back"** ></**View**>   </**LinearLayout**> |

##### 然后在现有的4个Fragment对应的布局中提供：

|  |
| --- |
| <**include layout="@layout/common\_title"**></**include**> |

##### 回到Fragment中：

|  |
| --- |
| **public** View onCreateView()(  initTitle();  **return** view; }  **private void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  *//需要的话，再调整TextView的显示* } |

### 功能四：提供Application子类、UIUtils工具类、联网配置AppNetConfig类、栈管理AppManager类

#### 创建子包：common，提供MyApplication类：

|  |
| --- |
| **public class** MyApplication **extends** Application {   **public static** Context *context*;  **public static** Handler *handler*;  **public static** Thread *mainThread*;  **public static int** *mainThreadId*;   @Override  **public void** onCreate() {  **super**.onCreate();  *context* = getApplicationContext();  *handler* = **new** Handler();  *mainThread* = Thread.*currentThread*();  *mainThreadId* = android.os.Process.*myTid*();   } } |

#### 创建子包：util，提供UIUtils类：

|  |
| --- |
| */\*\*  \* 专门提供为处理一些UI相关的问题而创建的工具类  \*/* **public class** UIUtils {   **public static** Context getContext(){  **return** MyApplication.*context*;  }   **public static** Handler getHandler(){  **return** MyApplication.*handler*;  }   **public static int** getColor(**int** colorId){  **return** *getContext*().getResources().getColor(colorId);  }   **public static** View getView(**int** viewId){  **return** View.*inflate*(*getContext*(), viewId, **null**);  }   **public static** String[] getStringArr(**int** arrId){  **return** *getContext*().getResources().getStringArray(arrId);  }   *//与屏幕分辨率相关的* **public static int** dp2px(**int** dp){  **float** density = *getContext*().getResources().getDisplayMetrics().**density**;  **return** (**int**) (density \* dp + 0.5);  }   **public static int** px2dp(**int** px){  **float** density = *getContext*().getResources().getDisplayMetrics().**density**;  **return** (**int**) (px / density + 0.5);  }  } |

然后在MainActivity和4个Fragment中，初始化View和设置TextView的color时，可使用UIUtils工具类声明的方法。

#### common包下提供AppNetConfig类：

|  |
| --- |
| */\*\*  \* 配置网络请求相关的地址  \*/* **public class** AppNetConfig {   **public static final** String ***HOST*** = **"192.168.56.1"**;*//提供ip地址   //提供web应用的地址* **public static final** String ***BASE\_URL*** = **"http://"** + ***HOST*** + **":8080/P2PInvest/"**;   **public static final** String ***INDEX*** = ***BASE\_URL*** + **"index"**;*//访问首页数据* **public static final** String ***LOGIN*** = ***BASE\_URL*** + **"login"**;*//访问登录的url* **public static final** String ***PRODUCT*** = ***BASE\_URL*** + **"product"**;*//访问“所有理财”的url* **public static final** String ***UPDATE*** = ***BASE\_URL*** + **"update.json"**;*//访问服务器端当前应用的版本信息* **public static final** String ***REGISTER*** = ***BASE\_URL*** + **"UserRegister"**;*//注册* **public static final** String ***FEEDBACK*** = ***BASE\_URL*** + **"FeedBack"**;*//用户反馈* } |

#### common包下提供AppManager类：

|  |
| --- |
| */\*\*  \* 统一应用程序中所有的Activity的栈管理  \* 涉及到activity的添加、删除指定、删除当前、删除所有、返回栈大小的方法  \*/* **public class** AppManager {   **private** Stack<Activity> **activityStack** = **new** Stack<>();   *//设置当前类是单例的* **private static** AppManager *appManager* = **new** AppManager();   **private** AppManager(){}   **public static** AppManager getInstance(){  **return** *appManager*;  }   **public void** addActivity(Activity activity){  **activityStack**.add(activity);  }   **public void** removeActivity(Activity activity){  **for**(**int** i = **activityStack**.size() - 1; i >= 0 ; i--) {  **if**(**activityStack**.get(i).getClass().equals(activity.getClass())){  activity.finish();  **activityStack**.remove(i);  **break**;  }  }  }   **public void** removeCurrent(){ *// Activity activity = activityStack.get(activityStack.size() - 1);* Activity activity = **activityStack**.lastElement();  activity.finish();  **activityStack**.remove(activity);  }   **public void** removeAll(){  **for**(**int** i = **activityStack**.size() - 1; i >= 0 ; i--) {  **activityStack**.get(i).finish();  **activityStack**.remove(i);  }  }   **public int** size(){  **return activityStack**.size();  } } |

##### 在MainActivity中，将当前的Activity添加到栈管理中：

|  |
| --- |
| **protected void** onCreate(Bundle savedInstanceState) {  *//将当前的Activity添加到栈管理中* AppManager.*getInstance*().addActivity(**this**);  } |

### 功能五：未处理异常的全局捕获

#### common包下提供异常捕获类CrashHandler：

|  |
| --- |
| **public class** CrashHandler **implements** Thread.UncaughtExceptionHandler {   **private static** CrashHandler *crashHandler* = **null**;  **private** Context **mContext**;  **private** Thread.UncaughtExceptionHandler **defaultUncaughtExceptionHandler**;   **private** CrashHandler(){   }  //懒汉式实现单例  **public static** CrashHandler getInstance(){  **if**(*crashHandler* == **null**){  *crashHandler* = **new** CrashHandler();  }  **return** *crashHandler*;  }   **public void** init(Context context){  **this**.**mContext** = context;  **this**.**defaultUncaughtExceptionHandler** = Thread.*getDefaultUncaughtExceptionHandler*();  Thread.*setDefaultUncaughtExceptionHandler*(**this**);  }   *//异常信息发生时的处理：可以考虑发送给服务器* @Override  **public void** uncaughtException(Thread thread, Throwable ex) { *// Log.e("TAG", "出现了异常！");* **if**(ex == **null**){  **defaultUncaughtExceptionHandler**.uncaughtException(thread,ex);  }**else**{  handleException(thread, ex);  }  }  }  **private void** handleException(Thread thread, Throwable ex) {  *//必须在主线程执行Toast* **new** Thread(){  **public void** run(){  *//Android系统当中，默认情况下，线程是没有开启looper消息处理的，但是主线程除外* Looper.*prepare*();  Toast.*makeText*(**mContext**, **"亲，出现异常了！"**, Toast.***LENGTH\_SHORT***).show();  Looper.*loop*();  }  }.start();  *//搜集异常信息* collectionException(ex);   **try** {  *//关闭资源* thread.*sleep*(2000);  AppManager.*getInstance*().removeAll();  android.os.Process.*killProcess*(android.os.Process.*myPid*());  *////关闭虚拟机，释放所有内存:参数0代表正常退出* System.*exit*(0);  } **catch** (InterruptedException e) {  e.printStackTrace();  } }  **private void** collectionException(Throwable ex) {  **final** String deviceInfo = Build.***DEVICE*** + **":"** + Build.VERSION.***SDK\_INT*** + **":"** + Build.***MODEL*** + **":"** + Build.***PRODUCT***;  **final** String message = ex.getMessage();  **new** Thread(){  **public void** run(){  *//可以通过联网将信息发送给后台，所以在分线程执行* Log.*e*(**"TAG"**, **"deviceInfo:"** + deviceInfo + **",message:"** + message);  }  }.start();  } |

##### 在MyApplication的onCreate()中初始化：

|  |
| --- |
| **public void** onCreate() {  *//初始化当前的异常处理器* CrashHandler.*getInstance*().init(**this**); } |

### 功能六： ScrollView控件的使用

#### 在fragment\_home.xml中设置布局：

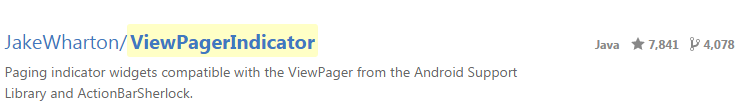
|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**></**include**>   <**ScrollView  android:id="@+id/myscrollview"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**FrameLayout  android:layout\_width="match\_parent"  android:layout\_height="150dp"**>   <**android.support.v4.view.ViewPager  android:id="@+id/vp\_barner"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**></**android.support.v4.view.ViewPager**>   *<!--<com.viewpagerindicator.CirclePageIndicator  android:id="@+id/circle\_barner"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom|center"  android:padding="10dp">  </com.viewpagerindicator.CirclePageIndicator>-->* </**FrameLayout**>    <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:layout\_marginTop="10dp"**>   <**TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerInParent="true"  android:drawablePadding="4dp"  android:drawableRight="@drawable/recommend"  android:text="乐享活180天计划第一期"  android:textColor="#303030"  android:textSize="20sp"** />  </**RelativeLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="60dp"  android:layout\_marginRight="60dp"  android:layout\_marginTop="5dp"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:background="@drawable/home\_attribute\_bg"  android:gravity="center"  android:text="本息保障"  android:textColor="#888888"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="4dp"  android:layout\_marginRight="4dp"  android:layout\_weight="1"  android:background="@drawable/home\_attribute\_bg"  android:gravity="center"  android:text="高额收益"  android:textColor="#888888"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:background="@drawable/home\_attribute\_bg"  android:gravity="center"  android:text="每日计息"  android:textColor="#888888"** />  </**LinearLayout**>   *<!-- 推荐产品信息-->* <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginBottom="10dp"  android:layout\_marginTop="10dp"  android:gravity="center"**>   *<!-- 圆环进度-->  <!--<com.atguigu.p2pmoney.ui.RoundProgress  android:id="@+id/p\_progresss"  android:layout\_width="120dp"  android:layout\_height="120dp"  app:roundColor="@android:color/darker\_gray"  app:roundProgressColor="@android:color/holo\_red\_dark"  app:roundWidth="10dp"  app:textColor="#18b4ed"  app:textSize="20sp">  </com.atguigu.p2pmoney.ui.RoundProgress>-->   <!--此Button暂时替换上面的RoundProgress -->* <**Button  android:id="@+id/p\_progresss"  android:layout\_width="120sp"  android:layout\_height="120sp"**></**Button**>  </**RelativeLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_horizontal|bottom"  android:orientation="horizontal"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="预期年利率"  android:textColor="#303030"  android:textSize="16sp"** />   <**TextView  android:id="@+id/p\_yearlv"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:gravity="center"  android:text="10.00%+2.00%"  android:textColor="#303030"  android:textSize="20sp"** />  </**LinearLayout**>   <**Button  android:id="@+id/button1"  android:layout\_width="match\_parent"  android:layout\_height="45dp"  android:layout\_gravity="center\_horizontal"  android:layout\_marginLeft="35dp"  android:layout\_marginRight="35dp"  android:layout\_marginTop="10dp"  android:background="@drawable/home"  android:text="立即加入"  android:textColor="#FFFFFF"  android:textSize="18sp"** />   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="48dp"  android:layout\_gravity="center"  android:drawableLeft="@drawable/guarantee\_icon"  android:gravity="center"  android:text="风险保障金由光大银行监管"  android:textColor="@color/product\_detail\_common"  android:textSize="13sp"** />   </**LinearLayout**>    </**ScrollView**>   </**LinearLayout**> |

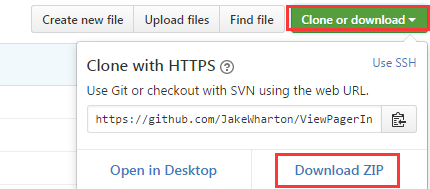
运行一下，看设计的布局

### 功能七：联网下载图片显示在ViewPager

#### github上下载框架：ViewPagerIndicator

import Module的方式导入到当前工程





#### 然后在fragment\_home.xml中的ScrollView中使用CirclePageIndicator

|  |
| --- |
| <**com.viewpagerindicator.CirclePageIndicator  android:id="@+id/circle\_barner"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_gravity="bottom|center"  android:padding="10dp"**> </**com.viewpagerindicator.CirclePageIndicator**> |

#### 在HomeFragment类中联网下载数据

##### 声明属性AsyncHttpClient：

|  |
| --- |
| AsyncHttpClient **client** = **new** AsyncHttpClient(); |

##### 在HomeFragment类的onCreateView()中加载方法initData():

|  |
| --- |
| **index** = **new** Index(); **client**.post(AppNetConfig.***INDEX***,**new** AsyncHttpResponseHandler(){  @Override  **public void** onSuccess(String content) {   JSONObject jsonObject = JSONObject.*parseObject*(content);   String proInfo = jsonObject.getString(**"proInfo"**);  Product product = JSON.*parseObject*(proInfo, Product.**class**);   String imageArr = jsonObject.getString(**"imageArr"**);  List<Image> imageList = JSON.*parseArray*(imageArr, Image.**class**);   **index**.**product** = product;  **index**.**imageList** = imageList;  Log.*e*(**"TAG"**, product.toString());   *//适配数据* MyAdapter myAdapter = **new** MyAdapter();  **vpBarner**.setAdapter(myAdapter);  *//设置关联的ViewPager* **circleBarner**.setViewPager(**vpBarner**);   }    @Override  **public void** onFailure(Throwable error, String content) {  **super**.onFailure(error, content);  Toast.*makeText*(MyApplication.*context*,error.getMessage(), Toast.***LENGTH\_SHORT***).show();  } }); |

其中，index声明为当前类的属性

###### Index类声明在bean包下：

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/8/4 0004.  \*/* **public class** Index {  **public** List<Image> **imageList**;  **public** Product **product**; } |

###### 其中，HomeFragment中提供内部类：

|  |
| --- |
| **class** MyAdapter **extends** PagerAdapter{    @Override  **public int** getCount() {  **return index**.**imageList** == **null**? 0 : **index**.**imageList**.size();  }   @Override  **public boolean** isViewFromObject(View view, Object object) {  **return** view == object;  }   @Override  **public** Object instantiateItem(ViewGroup container, **int** position) {  ImageView imageView = **new** ImageView(getActivity());  imageView.setScaleType(ImageView.ScaleType.***CENTER\_CROP***);   Picasso.*with*(getActivity()).load(**index**.**imageList**.get(position).**IMAURL**).into(imageView);   container.addView(imageView);  **return** imageView;  }   @Override  **public void** destroyItem(ViewGroup container, **int** position, Object object) {  container.removeView((View)object);  } } |

###### 其中，创建的bean包下提供Image类：

|  |
| --- |
| **public class** Image {  **public** String **ID**;   **public** String **IMAPAURL**;   **public** String **IMAURL**;  } |

###### 提供Product类：

|  |
| --- |
| **public class** Product {  **public** String **id**;  **public** String **memberNum**;  **public** String **minTouMoney**;  **public** String **money**;  **public** String **name**;  **public** String **progress**;  **public** String **suodingDays**;  **public** String **yearRate**;  } |

其中，Picasso为第三方联网下载数据的框架：

|  |
| --- |
| compile **'com.squareup.picasso:picasso:2.5.2'** |

思考：如何使得ViewPager中的图片自动切换，同时，如果手动滑动的话，移除自动切换功能，如果结束手动滑动，继续自动切换。

#### HomeFragment中加载的图片的显示效果的调整

现在的绝大数app都有banner界面，实现循环播放多个广告图片和手动滑动循环等功能。因为ViewPager并不支持循环翻页， 所以要实现循环还得需要自己去动手，我就把项目中的控件剔了出来，希望大家觉得有用。目前框架可以进行不同样式、不同动画设置， 以及完善的api方法能满足大部分的需求了。--作者

在build.gradle文件中集成库：（或者通过引用本地lib的方式）

|  |
| --- |
| compile 'com.youth.banner:banner:1.4' //指定版本 |

注释掉fragment\_home.xml文件中的<fragment></fragment>一组标签中显示viewpager和pageindicator的代码。然后改为：

|  |
| --- |
| <**com.youth.banner.Banner  android:id="@+id/banner"  android:layout\_width="match\_parent"  android:layout\_height="150dp"  atguigu:indicator\_height="7dp"  atguigu:indicator\_width="7dp"  atguigu:indicator\_margin="2dp"** /> |

回到HomeFragment中，注释掉Adapter和ViewPager的相关操作，然后：

|  |
| --- |
| *//设置banner样式* **banner**.setBannerStyle(BannerConfig.***CIRCLE\_INDICATOR\_TITLE***); *//设置图片加载器* **banner**.setImageLoader(**new** GlideImageLoader()); *//设置图片集合* List<String> imagesUrl = **new** ArrayList<>(images.size()); **for** (**int** i = 0; i < images.size(); i++) {  imagesUrl.add(images.get(i).**IMAURL**); } **banner**.setImages(imagesUrl); *//设置banner动画效果* **banner**.setBannerAnimation(Transformer.*ZoomOut*); *//设置标题集合（当banner样式有显示title时）* String[] titles = {**"硅谷金融金秋加息2%"**,**"乐享活180天计划"**,**"超级新手计划升级版"**,**"FASHION安心钱包计划"**}; **banner**.setBannerTitles(Arrays.*asList*(titles)); *//设置自动轮播，默认为true* **banner**.isAutoPlay(**true**); *//设置轮播时间* **banner**.setDelayTime(1500); *//设置指示器位置（当banner模式中有指示器时）* **banner**.setIndicatorGravity(BannerConfig.***CENTER***); *//banner设置方法全部调用完毕时最后调用* **banner**.start(); |

其中，

|  |
| --- |
| **public class** GlideImageLoader **implements** ImageLoader {   @Override  **public void** displayImage(Context context, Object path, ImageView imageView) {  Picasso.*with*(context).load((String)path).into(imageView);  } } |

### 功能八：自定义圆形进度条

#### 定义ui包，并提供自定义View：RoundProgress类

|  |
| --- |
| **public class** RoundProgress **extends** View {   **private int roundColor**;  **private int roundProgressColor**;  **private int textColor**;  **private float roundWidth**;  **private float textSize**;   *//自定义的属性* **private int max** = 100;  **private int viewWidth**;  **private** Paint **paint**;  **private int progress** = 50;   **public int** getMax() {  **return max**;  }   **public void** setMax(**int** max) {  **this**.**max** = max;  }   **public int** getProgress() {  **return progress**;  }   **public void** setProgress(**int** progress) {  **this**.**progress** = progress;  }   **public** RoundProgress(Context context) {  **this**(context, **null**);  }    **public** RoundProgress(Context context, AttributeSet attrs) {  **this**(context, attrs, 0);  }   **public** RoundProgress(Context context, AttributeSet attrs, **int** defStyleAttr) {  **super**(context, attrs, defStyleAttr);  *//读取自定义属性* TypedArray typedArray = context.obtainStyledAttributes(attrs, R.styleable.***RoundProgress***);  **roundColor** = typedArray.getColor(R.styleable.***RoundProgress\_roundColor***, Color.***RED***);  **roundProgressColor** = typedArray.getColor(R.styleable.***RoundProgress\_roundProgressColor***, Color.***GREEN***);  **textColor** = typedArray.getColor(R.styleable.***RoundProgress\_textColor***, Color.***GREEN***);  **roundWidth** = typedArray.getDimension(R.styleable.***RoundProgress\_roundWidth***, UIUtils.*dp2px*(10));  **textSize** = typedArray.getDimension(R.styleable.***RoundProgress\_textSize***, UIUtils.*dp2px*(20));   typedArray.recycle();   **paint** = **new** Paint();  **paint**.setAntiAlias(**true**);  }    @Override  **protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  **super**.onMeasure(widthMeasureSpec, heightMeasureSpec);  **viewWidth** = **this**.getMeasuredWidth();  }  @Override  **protected void** onDraw(Canvas canvas) {   *//画圆环* **paint**.setColor(**roundColor**);  **paint**.setStrokeWidth(**roundWidth**);  **paint**.setStyle(Paint.Style.***STROKE***);  canvas.drawCircle(**viewWidth**/2,**viewWidth**/2,**viewWidth**/2-**roundWidth**/2,**paint**);  *//画圆弧* RectF oval = **new** RectF(**roundWidth**/2,**roundWidth**/2,**viewWidth** - **roundWidth**/2,**viewWidth** - **roundWidth**/2);  **paint**.setColor(**roundProgressColor**);  canvas.drawArc(oval,0,**progress** \* 360 / **max**,**false**,**paint**);   *//画文本* **paint**.setTextSize(**textSize**);  **paint**.setColor(**textColor**);  **paint**.setStrokeWidth(0);  String text = **progress** \* 100 / **max** + **"%"**;  Rect bounds = **new** Rect();  **paint**.getTextBounds(text,0,text.length(),bounds);   canvas.drawText(text,**viewWidth**/2-bounds.width()/2,**viewWidth**/2+bounds.height()/2,**paint**);  }   } |

##### 其中，涉及到自定义视图属性

在res-values下新建attrs.xml文件，提供如下声明

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**resources**>  <**declare-styleable name="RoundProgress"**>  <**attr name="roundColor" format="color"**></**attr**>  <**attr name="roundProgressColor" format="color"**></**attr**>  <**attr name="textColor" format="color"**></**attr**>  <**attr name="roundWidth" format="dimension"**></**attr**>  <**attr name="textSize" format="dimension"**></**attr**>    </**declare-styleable**>   </**resources**> |

##### 接着，在fragment\_home.xml下使用自定义视图属性：

|  |
| --- |
| *<!-- 圆环进度-->* <**com.atguigu.p2p.ui.RoundProgress  android:id="@+id/p\_progresss"  android:layout\_width="120dp"  android:layout\_height="120dp"  app:roundColor="@android:color/darker\_gray"  app:roundProgressColor="@android:color/holo\_red\_dark"  app:roundWidth="10dp"  app:textColor="@color/text\_progress"  app:textSize="20sp"**> </**com.atguigu.p2p.ui.RoundProgress**>  要求：最外层布局中声明：  **xmlns:app="http://schemas.android.com/apk/res-auto"** |

#### 让当前的进度动起来：

##### 在HomeFragment类initData()的onSuccess()中：

|  |
| --- |
| **totalProgress** = Integer.*parseInt*(**index**.**product**.**progress**); **new** Thread(**runnable**).start(); |

###### 其中，totalProgress和runnable声明如下：

|  |
| --- |
| **private int totalProgress**;  **private** Runnable **runnable** = **new** Runnable() {  @Override  **public void** run() {  **pProgresss**.setProgress(0); *// pProgresss.setMax(100);//可修改max值* **for**(**int** i = 0; i < **totalProgress**; i++) {  **pProgresss**.setProgress(**pProgresss**.getProgress() + 1);  SystemClock.*sleep*(30);   **pProgresss**.postInvalidate();  }  }  }; |

### 功能九：提供自定义ScrollView，实现头部、尾部的下拉、上拉效果

#### ui包下创建类:MyScrollView

|  |
| --- |
| **public class** MyScrollView **extends** ScrollView {  **public** MyScrollView(Context context, AttributeSet attrs) {  **super**(context, attrs); }  } |

在fragment\_home.xml中替换<ScrollView>标签的使用

##### 回来设置MyScrollView的生命周期方法：

|  |
| --- |
| **private** View **innerView**;  **private** Rect **normal** = **new** Rect();  **private boolean animationFinish** = **true**;   @Override  **protected void** onFinishInflate() {  **super**.onFinishInflate();   **int** childCount = getChildCount();  **if** (childCount > 0) {  **innerView** = getChildAt(0);  }  }   **int lastY**,**downY**;  **int lastX**,**downX**;     @Override  **public boolean** onTouchEvent(MotionEvent ev) {  **if**(**innerView** == **null** || !**animationFinish**){  **return super**.onTouchEvent(ev);  }   **int** eventY = (**int**) ev.getY();  **switch** (ev.getAction()) {  **case** MotionEvent.***ACTION\_DOWN*** :  **downY** = **lastY** = eventY;  **break**;   **case** MotionEvent.***ACTION\_MOVE***:  **int** dy = eventY - **lastY**;   *//补充：可限制上滑、下滑的移动范围 // int totalY = eventY - downY; // if(totalY > UIUtils.dp2px(120) || totalY < -UIUtils.dp2px(120)){ // break; // }  // Log.e("TAG",innerView.getMeasuredHeight() + ":" + innerView.getHeight()); // Log.e("TAG",this.getHeight() + "");  //操作view进行拖动detailY的一半* **if**(isNeedMove()){  *//布局改变位置之前，记录一下正常状态的位置* **if** (**normal**.isEmpty()) {  **normal**.set(**innerView**.getLeft(), **innerView**.getTop(), **innerView**.getRight(), **innerView**.getBottom());  }  **innerView**.layout(**innerView**.getLeft(), **innerView**.getTop() + dy / 2, **innerView**.getRight(), **innerView**.getBottom() + dy / 2);  }   **lastY** = eventY;  **break**;   **case** MotionEvent.***ACTION\_UP***:  *//布局回滚到原来的位置* **if** (isNeedAnimation()) {  animation();  }  **break**;  }   **return super**.onTouchEvent(ev);  }   **private void** animation() {  TranslateAnimation animation = **new** TranslateAnimation(0, 0, 0, **normal**.**top** - **innerView**.getTop());  animation.setDuration(300);   animation.setAnimationListener(**new** Animation.AnimationListener() {  @Override  **public void** onAnimationStart(Animation animation) {  **animationFinish** = **false**;  }   @Override  **public void** onAnimationEnd(Animation animation) {  **innerView**.clearAnimation();  **innerView**.layout(**normal**.**left**, **normal**.**top**, **normal**.**right**, **normal**.**bottom**);  **normal**.setEmpty();  **animationFinish** = **true**;  }   @Override  **public void** onAnimationRepeat(Animation animation) {   }  });   **innerView**.startAnimation(animation);  }   */\*\*  \* 判断是否需要回滚  \*  \** ***@return*** *\*/* **private boolean** isNeedAnimation() {  **return** !**normal**.isEmpty();  }   **private boolean** isNeedMove() {  **int** measuredHeight = **innerView**.getMeasuredHeight();  **int** dy = measuredHeight - **this**.getHeight();  **int** scrollY = getScrollY();  *//左加右减，上加下减* **if**(scrollY == 0 || scrollY == dy){  **return true**;  }**else**{  **return false**;  }     } |

#### 运行发现，当down在ViewPager上时，滑动效果有异常情况。故：

|  |
| --- |
| @Override **public boolean** onInterceptTouchEvent(MotionEvent ev) {  **boolean** isIntercept = **false**;  **int** eventX = (**int**) ev.getX();  **int** eventY = (**int**) ev.getY();  **switch** (ev.getAction()) {  **case** MotionEvent.***ACTION\_DOWN*** :  **lastX** = **downX** = eventX;  **lastY** = **downY** = eventY;  **break**;  **case** MotionEvent.***ACTION\_MOVE***:  **int** totalX = Math.*abs*(eventX - **downX**);  **int** totalY = Math.*abs*(eventY - **downY**);  Log.*e*(**"TAG"**, **"totalX = "** + totalX + **",totalY = "** + totalY);  **if**(totalY > totalX && totalY > 6){  isIntercept = **true**;  }  **lastX** = eventX;  **lastY** = eventY;  **break**;   }   **return** isIntercept; } |

### 功能十：Fragment的抽取

#### 创建4个Fragment的公共的父类：在common包下创建BaseFragment

|  |
| --- |
| **public abstract class** BaseFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = UIUtils.*getView*(getLayoutid());  ButterKnife.*bind*(**this**,view);  initTitle();  initData();  **return** view;  }   **protected abstract void** initData();   **protected abstract void** initTitle();   **public abstract int** getLayoutid();   @Override  **public void** onDestroyView() {  **super**.onDestroyView();  ButterKnife.*unbind*(**this**);  } } |

然后修改HomeFragment、InvestFragment、MeFragment、MoreFragment的父类为BaseFragment。实现代码的抽取。以HomeFragment为例：

|  |
| --- |
| **public class** HomeFragment1 **extends** BaseFragment {  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  *//需要的话，再调整TextView的显示* }  @Override **public int** getLayoutid() {  **return** R.layout.***fragment\_home***; }  **public void** initData() {  //具体代码不修改  }  //删除原有的onDestroyView()方法。其它操作不改变。 |

另外三个Fragment处理类似，以InvestFragment为例：

|  |
| --- |
| **public class** InvestFragment **extends** BaseFragment {  @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;   @Override  **protected int** getLayoutId() {  **return** R.layout.***fragment\_invest***;  }   @Override  **protected void** initData(String content) {   }   @Override  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  **tvTitle**.setText(**"我要投资"**);  } } |

### 功能十一：联网加载数据的4种状态的抽取及代码优化

#### 在ui包下创建类LoadingPage：

|  |
| --- |
| **public abstract class** LoadingPage **extends** FrameLayout {    *//记录4种不同的显示状态* **private static final int *STATE\_LOADING*** = 1;  **private static final int *STATE\_ERROR*** = 2;  **private static final int *STATE\_EMPTY*** = 3;  **private static final int *STATE\_SUCCESS*** = 4;  *//当前状态* **private int state\_current** = 1;  *//加载4种不同的界面* **private** View **loadingView**;  **private** View **errorView**;  **private** View **emptyView**;  **private** View **successView**;  **private** LayoutParams **params**;   AsyncHttpClient **client** = **new** AsyncHttpClient();   **private** ResultState **resultState** = **null**;   **public** LoadingPage(Context context) {  **this**(context, **null**);  }   **public** LoadingPage(Context context, AttributeSet attrs) {  **this**(context, attrs, 0);  }   **public** LoadingPage(Context context, AttributeSet attrs, **int** defStyleAttr) {  **super**(context, attrs, defStyleAttr);   init();  }  **private void** init() {  *//视图与布局的关联* **params** = **new** LayoutParams(ViewGroup.LayoutParams.***MATCH\_PARENT***, ViewGroup.LayoutParams.***MATCH\_PARENT***);  **if** (**loadingView** == **null**) {  **loadingView** = UIUtils.*getView*(R.layout.***page\_loading***);  addView(**loadingView**, **params**);  }  **if** (**errorView** == **null**) {  **errorView** = UIUtils.*getView*(R.layout.***page\_error***);  addView(**errorView**, **params**);  }  **if** (**emptyView** == **null**) {  **emptyView** = UIUtils.*getView*(R.layout.***page\_empty***);  addView(**emptyView**, **params**);  }  *//加载布局的显示* showSafePage();   }  **private void** showSafePage() {  *//需要主线程执行* UIUtils.*runOnUiThread*(**new** Runnable() {   @Override  **public void** run() {  showPage();  }  });  }   **private void** showPage() {  **loadingView**.setVisibility(**state\_current** == ***STATE\_LOADING*** ? View.***VISIBLE*** : View.***GONE***);  **errorView**.setVisibility(**state\_current** == ***STATE\_ERROR*** ? View.***VISIBLE*** : View.***GONE***);  **emptyView**.setVisibility(**state\_current** == ***STATE\_EMPTY*** ? View.***VISIBLE*** : View.***GONE***);   **if** (**successView** == **null**) {  **successView** = UIUtils.*getView*(layoutId());  addView(**successView**, **params**);  }  **successView**.setVisibility(**state\_current** == ***STATE\_SUCCESS*** ? View.***VISIBLE*** : View.***GONE***);  }  **public void** show() {  *//初始化当前状态* **if** (**state\_current** != ***STATE\_LOADING***) {  **state\_current** = ***STATE\_LOADING***;  }   String url = url();  Log.*e*(**"TAG"**, **"url = "** + url);  **if** (TextUtils.*isEmpty*(url)) {  **resultState** = ResultState.***SUCCESS***;  **resultState**.setContent(**""**);  loadPage();  } **else** {  **client**.get(url, params(), **new** AsyncHttpResponseHandler() {  @Override  **public void** onSuccess(String content) {  **if** (TextUtils.*isEmpty*(content)) { *// state\_current = STATE\_EMPTY; // showSafePage();* **resultState** = ResultState.***EMPTY***;  **resultState**.setContent(**""**);  } **else** { *// state\_current = STATE\_SUCCESS; // showSafePage();* **resultState** = ResultState.***SUCCESS***;  **resultState**.setContent(content);   Log.*e*(**"TAG"**, **"content = "** + content);  }   loadPage();  }   @Override  **public void** onFailure(Throwable error, String content) { *// state\_current = STATE\_ERROR; // showSafePage();* **resultState** = ResultState.***ERROR***;  **resultState**.setContent(**""**);   loadPage();  }   });  }  }   **private void** loadPage() {  Log.*e*(**"TAG"**, **"resultState = "** + **resultState**);  **switch** (**resultState**) {  **case *ERROR***:  **state\_current** = ***STATE\_ERROR***;  **break**;  **case *EMPTY***:  **state\_current** = ***STATE\_EMPTY***;  **break**;  **case *SUCCESS***:  **state\_current** = ***STATE\_SUCCESS***;  **break**;  }  showSafePage();   **if** (**state\_current** == ***STATE\_SUCCESS***) {  onSuccess(**resultState**, **successView**);  }  }  *//内部类：保存联网状态以及返回的数据* **public enum** ResultState {  ***ERROR***(2), ***EMPTY***(3), ***SUCCESS***(4);   **private int state**;  **private** String **content**;   **public** String getContent() {  **return content**;  }   **public void** setContent(String content) {  **this**.**content** = content;  }   ResultState(**int** state) {  **this**.**state** = state;  }   }   **protected abstract** RequestParams params();   **protected abstract** String url();   **protected abstract void** onSuccess(ResultState resultState, View successView);   **public abstract int** layoutId();  } |

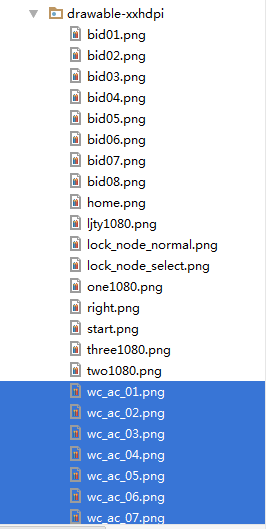
需要提供对应的page\_error.xml、page\_empty.xml和page\_loading.xml的布局文件。以及必要的图片。直接从现有的项目中拷贝过来。

##### 设置加载页面“加载图标”的动画：

在res-layout的page\_loading.xml中操作：

|  |
| --- |
| <**ProgressBar  android:layout\_width="100dp"  android:layout\_height="110dp"  style="?android:progressBarStyleInverse"  android:indeterminateDrawable="@drawable/loading\_list"** /> |

其中，drawable-xhdpi下提供：



其中，在drawable下提供animation-list的drawable动画

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**animation-list xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="false"**>   <**item android:drawable="@drawable/wc\_ac\_01" android:duration="30"**/>  <**item android:drawable="@drawable/wc\_ac\_02" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_03" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_04" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_05" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_06" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_07" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_08" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_09" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_10" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_11" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_12" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_13" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_14" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_15" android:duration="30"** />  <**item android:drawable="@drawable/wc\_ac\_16" android:duration="30"** /> </**animation-list**> |

##### 对应的UIUtils类添加必要的方法：

|  |
| --- |
| *//主线程执行的操作* **public static void** runOnUiThread(Runnable runnable){  **if**(*isMainThread*()){  runnable.run();  }**else**{  *getHandler*().post(runnable);  } }  **private static boolean** isMainThread() {  **int** myTid = android.os.Process.*myTid*();  **if**(myTid == MyApplication.*mainThreadId*){  **return true**;  }  **return false**;  } |

#### 对应的修改BaseFragment，在类中调用LoadingPage：

|  |
| --- |
| **public abstract class** BaseFragment **extends** Fragment {   **private** LoadingPage **loadingPage**;   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  **loadingPage** = **new** LoadingPage(container.getContext()){   @Override  **protected** RequestParams params() {  **return** getParams();  }   @Override  **protected** String url() {  **return** getUrl();  }   @Override  **protected void** onSuccess(ResultState resultState, View successView) {  ButterKnife.*bind*(BaseFragment.**this**,successView);  initTitle();  initData(resultState.getContent());  Log.*e*(**"TAG"**, **"content:"** + resultState.getContent());  }   @Override  **public int** layoutId() {   **return** getLayoutId();  }  };    *// View view = UIUtils.getView(getLayoutId()); // ButterKnife.bind(this, view); // initTitle(); // initData();* **return loadingPage**;  }  @Override **public void** onActivityCreated(@Nullable Bundle savedInstanceState) {  **super**.onActivityCreated(savedInstanceState);  UIUtils.*getHandler*().postDelayed(**new** Runnable() {  @Override  **public void** run() {  show();  }  },2000);  }  **public void** show(){  Log.*e*(**"TAG"**, (**loadingPage** == **null**) + **""**);  **loadingPage**.show();  }  *//获取联网地址* **protected abstract** String getUrl();  *//获取请求参数* **protected abstract** RequestParams getParams();  *//获取加载的布局* **protected abstract int** getLayoutId();  *//加载数据* **protected abstract void** initData(String content);  *//初始化title* **public abstract void** initTitle();   @Override  **public void** onDestroyView() {  **super**.onDestroyView();  ButterKnife.*unbind*(**this**);  } } |

#### 对应的HomeFragment也修改为如下：

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/8/2 0002.  \*/* **public class** HomeFragment **extends** BaseFragment {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***vp\_barner***)  ViewPager **vpBarner**;  @Bind(R.id.***circle\_barner***)  CirclePageIndicator **circleBarner**;  @Bind(R.id.***textView1***)  TextView **textView1**;  @Bind(R.id.***p\_progresss***)  RoundProgress **pProgresss**;  @Bind(R.id.***p\_yearlv***)  TextView **pYearlv**;  @Bind(R.id.***button1***)  Button **button1**;  @Bind(R.id.***myscrollview***)  MyScrollView **myscrollview**;   **private** Index **index**;  **private int totalProgress**;  *//进度条的加载* **private** Runnable **runnable** = **new** Runnable() {  @Override  **public void** run() {  **pProgresss**.setProgress(0); *// pProgresss.setMax(100);//可修改max值* **for** (**int** i = 0; i < **totalProgress**; i++) {  **pProgresss**.setProgress(**pProgresss**.getProgress() + 1);  SystemClock.*sleep*(30);   **pProgresss**.postInvalidate();  }  }  };    @Override  **protected** String getUrl() {  **return** AppNetConfig.***INDEX***;  }   @Override  **protected** RequestParams getParams() {  **return new** RequestParams();  }   @Override  **protected int** getLayoutId() {  **return** R.layout.***fragment\_home***;  }   @Override  **protected void** initData(String content) {  **if** (!TextUtils.*isEmpty*(content)) {  **index** = **new** Index();  JSONObject jsonObject = JSONObject.*parseObject*(content);   String proInfo = jsonObject.getString(**"proInfo"**);  Product product = JSON.*parseObject*(proInfo, Product.**class**);   String imageArr = jsonObject.getString(**"imageArr"**);  List<Image> imageList = JSON.*parseArray*(imageArr, Image.**class**);   **index**.**product** = product;  **index**.**imageList** = imageList;   *//适配数据* MyAdapter myAdapter = **new** MyAdapter();  **vpBarner**.setAdapter(myAdapter);  *//设置关联的ViewPager* **circleBarner**.setViewPager(**vpBarner**);   **totalProgress** = Integer.*parseInt*(**index**.**product**.**progress**);  **new** Thread(**runnable**).start();  }   }   @Override  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  *//需要的话，再调整TextView的显示* }    **class** MyAdapter **extends** PagerAdapter {    @Override  **public int** getCount() {  **return index**.**imageList** == **null** ? 0 : **index**.**imageList**.size();  }   @Override  **public boolean** isViewFromObject(View view, Object object) {  **return** view == object;  }   @Override  **public** Object instantiateItem(ViewGroup container, **int** position) {  ImageView imageView = **new** ImageView(getActivity());  imageView.setScaleType(ImageView.ScaleType.***CENTER\_CROP***);   Picasso.*with*(getActivity()).load(**index**.**imageList**.get(position).**IMAURL**).into(imageView);   container.addView(imageView);  **return** imageView;  }   @Override  **public void** destroyItem(ViewGroup container, **int** position, Object object) {  container.removeView((View) object);  }  }  } |

#### 对应的另外3个Fragment也需要调整:

继承父类中的抽象方法重写，操作类似。以InvestFragment为例：

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/8/2   \*/* **public class** InvestFragment **extends** BaseFragment {  @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;   @Override  **protected** String getUrl() {  **return null**;  }   @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected int** getLayoutId() {  **return** R.layout.***fragment\_invest***;  }   @Override  **protected void** initData(String content) {   }   @Override  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  **tvTitle**.setText(**"我要投资"**);  } } |

### 功能十二：InvestFragment整体界面的设置

#### InvestFragment设置如下：

|  |
| --- |
| **public class** InvestFragment **extends** BaseFragment {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***tab\_indictor***)  TabPageIndicator **tabIndictor**;  @Bind(R.id.***pager***)  ViewPager **pager**;  **private** List<Fragment> **fragmentList** = **new** ArrayList<Fragment>();  **private** MyAdapter **adapter**;   @Override  **protected** String getUrl() {  **return null**;  }   @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected int** getLayoutId() {  **return** R.layout.***fragment\_invest***;  }   @Override  **protected void** initData(String content) {  initFragmentList();   **adapter** = **new** MyAdapter(getFragmentManager());  **pager**.setAdapter(**adapter**);   **tabIndictor**.setViewPager(**pager**);  }   **private void** initFragmentList() {  ProductListFragment productListFragment = **new** ProductListFragment();  ProductRecommondFragment productRecommondFragment = **new** ProductRecommondFragment();  HotProductFragment hotProductFragment = **new** HotProductFragment();   **fragmentList**.add(productListFragment);  **fragmentList**.add(productRecommondFragment);  **fragmentList**.add(hotProductFragment);  }   @Override  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View.***GONE***);  **tvTitle**.setText(**"我要投资"**);  }   **class** MyAdapter **extends** FragmentPagerAdapter {   **public** MyAdapter(FragmentManager fm) {  **super**(fm);  }   @Override  **public** Fragment getItem(**int** position) {  **return fragmentList**.get(position);  }   @Override  **public int** getCount() {  **return fragmentList** == **null**? 0 : **fragmentList**.size();  }   @Override  **public** CharSequence getPageTitle(**int** position) {   **return** UIUtils.*getStringArr*(R.array.***touzi\_tab***)[position];  }  }  } |

#### 对应的布局为：

|  |
| --- |
| <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#00ff00"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>   <**com.viewpagerindicator.TabPageIndicator  android:id="@+id/tab\_indictor"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"**>  </**com.viewpagerindicator.TabPageIndicator**>   <**android.support.v4.view.ViewPager  android:id="@+id/pager"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>  </**android.support.v4.view.ViewPager**> </**LinearLayout**> |

#### 其中，需要创建内部的三个Fragment:

ProductListFragment、ProductRecommondFragment和HotProductFragment类：

##### 其中，ProductListFragment类：

|  |
| --- |
| **public class** ProductListFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getActivity(), R.layout.***fragment\_productlist***,**null**);   **return** view;  } } |

###### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="25sp"  android:textColor="@android:color/holo\_red\_dark"  android:text="全部理财"** />  </**LinearLayout**> |

##### 其中，ProductRecommondFragment类：

|  |
| --- |
| **public class** ProductRecommondFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getActivity(), R.layout.***fragment\_productrecommond***,**null**);   **return** view;  } } |

###### 对应的布局：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="25sp"  android:textColor="@android:color/holo\_red\_dark"  android:text="推荐理财"** />  </**LinearLayout**> |

##### 其中，HotProductFragment类：

|  |
| --- |
| **public class** HotProductFragment **extends** Fragment {   @Nullable  @Override  **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  View view = View.*inflate*(getActivity(), R.layout.***fragment\_producthot***,**null**);   **return** view;  } } |

###### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:textSize="25sp"  android:textColor="@android:color/holo\_red\_dark"  android:text="热门理财"** />  </**LinearLayout**> |

#### 接着，需要设置style.xml:

|  |
| --- |
| *<!--自定义viewpagerIndicator样式-->* <**style name="StyledIndicators" parent="@android:style/Theme.Light"**>  <**item name="vpiTabPageIndicatorStyle"**>@style/CustomTabPageIndicator</**item**>  <**item name="android:windowNoTitle"**>true</**item**> </**style**>  <**style name="CustomTabPageIndicator" parent="Widget.TabPageIndicator"**>  <**item name="android:background"**>@drawable/tab\_indicator</**item**>  <**item name="android:textAppearance"**>@style/CustomTabPageIndicator.Text</**item**>  <**item name="android:textSize"**>14sp</**item**>  <**item name="android:dividerPadding"**>8dp</**item**>  <**item name="android:showDividers"**>middle</**item**>  <**item name="android:paddingLeft"**>10dp</**item**>  <**item name="android:paddingRight"**>10dp</**item**>  <**item name="android:fadingEdge"**>horizontal</**item**>  <**item name="android:fadingEdgeLength"**>8dp</**item**> </**style**>  <**style name="CustomTabPageIndicator.Text" parent="android:TextAppearance.Medium"**>  <**item name="android:typeface"**>monospace</**item**>  <**item name="android:textColor"**>@drawable/selector\_tabtext</**item**> </**style**> |

##### 其中，用到的drawable资源如下：

selector\_tabtext.xml 和 tab\_indicator.xml

###### 【selector\_tabtext.xml】

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  <**item android:state\_selected="true" android:color="#EE2C2C"**/>  <**item android:state\_pressed="true" android:color="#EE2C2C"**/>  <**item android:state\_focused="true" android:color="#EE2C2C"**/>  <**item android:color="@android:color/black"**/> </**selector**> |

###### 【tab\_indicator.xml】

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  <**item android:state\_selected="false" android:state\_pressed="false" android:drawable="@android:color/transparent"**/>  <**item android:state\_selected="false" android:state\_pressed="true" android:drawable="@android:color/transparent"**/>  <**item android:state\_selected="true" android:state\_pressed="false" android:drawable="@color/title\_text"**/>  <**item android:state\_selected="true" android:state\_pressed="true" android:drawable="@color/title\_text"**/> </**selector**> |

#### 接着在AndroidManifest.xml中使用样式：

|  |
| --- |
| <**application  android:name=".common.MyApplication"  android:allowBackup="true"  android:icon="@mipmap/ic\_launcher"  android:label="@string/app\_name"  android:supportsRtl="true"  android:theme="@style/StyledIndicators"**> |

#### 同时需要调整Context参数：

为了保证设置的theme有效，需要保证加载布局的Context为Activity实例，而非Application实例

|  |
| --- |
| **public abstract class** LoadingPage **extends** FrameLayout {  **private** Context **mContext**;  **public** LoadingPage(Context context, AttributeSet attrs, **int** defStyleAttr) {  **super**(context, attrs, defStyleAttr);  **this**.**mContext** = context;  init(); }  **private void** showPage() {  **loadingView**.setVisibility(**state\_current** == ***STATE\_LOADING*** ? View.***VISIBLE*** : View.***GONE***);  **errorView**.setVisibility(**state\_current** == ***STATE\_ERROR*** ? View.***VISIBLE*** : View.***GONE***);  **emptyView**.setVisibility(**state\_current** == ***STATE\_EMPTY*** ? View.***VISIBLE*** : View.***GONE***);   **if** (**successView** == **null**) { *// successView = UIUtils.getView(layoutId());* **successView** = View.*inflate*(**mContext**,layoutId(),**null**);  addView(**successView**, **params**);  }  **successView**.setVisibility(**state\_current** == ***STATE\_SUCCESS*** ? View.***VISIBLE*** : View.***GONE***);  } |

### 功能十三：ProductListFragment：ListView适配不同的item布局

#### ProductListFragment提供如下：

|  |
| --- |
| */\*\*  \* 关于ListView的使用：4个元素：①ListView ②集合数据 ③ Adapter ④ item layout  \*  \*/* **public class** ProductListFragment **extends** BaseFragment {  @Bind(R.id.***lv\_product\_list***)  ListView **lvProductList**;  **private** List<Product> **products**;   @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected** String getUrl() {  **return** AppNetConfig.***PRODUCT***;  }   @Override  **protected void** initData(String content) {*//content即为响应成功情况下，返回的product.json数据  //解析json数据* JSONObject jsonObject = JSON.*parseObject*(content);  **boolean** isSuccess = jsonObject.getBoolean(**"success"**);  **if**(isSuccess){  String data = jsonObject.getString(**"data"**);  *//解析得到集合数据* **products** = JSON.*parseArray*(data, Product.**class**);  }  *// //创建Adapter的实例 // ProductAdapter productAdapter = new ProductAdapter(products); // //显示列表 // lvProductList.setAdapter(productAdapter);   //通用方式一：暴露getView()供子类实现。开发中可以使用 // ProductAdapter1 productAdapter1 = new ProductAdapter1(products); // lvProductList.setAdapter(productAdapter1);   //通用方式二：并没有使用ViewHolder减少findViewById()执行次数。开发中不可取。 // ProductAdapter2 productAdapter2 = new ProductAdapter2(products); // lvProductList.setAdapter(productAdapter2);  //通用方式三：既使用了ViewHolder，同时实现了更简洁的抽取* ProductAdapter3 productAdapter3 = **new** ProductAdapter3(**products**);  **lvProductList**.setAdapter(productAdapter3);   }   @Override  **protected void** initTitle() {   }   @Override  **public int** getLayoutId() {  **return** R.layout.***fragment\_product\_list***;  } } |

#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   *<!--<TextView-->  <!--android:id="@+id/tv\_product\_title"-->  <!--android:layout\_width="wrap\_content"-->  <!--android:layout\_height="wrap\_content"-->  <!--android:ellipsize="marquee"-->  <!--android:singleLine="true"-->  <!--android:padding="5dp"-->  <!--android:text="硅谷金融壕运当头，首投返现最高188元"-->  <!--android:textColor="@android:color/holo\_red\_dark"-->  <!--android:textSize="20sp"/>-->* <**com.atguigu.p2pinvest0714.ui.MyTextView  android:id="@+id/tv\_product\_title"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:ellipsize="marquee"  android:singleLine="true"  android:padding="5dp"  android:text="硅谷金融壕运当头，首投返现最高188元"  android:textColor="@android:color/holo\_red\_dark"  android:textSize="20sp"**/>   <**ListView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:id="@+id/lv\_product\_list"  android:dividerHeight="10dp"  android:divider="@color/title\_text"** ></**ListView**>  </**LinearLayout**> |

#### ListView对应的item的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:app="http://schemas.android.com/apk/res-auto"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@android:color/white"  android:orientation="vertical"**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**/>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:gravity="center\_vertical"  android:orientation="vertical"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:padding="15dp"**>   <**TextView  android:id="@+id/p\_name"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:text="超级新手计划"  android:textColor="#333333"  android:textSize="16sp"**/>   <**ImageView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_gravity="center\_vertical"  android:src="@drawable/right"**/>  </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_marginLeft="15dp"  android:layout\_marginRight="15dp"  android:background="@color/divider\_line"**/>  </**LinearLayout**>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"  android:padding="15dp"**>    <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:orientation="vertical"**>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:orientation="horizontal"**>   <**TextView  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="left"  android:text="产品金额"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>   <**TextView  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="center"  android:text="预期年利率"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>   <**TextView  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="right"  android:text="锁定期"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="5dp"  android:orientation="horizontal"**>    <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="left"**>   <**TextView  android:id="@+id/p\_money"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="10"  android:textColor="@color/product\_red\_common"  android:textSize="14sp"  android:textStyle="bold"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="万"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="center"**>   <**TextView  android:id="@+id/p\_yearlv"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="8.00"  android:textColor="@color/product\_red\_common"  android:textSize="14sp"  android:textStyle="bold"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="%"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="right"**>   <**TextView  android:id="@+id/p\_suodingdays"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="30"  android:textColor="@color/product\_red\_common"  android:textSize="14sp"  android:textStyle="bold"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="天"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="5dp"  android:orientation="horizontal"**>   <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="left"**>   <**TextView  android:layout\_width="1dp"  android:layout\_height="12dp"  android:background="@color/divider\_line"**/>   <**TextView  android:id="@+id/p\_minzouzi"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="4dp"  android:drawableLeft="@drawable/manage\_list\_count"  android:drawablePadding="4dp"  android:text="100起"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="center"**>   <**TextView  android:layout\_width="1dp"  android:layout\_height="12dp"  android:background="@color/divider\_line"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="4dp"  android:drawableLeft="@drawable/manage\_list\_guarantee"  android:drawablePadding="4dp"  android:text="本息保障"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>   <**LinearLayout  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_weight="1"  android:gravity="right"**>   <**TextView  android:layout\_width="1dp"  android:layout\_height="12dp"  android:background="@color/divider\_line"**/>   <**TextView  android:id="@+id/p\_minnum"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="4dp"  android:drawableLeft="@drawable/manage\_list\_member"  android:drawablePadding="4dp"  android:text="100"  android:textColor="@color/product\_detail\_common"  android:textSize="12sp"**/>  </**LinearLayout**>  </**LinearLayout**>  </**LinearLayout**> |

#### 跑马灯效果的实现

方式一：在ProductListFragment中：

|  |
| --- |
| @Override  **protected void** initData(String content) { *// tv.setFocusable(true); // tv.setFocusableInTouchMode(true); // tv.requestFocus();* } |

方式二：在ui包下自定义控件：AlwaysTextView

|  |
| --- |
| */\*\*  \* Created by shkstart  \*/* **public class** AlwaysTextView **extends** TextView {   **public** AlwaysTextView(Context context) {  **super**(context);  }   **public** AlwaysTextView(Context context, AttributeSet attrs) {  **super**(context, attrs);  }   **public** AlwaysTextView(Context context, AttributeSet attrs, **int** defStyleAttr) {  **super**(context, attrs, defStyleAttr);  }   @Override  **public boolean** isFocused() {  **return true**;  } } |

然后在ProductListFragment中使用如上的TextView标签声明即可。

#### 希望Adapter中加载的Item有不同的样式

可以重写BaseAdapter中的两个方法：

|  |
| --- |
| @Override **public int** getItemViewType(**int** position) {  **if**(position == 0){  **return** 0;  }**else**{  **return** 1;  } }  */\*\*  \* 返回item类型的个数  \** ***@return*** *\*/* @Override **public int** getViewTypeCount() {  **return** 2; } |

修改另外两个方法：

|  |
| --- |
| @Override **public int** getCount() {  **return productList** == **null** ? 0 : **productList**.size() + 1; }  @Override **public** View getView(**int** position, View convertView, ViewGroup parent) {  **int** itemViewType = getItemViewType(position);  **if** (itemViewType == 0) {   TextView tv = **new** TextView(getActivity());  tv.setText(**"不一样的烟火"**);  tv.setTextColor(Color.***RED***);  tv.setTextSize(UIUtils.*dp2px*(30));  **return** tv;   } **else** {  ViewHolder viewHolder;  **if** (convertView == **null**) {  convertView = View.*inflate*(getActivity(), R.layout.***item\_product\_list***, **null**);   viewHolder = **new** ViewHolder(convertView);  convertView.setTag(viewHolder);   } **else** {  viewHolder = (ViewHolder) convertView.getTag();  }   Product product = **productList**.get(position-1);  viewHolder.**pMinzouzi**.setText(product.**minTouMoney**);  viewHolder.**pMoney**.setText(product.**money**);  viewHolder.**pName**.setText(product.**name**);  viewHolder.**pProgresss**.setProgress(Integer.*parseInt*(product.**progress**));  viewHolder.**pSuodingdays**.setText(product.**suodingDays**);  viewHolder.**pYearlv**.setText(product.**yearLv**);  viewHolder.**pMinnum**.setText(product.**memberNum**);   **return** convertView;  } } |

### 功能十四：Adapter的抽取

#### 抽取方式一：

##### common包下提供类MyBaseAdapter:

|  |
| --- |
| **public abstract class** MyBaseAdapter<T> **extends** BaseAdapter {   **public** List<T> **data**;  **public** MyBaseAdapter(List<T> data){  **this**.**data** = data;  }   @Override  **public int** getCount() {  **return data** == **null** ? 0 : **data**.size();  }   @Override  **public** Object getItem(**int** position) {  **return data**.get(position);  }   @Override  **public long** getItemId(**int** position) {  **return** position;  }   @Override  **public** View getView(**int** position, View convertView, ViewGroup parent) {  **return** getYourView(position,convertView,parent);  }   **public abstract** View getYourView(**int** position,View convertView,ViewGroup parent); } |

##### 在ProductListFragment中提供MyBaseAdapter的子类：

|  |
| --- |
| *//提供另一个Adapter* **class** MyAdapter2 **extends** MyBaseAdapter<Product>{   **public** MyAdapter2(List<Product> data) {  **super**(data);  }   @Override  **public** View getYourView(**int** position, View convertView, ViewGroup parent) {  ViewHolder viewHolder;  **if**(convertView == **null**){  convertView = View.*inflate*(getActivity(),R.layout.***item\_product\_list***,**null**);  viewHolder = **new** ViewHolder(convertView);  convertView.setTag(viewHolder);  }**else**{  viewHolder = (ViewHolder) convertView.getTag();  }  Product product = **data**.get(position);  viewHolder.**pMinzouzi**.setText(product.**minTouMoney**);  viewHolder.**pMoney**.setText(product.**money**);  viewHolder.**pName**.setText(product.**name**);  viewHolder.**pProgresss**.setProgress(Integer.*parseInt*(product.**progress**));  viewHolder.**pSuodingdays**.setText(product.**suodingDays**);  viewHolder.**pYearlv**.setText(product.**yearLv**);  viewHolder.**pMinnum**.setText(product.**memberNum**);   **return** convertView;  } } |

##### 然后使用：在onSuccess()回调中：

|  |
| --- |
| *//方式二：* MyAdapter2 adapter = **new** MyAdapter2(productList); **lv**.setAdapter(adapter); |

#### 抽取方式二：

##### 复制上一个MyBaseAdapter，得到MyBaseAdatper1

|  |
| --- |
| **public abstract class** MyBaseAdapter1<T> **extends** BaseAdapter {  **public** List<T> **data**;   **public** MyBaseAdapter1(List<T> data) {  **this**.**data** = data;  }   @Override  **public int** getCount() {  **return data** == **null** ? 0 : **data**.size();  }   @Override  **public** Object getItem(**int** position) {  **return data**.get(position);  }   @Override  **public long** getItemId(**int** position) {  **return** position;  }   @Override  **public** View getView(**int** position, View convertView, ViewGroup parent) {  T t = **data**.get(position);  ViewHolder viewHolder = **null**;  **if**(convertView==**null**){  convertView = initView();  viewHolder = **new** ViewHolder(convertView);  }**else**{  viewHolder = (ViewHolder) convertView.getTag();  }  *//设置数据* setData(convertView,t);  **return** convertView;  }   **protected abstract void** setData(View convertView, T t);   **protected abstract** View initView();   **static class** ViewHolder{   **public** ViewHolder(View convertView) {  convertView.setTag(**this**);  }  } } |

##### 在ProductListFragment中：提供MyBaseAdapter1的内部实现类：

|  |
| --- |
| *//提供第三种Adapter* **class** MyAdapter3 **extends** MyBaseAdapter1<Product>{   **public** MyAdapter3(List<Product> data) {  **super**(data);  }   @Override  **protected void** setData(View convertView, Product product) {  *//此时没有使用ViewHolder。所以待优化* ((TextView)convertView.findViewById(R.id.***p\_name***)).setText(product.**name**);  Log.*e*(**"TAG"**, **"setData()"**);  }   @Override  **protected** View initView() {  **return** View.*inflate*(getActivity(),R.layout.***item\_product\_list***,**null**);  } } |

##### 然后使用：在onSuccess()回调中：

|  |
| --- |
| *//方式三：* MyAdapter3 adapter = **new** MyAdapter3(productList); **lv**.setAdapter(adapter); |

#### 抽取方式三：

针对于上面的这种方式中没有体现ViewHolder的使用，所有继续优化。

##### 提供common下的BaseHolder类：

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/8/9 0009.  \*  \* getView的实现全部交给BaseHolder类去做  \*/* **public abstract class** BaseHolder<T> {   **private** View **rootView**;  **private** T **mData**;   **public** BaseHolder() {  **this**.**rootView** = initView();  **this**.**rootView**.setTag(**this**);  ButterKnife.*bind*(BaseHolder.**this**, **rootView**);  }  **public** T getData() {  **return mData**;  }   **public void** setData(T t) {  **this**.**mData** = t;  refreshView();  }   **public** View getRootView() {  **return rootView**;  }   **public abstract** View initView();   **protected abstract void** refreshView(); } |

##### 复制MyBaseAdapter1，得到MyBaseAdapter2，并修改getView()方法为如下：

|  |
| --- |
| @Override **public** View getView(**int** position, View convertView, ViewGroup parent) {  BaseHolder holder = **null**;  **if**(convertView==**null**){  holder = getHolder();  }**else**{  holder = (BaseHolder) convertView.getTag();  }  *//设置数据* holder.setData(**data**.get(position));  **return** holder.getRootView(); }  **protected abstract** BaseHolder getHolder(); |

##### 回到ProductListFragment中提供MyBaseAdapter2的子类以及BaseHolder的子类：

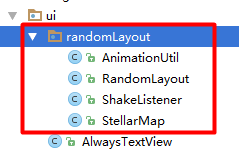
|  |
| --- |
| *//提供第四种Adapter* **class** MyAdapter4 **extends** MyBaseAdapter2<Product> {   **public** MyAdapter4(List<Product> data) {  **super**(data);  }   @Override  **protected** BaseHolder getHolder() {  **return new** MyViewHolder();  } }  **class** MyViewHolder **extends** BaseHolder<Product> {   @Bind(R.id.***p\_name***)  TextView **pName**;  @Bind(R.id.***p\_money***)  TextView **pMoney**;  @Bind(R.id.***p\_yearlv***)  TextView **pYearlv**;  @Bind(R.id.***p\_suodingdays***)  TextView **pSuodingdays**;  @Bind(R.id.***p\_minzouzi***)  TextView **pMinzouzi**;  @Bind(R.id.***p\_minnum***)  TextView **pMinnum**;  @Bind(R.id.***p\_progresss***)  RoundProgress **pProgresss**;   @Override  **public** View initView() {  **return** View.*inflate*(getActivity(), R.layout.***item\_product\_list***, **null**);  }   @Override  **protected void** refreshView() {  Product product = getData();  **pName**.setText(product.**name**);  **pMoney**.setText(product.**money**);  **pYearlv**.setText(product.**yearLv**);  **pSuodingdays**.setText(product.**suodingDays**);  **pMinzouzi**.setText(product.**minTouMoney**);  **pMinnum**.setText(product.**memberNum**);  **pProgresss**.setProgress(Integer.*parseInt*(product.**progress**));   } } |

##### 然后使用：在onSuccess()回调中：

|  |
| --- |
| *//方式四：* MyAdapter4 adapter = **new** MyAdapter4(productList); **lv**.setAdapter(adapter); |

### 功能十五：”推荐理财”的随机飞入效果

#### 导入randomLayout的包下的代码:



#### 加载其中StellarMap类在ProductRecommondFragment对应的布局中：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**com.atguigu.p2p.ui.randomLayout.StellarMap  android:id="@+id/stellarMap"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>    </**com.atguigu.p2p.ui.randomLayout.StellarMap**>  </**LinearLayout**> |

#### 回到ProductRecommondFragment类中：

|  |
| --- |
| **public class** ProductRecommondFragment **extends** BaseFragment {   @Bind(R.id.***stellarmap***)  StellarMap **stellarmap**;  **private** String[] **datas** = **new** String[]{**"超级新手计划"**, **"乐享活系列90天计划"**, **"钱包计划"**, **"30天理财计划(加息2%)"**, **"90天理财计划(加息5%)"**, **"180天理财计划(加息10%)"**,  **"林业局投资商业经营"**, **"中学老师购买车辆"**, **"屌丝下海经商计划"**, **"新西游影视拍摄投资"**, **"Java培训老师自己周转"**, **"养猪场扩大经营"**,  **"旅游公司扩大规模"**, **"阿里巴巴洗钱计划"**, **"铁路局回款计划"**, **"高级白领赢取白富美投资计划"** };   **private** String[] **firstDatas** = **new** String[**datas**.**length** / 2];  **private** String[] **secondDatas** = **new** String[**datas**.**length** - **datas**.**length** / 2];   **private** Random **random** = **new** Random();  @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected** String getUrl() {  **return null**;  }   @Override  **protected void** initData(String content) {  *//初始化数据* **for**(**int** i = 0; i < **datas**.**length**; i++) {  **if**(i < **datas**.**length** / 2){  **firstDatas**[i] = **datas**[i];  }**else**{  **secondDatas**[i - **datas**.**length** / 2] = **datas**[i];  }  }    StellarAdapter stellarAdapter = **new** StellarAdapter();  **stellarmap**.setAdapter(stellarAdapter);   **stellarmap**.setInnerPadding(10,10,10,10);   **stellarmap**.setGroup(0, **true**);  **stellarmap**.setRegularity(10,10);   }   @Override  **protected void** initTitle() {  }   @Override  **public int** getLayoutId() {  **return** R.layout.***fragment\_product\_recommond***;  }    **class** StellarAdapter **implements** StellarMap.Adapter {  *//显示组数* @Override  **public int** getGroupCount() {  **return** 2;  }  *//返回每组显示的个数* @Override  **public int** getCount(**int** group) {  **if**(group == 0){  **return datas**.**length** / 2;  }**else**{  **return datas**.**length** - **datas**.**length** / 2;  }  }  *//每组各个位置上显示的视图* @Override  **public** View getView(**int** group, **int** position, View convertView) {  Log.*e*(**"TAG"**, **"context:"** + getContext());  **final** TextView tv = **new** TextView(getActivity());  **if**(group == 0){  tv.setText(**firstDatas**[position]);  }**else**{  tv.setText(**secondDatas**[position]);  }  *//获取随机颜色* **int** red = **random**.nextInt(211);  **int** green = **random**.nextInt(211);  **int** blue = **random**.nextInt(211);  tv.setTextColor(Color.*rgb*(red,green,blue));  tv.setTextSize(**random**.nextInt(UIUtils.*dp2px*(10)) + 10);    *//设置监听* tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Toast.*makeText*(getContext(),tv.getText(), Toast.***LENGTH\_SHORT***).show();  }  });   **return** tv;  }  *//不调用* @Override  **public int** getNextGroupOnPan(**int** group, **float** degree) {  **return** 0;  }  *//下一组显示缩放动画的组别* @Override  **public int** getNextGroupOnZoom(**int** group, **boolean** isZoomIn) {  **if**(group == 0){  **return** 1;  }**else**{  **return** 0;  }  }  } } |

### 功能十六：”热门理财”的流式布局的实现

#### 创建ui包下的自定义视图：FlowLayout如下：

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/8/10 0010.  \*/* **public class** FlowLayout **extends** ViewGroup {  **public** FlowLayout(Context context) {  **super**(context);  }   **public** FlowLayout(Context context, AttributeSet attrs) {  **super**(context, attrs);  }   **public** FlowLayout(Context context, AttributeSet attrs, **int** defStyleAttr) {  **super**(context, attrs, defStyleAttr);  }   @Override  **protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  **int** widthMode = MeasureSpec.*getMode*(widthMeasureSpec);  **int** widthSize = MeasureSpec.*getSize*(widthMeasureSpec);  **int** heightMode = MeasureSpec.*getMode*(heightMeasureSpec);  **int** heightSize = MeasureSpec.*getSize*(heightMeasureSpec);   *//记录当前布局的宽高* **int** width = 0;  **int** height = 0;   *//记录一行视图的宽度、高度* **int** lineWidth = 0;  **int** lineHeight = 0;   *//获取在at\_most模式下的布局的宽度、高度* **int** childCount = getChildCount();  **for** (**int** i = 0; i < childCount; i++) {  View childView = getChildAt(i);   measureChild(childView, widthMeasureSpec, heightMeasureSpec);  **int** childWidth = childView.getMeasuredWidth();  **int** childHeight = childView.getMeasuredHeight();  MarginLayoutParams mp = ((MarginLayoutParams) childView.getLayoutParams());  *//表示当前的子视图还可以放在一行上* **if** (lineWidth + childWidth + mp.**leftMargin** + mp.**rightMargin** < widthSize) {  lineWidth += childWidth + mp.**leftMargin** + mp.**rightMargin**;   lineHeight = Math.*max*(lineHeight, childHeight + mp.**topMargin** + mp.**bottomMargin**);   } **else** {*//换行* width = Math.*max*(width, lineWidth);  height += lineHeight;   *//重置* lineWidth = childWidth + mp.**leftMargin** + mp.**rightMargin**;  lineHeight = childHeight + mp.**topMargin** + mp.**bottomMargin**;  }   **if** (i == childCount - 1) {  width = Math.*max*(width, lineWidth);  height += lineHeight;  }  }   Log.*e*(**"TAG"**, **"width = "** + width + **",height = "** + height);  setMeasuredDimension(widthMode == MeasureSpec.***EXACTLY*** ? widthSize : width, heightMode == MeasureSpec.***EXACTLY*** ? heightSize : height);  }   **private** List<List<View>> **allViews** = **new** ArrayList<List<View>>();  **private** List<Integer> **allHeights** = **new** ArrayList<>();   @Override  **protected void** onLayout(**boolean** changed, **int** left, **int** top, **int** right, **int** bottom) {  **int** width = getWidth();  **int** childCount = getChildCount();   **int** lineWidth = 0;  **int** lineHeight = 0;  List<View> lineViews = **new** ArrayList<>();   **for** (**int** i = 0; i < childCount; i++) {  View childView = getChildAt(i);   **int** childWidth = childView.getMeasuredWidth();  **int** childHeight = childView.getMeasuredHeight();  MarginLayoutParams mp = (MarginLayoutParams) childView.getLayoutParams();  *//不换行* **if** (lineWidth + childWidth + mp.**leftMargin** + mp.**rightMargin** < width) {  lineWidth += childWidth + mp.**leftMargin** + mp.**rightMargin**;  lineViews.add(childView);  lineHeight = Math.*max*(lineHeight, childHeight + mp.**topMargin** + mp.**bottomMargin**);   } **else** {*//换行* **allViews**.add(lineViews);  **allHeights**.add(lineHeight);  lineViews = **new** ArrayList<>();  lineViews.add(childView);   lineWidth = childWidth + mp.**leftMargin** + mp.**rightMargin**;  lineHeight = childHeight + mp.**topMargin** + mp.**bottomMargin**;  }   **if** (i == childCount - 1) {  **allViews**.add(lineViews);  **allHeights**.add(lineHeight);  }   }  Log.*e*(**"TAG"**, **"allViews.size = "** + **allViews**.size() + **",allHeights.size = "** + **allHeights**.size());   *//通过计算每一行的每一个子view的left,top,right,bottom,摆放每一行的每一个子view的位置* **int** x = 0;  **int** y = 0;   **for** (**int** i = 0; i < **allViews**.size(); i++) {  **int** curLineHeight = **allHeights**.get(i);  *//当前行的所有子view* List<View> views = **allViews**.get(i);  **for** (View view : views) {  **int** viewWidth = view.getMeasuredWidth();  **int** viewHeight = view.getMeasuredHeight();  MarginLayoutParams mp = (MarginLayoutParams) view.getLayoutParams();  **int** lc = x + mp.**leftMargin**;  **int** tc = y + mp.**topMargin**;  **int** rc = lc + viewWidth;  **int** bc = tc + viewHeight;  view.layout(lc, tc, rc, bc);  x += viewWidth + mp.**rightMargin** + mp.**leftMargin**;  }  x = 0;  y += curLineHeight;  }  }   *//有了如下的方法，在上面通过child就可以getLayoutParams()* @Override  **public** ViewGroup.LayoutParams generateLayoutParams(AttributeSet attrs) {  MarginLayoutParams mp = **new** MarginLayoutParams(getContext(), attrs);  **return** mp;  } } |

#### 此布局声明在ProductHotFragment对应的布局中：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**com.atguigu.p2p.ui.FlowLayout  android:id="@+id/flow\_hot"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:background="@android:color/holo\_green\_light"**>   </**com.atguigu.p2p.ui.FlowLayout**>  </**LinearLayout**> |

#### 对应的ProductHotFragment如下：

|  |
| --- |
| */\*\*  \* A simple {****@link*** *Fragment} subclass.  \*/* **public class** ProductHotFragment **extends** BaseFragment {    @Bind(R.id.***flow\_layout***)  FlowLayout **flowLayout**;   *//提供页面要显示的数据* **private** String[] **datas** = **new** String[]{**"新手计划"**, **"乐享活系列90天计划"**, **"钱包"**, **"30天理财计划(加息2%)"**,  **"林业局投资商业经营与大捞一笔"**, **"中学老师购买车辆"**, **"屌丝下海经商计划"**, **"新西游影视拍"**,  **"Java培训老师自己周转"**, **"HelloWorld"**, **"C++-C-ObjectC-java"**, **"Android vs ios"**, **"算法与数据结构"**, **"JNI与NDK"**, **"team working"**};  **private** Random **random**;   @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected** String getUrl() {  **return null**;  }   @Override  **protected void** initData(String content) {  **random** = **new** Random();  *//1.动态的创建TextView* **for**(**int** i = 0; i < **datas**.**length**; i++) {   **final** TextView tv = **new** TextView(getActivity());   *//设置TextView的属性* tv.setText(**datas**[i]);  tv.setTextSize(UIUtils.*dp2Px*(10));   ViewGroup.MarginLayoutParams mp = **new** ViewGroup.MarginLayoutParams(ViewGroup.LayoutParams.***WRAP\_CONTENT***,ViewGroup.LayoutParams.***WRAP\_CONTENT***);  mp.**leftMargin** = UIUtils.*dp2Px*(8);  mp.**rightMargin** = UIUtils.*dp2Px*(8);  mp.**topMargin** = UIUtils.*dp2Px*(8);  mp.**bottomMargin** = UIUtils.*dp2Px*(8);  tv.setLayoutParams(mp);   *//设置textView的背景* **int** red = **random**.nextInt(211);  **int** green = **random**.nextInt(211);  **int** blue = **random**.nextInt(211);  *//测试一： // tv.setBackground(DrawUtils.getDrawable(Color.rgb(red,green,blue),UIUtils.dp2Px(5)));   //测试二：* tv.setBackground(DrawUtils.*getSelector*(DrawUtils.*getDrawable*(Color.*rgb*(red, green, blue),UIUtils.*dp2Px*(5)),DrawUtils.*getDrawable*(Color.***WHITE***,UIUtils.*dp2Px*(5))));  *//保存按下能显示selector的效果，需要设置一个如下的属性 // tv.setClickable(true);   //添加点击事件，也是实现显示selector的效果的一种方式* tv.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Toast.*makeText*(ProductHotFragment.**this**.getActivity(), tv.getText(), Toast.***LENGTH\_SHORT***).show();  }  });   *//设置内边距* **int** padding = UIUtils.*dp2Px*(5);  tv.setPadding(padding,padding,padding,padding);   *// 2.添加到FlowLayout布局中* **flowLayout**.addView(tv);  }   }   @Override  **protected void** initTitle() {   }  @Override  **public int** getLayoutId() {  **return** R.layout.***fragment\_product\_hot***;  } } |

#### 其中，需要在util包下声明类：DrawableUtil

|  |
| --- |
| */\*\*  \* Created by shkstart.  \*/* **public class** DrawableUtil {   */\*\*  \* 定义一个shape资源  \*  \** ***@param rgb*** *\** ***@param corneradius*** *\** ***@return*** *\*/* **public static** GradientDrawable getDrawable(**int** rgb, **int** corneradius) {  GradientDrawable gradientDrawable = **new** GradientDrawable();  gradientDrawable.setColor(rgb);  gradientDrawable.setGradientType(GradientDrawable.***RECTANGLE***);  gradientDrawable.setCornerRadius(corneradius);  gradientDrawable.setStroke(UIUtils.*dp2px*(1), rgb);  **return** gradientDrawable;  }   **public static** StateListDrawable getSelector(Drawable normalDrawable,Drawable pressDrawable) {  StateListDrawable stateListDrawable = **new** StateListDrawable();  *//给当前的颜色选择器添加选中图片指向状态，未选中图片指向状态* stateListDrawable.addState(**new int**[]{android.R.attr.***state\_enabled***, android.R.attr.***state\_pressed***}, pressDrawable);  stateListDrawable.addState(**new int**[]{android.R.attr.***state\_enabled***}, normalDrawable);  *//设置默认状态* stateListDrawable.addState(**new int**[]{}, normalDrawable);  **return** stateListDrawable;  }  } |

### 功能十七：提供用户登录界面

#### MeFragment如下：

|  |
| --- |
| **public class** MeFragment **extends** BaseFragment {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***imageView1***)  ImageView **imageView1**;  @Bind(R.id.***icon\_time***)  RelativeLayout **iconTime**;  @Bind(R.id.***textView11***)  TextView **textView11**;  @Bind(R.id.***relativeLayout1***)  RelativeLayout **relativeLayout1**;  @Bind(R.id.***recharge***)  ImageView **recharge**;  @Bind(R.id.***withdraw***)  ImageView **withdraw**;  @Bind(R.id.***ll\_touzi***)  TextView **llTouzi**;  @Bind(R.id.***ll\_touzi\_zhiguan***)  TextView **llTouziZhiguan**;  @Bind(R.id.***ll\_zichang***)  TextView **llZichang**;     @Override  **protected** String getUrl() {  **return null**;  }   @Override  **protected** RequestParams getParams() {  **return null**;  }   @Override  **protected int** getLayoutId() {  **return** R.layout.***fragment\_me***;  }   @Override  **protected void** initData(String content) {  isLogin();  }   **private void** isLogin() {  SharedPreferences sp = getActivity().getSharedPreferences(**"user\_info"**, Context.***MODE\_PRIVATE***);  String name = sp.getString(**"name"**, **""**);  **if** (TextUtils.*isEmpty*(name)) {  *//未登录* showLoginDialog();  } **else** {  *//已登录--处理信息*  }  }   **private void** showLoginDialog() {  AlertDialog.Builder builder = **new** AlertDialog.Builder(getActivity());  builder.setTitle(**"登录"**);  builder.setMessage(**"必须先登录"**);  builder.setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) { *// Toast.makeText(MeFragment.this.getActivity(), "登录", Toast.LENGTH\_SHORT).show();* ((BaseActivity) getActivity()).gotoActivity(LoginActivity.**class**, **null**);  }  });  builder.setCancelable(**false**);  builder.create().show();  }   @Override  **public void** initTitle() {  **ivTitleBack**.setVisibility(View.***GONE***);  **ivTitleSetting**.setVisibility(View. VISIBLE);  **tvTitle**.setText(**"我的资产"**);  }   } |

#### 提供MeFragment对应的布局：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_top"** />   <**com.atguigu.p2p.ui.MyScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>    <**RelativeLayout  android:id="@+id/relativeLayout1"  android:layout\_width="match\_parent"  android:layout\_height="100dp"  android:background="@color/my\_title\_bg"  android:gravity="center"**>   <**RelativeLayout  android:id="@+id/icon\_time"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerHorizontal="true"**>   <**ImageView  android:layout\_width="64dp"  android:layout\_height="64dp"  android:scaleType="fitXY"  android:src="@drawable/my\_user\_bg\_icon"** />   <**ImageView  android:id="@+id/imageView1"  android:layout\_width="62dp"  android:layout\_height="62dp"  android:layout\_centerInParent="true"  android:scaleType="fitXY"  android:src="@drawable/my\_user\_default"** />  </**RelativeLayout**>   <**TextView  android:id="@+id/textView11"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_below="@id/icon\_time"  android:layout\_marginTop="10dp"  android:gravity="center"  android:text="Hi,wecome!"  android:textColor="@color/white"  android:textSize="14sp"** />  </**RelativeLayout**>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="10dp"  android:orientation="vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:text="学习提示:"  android:textColor="@android:color/holo\_red\_dark"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="5dp"  android:text="充值我们将以集成支付宝sdk支付金额演示充值功能"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="5dp"  android:text="提现业务:在p2p金融平台中,提现都不是实时到账的,而是审核提现,系统将会告知24小时内进行转账,如果未收到,请联系客服..."** />  </**LinearLayout**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_margin="10dp"  android:gravity="center"**>   <**ImageView  android:id="@+id/recharge"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_marginRight="9dp"  android:layout\_weight="1"  android:src="@drawable/my\_recharge\_bg"** />   <**ImageView  android:id="@+id/withdraw"  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="9dp"  android:layout\_weight="1"  android:src="@drawable/my\_withdraw\_bg"** />  </**LinearLayout**>   *<!--充分使用textview属性代替简单LinearLayout布局-->* <**TextView  android:id="@+id/ll\_touzi"  style="@style/LoginStyle"  android:drawableLeft="@drawable/my\_invest\_manage"  android:text="投资管理"** />   <**TextView  android:id="@+id/ll\_touzi\_zhiguan"  style="@style/LoginStyle"  android:drawableLeft="@drawable/my\_invest\_manage"  android:text="投资管理(直观)"** />   <**TextView  android:id="@+id/ll\_zichang"  style="@style/LoginStyle"  android:drawableLeft="@drawable/my\_asset\_manage"  android:text="资产管理"** />   </**LinearLayout**>  </**com.atguigu.p2p.ui.MyScrollView**>  </**LinearLayout**> |

##### 其中，使用了样式：

|  |
| --- |
| <**style name="LoginStyle"**>  <**item name="android:layout\_width"**>match\_parent</**item**>  <**item name="android:layout\_height"**>44dp</**item**>  <**item name="android:layout\_marginTop"**>1dp</**item**>  <**item name="android:background"**>@color/white</**item**>  <**item name="android:drawablePadding"**>8dp</**item**>  <**item name="android:drawableRight"**>@drawable/right</**item**>  <**item name="android:gravity"**>center\_vertical</**item**>  <**item name="android:paddingLeft"**>20dp</**item**>  <**item name="android:paddingRight"**>20dp</**item**>  <**item name="android:textColor"**>@color/product\_detail\_common</**item**>  <**item name="android:textSize"**>13sp</**item**> </**style**> |

#### 在common包下创建BaseActivity类：

（不需要在AndroidMenifest.xml中注册，也不需要布局文件）

|  |
| --- |
| **public abstract class** BaseActivity **extends** FragmentActivity {  **protected** AsyncHttpClient **client** = **new** AsyncHttpClient();  @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(getLayoutId());  ButterKnife.*bind*(**this**);   *//将当前的Activity添加到栈管理中* AppManager.*getInstance*().addActivity(**this**);   initTitle();  initData();  }   **protected abstract int** getLayoutId();   **protected abstract void** initData();   **protected abstract void** initTitle();   *//关闭当前activity* **public void** closeCurrent(){  AppManager.*getInstance*().removeCurrent();  }   *//保存用户信息的操作:使用sp存储* **public void** saveUser(User user){  SharedPreferences sp = **this**.getSharedPreferences(**"user\_info"**, Context.***MODE\_PRIVATE***);  SharedPreferences.Editor editor = sp.edit();  editor.putString(**"name"**,user.**name**);  editor.putString(**"imageUrl"**,user.**imageUrl**);  editor.putBoolean(**"isCredit"**, user.**isCredit**);  editor.putString(**"phone"**,user.**phone**);  editor.commit();*//只有提交以后，才可以创建此文件，并保存数据* }  *//读取数据，得到内存中的User对象* **public** User readUser(){  User user = **new** User();  SharedPreferences sp = **this**.getSharedPreferences(**"user\_info"**, Context.***MODE\_PRIVATE***);  user.**name** = sp.getString(**"name"**,**""**);  user.**imageUrl** = sp.getString(**"imageUrl"**,**""**);  user.**isCredit** = sp.getBoolean(**"isCredit"**,**false**);  user.**phone** = sp.getString(**"phone"**,**""**);   **return** user; }  */\*\*  \* 跳转到目标activity  \*  \** ***@param clazz*** *\** ***@param bundle*** *\*/* **public void** gotoActivity(Class clazz, Bundle bundle) {  Intent it = **new** Intent(**this**, clazz);  **if** (bundle != **null**) {  it.putExtra(**"param"**, bundle);  }  startActivity(it);  } |

#### 接着修改原来的MainActivity，修改其父类，已经修改相关的方法：

|  |
| --- |
| **public class** MainActivity **extends** BaseActivity {  @Override **protected int** getLayoutId() {  **return** R.layout.***activity\_main***; }  @Override **protected void** initTitle() {  }  //删除原有的onCreate()方法  } |

#### 在activity包下创建LoginActivity类：

|  |
| --- |
| **public class** LoginActivity **extends** BaseActivity {   @Bind(R.id.***iv\_top\_back***)  ImageView **ivTopBack**;  @Bind(R.id.***tv\_top\_title***)  TextView **tvTopTitle**;  @Bind(R.id.***iv\_top\_setting***)  ImageView **ivTopSetting**;  @Bind(R.id.***textView1***)  TextView **textView1**;  @Bind(R.id.***log\_ed\_mob***)  EditText **logEdMob**;  @Bind(R.id.***about\_com***)  RelativeLayout **aboutCom**;  @Bind(R.id.***tv\_2***)  TextView **tv2**;  @Bind(R.id.***log\_ed\_pad***)  EditText **logEdPad**;  @Bind(R.id.***log\_log\_btn***)  Button **logLogBtn**;   @Override  **protected void** initTitle() {  **ivTopBack**.setVisibility(View.***VISIBLE***);  **tvTopTitle**.setText(**"用户登录"**);  **ivTopSetting**.setVisibility(View.***INVISIBLE***);  }   @Override  **protected void** initData() {   }   @Override  **protected int** getLayoutId() {  **return** R.layout.***activity\_login***;  }   @OnClick(R.id.***iv\_top\_back***)  **public void** back(View view){ *// this.removeCurrentActivity();* **this**.removeAll();  goToActivity(MainActivity.**class**,**null**);  }   @OnClick(R.id.***log\_log\_btn***)  **public void** login(View view){  *//1.获取手机号和加密以后的密码* String number = **logEdMob**.getText().toString().trim();  String password = **logEdPad**.getText().toString().trim();  *//判断输入的信息是否存在空* **if**(!TextUtils.*isEmpty*(number) && !TextUtils.*isEmpty*(password)){  *//2.联网将用户数据发送给服务器，其中手机号和密码作为请求参数* String url = AppNetConfig.***LOGIN***;  RequestParams params = **new** RequestParams();  params.put(**"phone"**,number);  params.put(**"password"**, MD5Utils.*MD5*(password));  **client**.post(url,params,**new** AsyncHttpResponseHandler(){  *//3.得到响应数据：成功* @Override  **public void** onSuccess(String content) {  Log.*e*(**"TAG"**, content);  *//3.1解析json数据* JSONObject jsonObject = JSON.*parseObject*(content);  **boolean** isSuccess = jsonObject.getBoolean(**"success"**);  **if**(!isSuccess){  Toast.*makeText*(LoginActivity.**this**, **"用户名不存在或密码不正确"**, Toast.***LENGTH\_SHORT***).show();   }**else**{  String data = jsonObject.getString(**"data"**);  User user = JSON.*parseObject*(data, User.**class**);  *//3.2保存得到的用户信息（使用sp存储）* saveUser(user);  *//3.3重新加载页面，显示用户的信息在MeFragment中* LoginActivity.**this**.removeAll();  LoginActivity.**this**.goToActivity(MainActivity.**class**,**null**);  }   }  *//4.连接失败* @Override  **public void** onFailure(Throwable error, String content) {  Log.*e*(**"TAG"**, **"fail"**);  Toast.*makeText*(LoginActivity.**this**, **"联网失败"**, Toast.***LENGTH\_SHORT***).show();  }  });  }**else**{  Toast.*makeText*(LoginActivity.**this**, **"用户名或密码不能为空"**, Toast.***LENGTH\_SHORT***).show();  }  } } |

##### 对应的布局：activity\_login为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>   <**RelativeLayout  android:id="@+id/about\_com"  android:layout\_width="match\_parent"  android:layout\_height="44dp"  android:background="@color/white"  android:paddingLeft="17dp"  android:paddingRight="17dp"**>   <**TextView  android:id="@+id/textView1"  android:layout\_width="95dp"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:text="手机号:"  android:textColor="#333333"  android:textSize="14sp"**/>   <**EditText  android:id="@+id/log\_ed\_mob"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_centerVertical="true"  android:layout\_toRightOf="@id/textView1"  android:background="@null"  android:gravity="center\_vertical"  android:hint="请输入手机号"  android:inputType="number"  android:maxLines="1"  android:singleLine="true"  android:textColor="#5c5c5c"  android:textSize="14sp"**/>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="?android:listDivider"**/>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="44dp"  android:background="@color/white"  android:paddingLeft="17dp"  android:paddingRight="17dp"**>   <**TextView  android:id="@+id/tv\_2"  android:layout\_width="95dp"  android:layout\_height="wrap\_content"  android:layout\_alignParentLeft="true"  android:layout\_centerVertical="true"  android:text="密码:"  android:textColor="#333333"  android:textSize="14sp"**/>   <**EditText  android:id="@+id/log\_ed\_pad"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_centerVertical="true"  android:layout\_toRightOf="@id/tv\_2"  android:background="@null"  android:hint="请输入密码"  android:inputType="textPassword"  android:maxLines="1"  android:singleLine="true"  android:textColor="#5c5c5c"  android:textSize="14sp"**/>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="?android:listDivider"**/>   <**Button  android:id="@+id/log\_log\_btn"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="15dp"  android:background="@drawable/btn\_01"  android:gravity="center"  android:text="登　录"  android:textColor="#ffffff"  android:textSize="18sp"**/> </**LinearLayout**> |

##### 其中，需要在bean包下提供类User:

|  |
| --- |
| */\*\*  \* Created by shkstart on 2016/11/16 0016.  \*/* **public class** User {  **public int id**;*//编号* **public** String **name**;*//姓名* **public** String **password**;*//密码* **public** String **phone**;*//手机号* **public** String **imageUrl**;*//头像地址* **public boolean isCredit**;*//是否公安部认证* **public** User() {  }   **public** User(**int** id, String name, String password, String phone, String imageUrl, **boolean** isCredit) {  **this**.**id** = id;  **this**.**name** = name;  **this**.**password** = password;  **this**.**phone** = phone;  **this**.**imageUrl** = imageUrl;  **this**.**isCredit** = isCredit;  } } |

##### 其中，util包下提供MD5Utils工具类：

|  |
| --- |
| **public class** MD5Utils {   **public static** String MD5(String sourceStr) {  String result = **""**;  **try** {  MessageDigest md = MessageDigest.*getInstance*(**"MD5"**);  md.update(sourceStr.getBytes());  **byte** b[] = md.digest();  **int** i;  StringBuffer buf = **new** StringBuffer(**""**);  **for** (**int** offset = 0; offset < b.**length**; offset++) {  i = b[offset];  **if** (i < 0)  i += 256;  **if** (i < 16)  buf.append(**"0"**);  buf.append(Integer.*toHexString*(i));  }  result = buf.toString();  } **catch** (NoSuchAlgorithmException e) {  System.***out***.println(e);  }  **return** result;  }  } |

### 功能十八：头像的圆形处理

#### 在MeFragment中提供登录方法如下：

|  |
| --- |
| **private void** isLogin() {  SharedPreferences sp = getActivity().getSharedPreferences(**"user\_info"**, Context.***MODE\_PRIVATE***);  String uf\_acc = sp.getString(**"UF\_ACC"**, **""**);  **if** (TextUtils.*isEmpty*(uf\_acc)) {  *//未登录* showLoginDialog();  } **else** {  *//已登录--处理信息* doUser();  } }  **private void** doUser() {  Login login = ((BaseActivity) getActivity()).getLogin();  **textView11**.setText(login.**UF\_ACC**);  Picasso.*with*(getActivity()).load(login.**UF\_AVATAR\_URL**).transform(**new** Transformation() {  @Override  **public** Bitmap transform(Bitmap source) {  Bitmap zoom = BitMapUtil.*zoom*(source, UIUtils.*dp2px*(62), UIUtils.*dp2px*(62));  Bitmap circleBitMap = BitMapUtil.*circleBitMap*(zoom);  *//1:transform当中处理完图片之后，需要调用recylce方法回收* source.recycle();  **return** circleBitMap;  }   @Override  **public** String key() {  *//2:重写key方法的返回值，不能是null* **return ""**;  }  }).into(**imageView1**); } |

#### 需要在util包下提供类BitMapUtil：

|  |
| --- |
| */\*\*  \* Created by shkstart.  \*/* **public class** BitMapUtil {   */\*\*  \* 图片缩放  \* wf.wh必须不能是int  \*  \** ***@param source*** *\** ***@param wf*** *\** ***@param hf*** *\** ***@return*** *\*/* **public static** Bitmap zoom(Bitmap source, **float** wf, **float** hf) {  Matrix matrix = **new** Matrix();  **float** sx = wf / source.getWidth();  **float** sy = hf / source.getHeight();  Log.*e*(**"zoubo"**, **"sx---:"** + sx);  matrix.postScale(sx, sy);  **return** Bitmap.*createBitmap*(source, 0, 0, source.getWidth(), source.getHeight(), matrix, **true**);  }    */\*\*  \* 头像圆形裁剪  \*  \** ***@param source*** *\** ***@return*** *\*/* **public static** Bitmap circleBitMap(Bitmap source) {  **final** Paint paint = **new** Paint();  *//抗锯齿效果* **int** width = source.getWidth();  paint.setAntiAlias(**true**);  *//指定大小bitmap* Bitmap target = Bitmap.*createBitmap*(width, width, Bitmap.Config.***ARGB\_8888***);  *//根据target生成一个画布* Canvas canvas = **new** Canvas(target);  *//在画布上画了一个圆  //参数CX,cy-->确定绘制圆的圆心点  //半径参数  //画笔* canvas.drawCircle(width / 2, width / 2, width / 2, paint);  *//这句话是关键:  //分析：我们以一张图片作为画布，在上面画了一个圆-->画图展示-->"这时候,绘制的圆和图片本身就出现了一个圆形的交集图案"  //setXfermode：设置当绘制的图像出现相交情况时候的处理方式的,它包含的常用模式有哪几种  //PorterDuff.Mode.SRC\_IN 取两层图像交集部门,只显示上层图像,注意这里是指取相交叉的部分,然后显示上层图像  //PorterDuff.Mode.DST\_IN 取两层图像交集部门,只显示下层图像* paint.setXfermode(**new** PorterDuffXfermode(PorterDuff.Mode.***SRC\_IN***));  *//使用设置了setXfermode方案的paint绘制图像* canvas.drawBitmap(source, 0, 0, paint);  **return** target;  } } |

### 功能十九：调用系统拍照、图库选择图片

#### 在MeFragment中提供回调方法：

|  |
| --- |
| @OnClick(R.id.***iv\_title\_setting***) **public void** clickUserSetting(View view) {  ((BaseActivity) getActivity()).gotoActivity(UserInfoActivity.**class**, **null**); } |

#### 在activity包下创建UserInfoActivity:

|  |
| --- |
| **public class** UserInfoActivity **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***imageView1***)  ImageView **imageView1**;  @Bind(R.id.***textView1***)  TextView **textView1**;  @Bind(R.id.***loginout***)  Button **loginout**;   @Override  **protected void** initData() {   }   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"用户信息"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }  @OnClick(R.id.***iv\_common\_back***)  **public void** back(View view){ *// closeCurrentActivity();* AppManager.*getInstance*().removeAll();  goToActivity(MainActivity.**class**,**null**);  }  @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_user\_info***;  }    **private static final int *CAMERA*** = 100;   **private static final int *PICTURE*** = 200;   *//更换头像的回调方法* @OnClick(R.id.***textView1***) **public void** changeIcon(View view) {   **new** AlertDialog.Builder(**this**)  .setTitle(**"选择来源"**)  .setItems(**new** String[]{**"拍照"**, **"图库"**}, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  **switch** (which) {  **case** 0:  *//打开系统拍照程序，选择拍照图片* Intent camera = **new** Intent(MediaStore.***ACTION\_IMAGE\_CAPTURE***);  startActivityForResult(camera, ***CAMERA***);  **break**;  **case** 1:  *//打开系统图库程序，选择图片* Intent picture = **new** Intent(Intent.***ACTION\_PICK***, MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***);  startActivityForResult(picture, ***PICTURE***);  **break**;  }  }  })  .show(); }  @Override **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  **super**.onActivityResult(requestCode, resultCode, data);   **if** (requestCode == ***CAMERA*** && resultCode == ***RESULT\_OK*** && data != **null**) {  *//拍照* Bundle bundle = data.getExtras();  *// 获取相机返回的数据，并转换为图片格式* Bitmap bitmap = (Bitmap) bundle.get(**"data"**);  *//bitmap圆形裁剪* bitmap = BitmapUtils.*zoom*(bitmap, UIUtils.*dp2px*(62), UIUtils.*dp2px*(62));  Bitmap circleImage = BitmapUtils.*circleBitmap*(bitmap);    *//真是项目当中，是需要上传到服务器的..这步我们就不做了。* **imageView1**.setImageBitmap(circleImage);  *//将图片保存在本地* saveImage(circleImage);   } **else if** (requestCode == ***PICTURE*** && resultCode == ***RESULT\_OK*** && data != **null**) {  *//图库* Uri selectedImage = data.getData();  Log.*e*(**"TAG"**, **"selectedImage = "** + selectedImage);  *//这里返回的uri情况就有点多了  //\*\*:在4.4.2之前返回的uri是:content://media/external/images/media/3951或者file://....在4.4.2返回的是content://com.android.providers.media.documents/document/image:3951或者  //总结：uri的组成，eg:content://com.example.project:200/folder/subfolder/etc  //content:--->"scheme"  //com.example.project:200-->"host":"port"--->"authority"[主机地址+端口(省略) =authority]  //folder/subfolder/etc-->"path" 路径部分  //android各个不同的系统版本,对于获取外部存储上的资源，返回的Uri对象都可能各不一样,所以要保证无论是哪个系统版本都能正确获取到图片资源的话  //就需要针对各种情况进行一个处理了* String pathResult = getPath(selectedImage);  Bitmap decodeFile = BitmapFactory.*decodeFile*(pathResult);  Bitmap zoomBitmap = BitmapUtils.*zoom*(decodeFile, UIUtils.*dp2px*(62), UIUtils.*dp2px*(62));  *//bitmap圆形裁剪* Bitmap circleImage = BitmapUtils.*circleBitmap*(zoomBitmap);  *//真是项目当中，是需要上传到服务器的..这步我们就不做了。* **imageView1**.setImageBitmap(circleImage);  saveImage(circleImage);  } }  */\*\*  \* 根据系统相册选择的文件获取路径  \*  \** ***@param uri*** *\*/* @SuppressLint(**"NewApi"**)  **private** String getPath(Uri uri) {  **int** sdkVersion = Build.VERSION.***SDK\_INT***;  *//高于4.4.2的版本* **if** (sdkVersion >= 19) {  Log.*e*(**"zoubo"**, **"uri auth: "** + uri.getAuthority());  **if** (isExternalStorageDocument(uri)) {  String docId = DocumentsContract.*getDocumentId*(uri);  String[] split = docId.split(**":"**);  String type = split[0];  **if** (**"primary"**.equalsIgnoreCase(type)) {  **return** Environment.*getExternalStorageDirectory*() + **"/"** + split[1];  }  } **else if** (*isDownloadsDocument*(uri)) {  **final** String id = DocumentsContract.*getDocumentId*(uri);  **final** Uri contentUri = ContentUris.*withAppendedId*(Uri.*parse*(**"content://downloads/public\_downloads"**),  Long.*valueOf*(id));  **return** *getDataColumn*(**this**, contentUri, **null**, **null**);  } **else if** (*isMediaDocument*(uri)) {  **final** String docId = DocumentsContract.*getDocumentId*(uri);  **final** String[] split = docId.split(**":"**);  **final** String type = split[0];   Uri contentUri = **null**;  **if** (**"image"**.equals(type)) {  contentUri = MediaStore.Images.Media.***EXTERNAL\_CONTENT\_URI***;  } **else if** (**"video"**.equals(type)) {  contentUri = MediaStore.Video.Media.***EXTERNAL\_CONTENT\_URI***;  } **else if** (**"audio"**.equals(type)) {  contentUri = MediaStore.Audio.Media.***EXTERNAL\_CONTENT\_URI***;  }  **final** String selection = **"\_id=?"**;  **final** String[] selectionArgs = **new** String[]{split[1]};  **return** *getDataColumn*(**this**, contentUri, selection, selectionArgs);  } **else if** (*isMedia*(uri)) {  String[] proj = {MediaStore.Images.Media.***DATA***};  Cursor actualimagecursor = **this**.managedQuery(uri, proj, **null**, **null**, **null**);  **int** actual\_image\_column\_index = actualimagecursor.getColumnIndexOrThrow(MediaStore.Images.Media.***DATA***);  actualimagecursor.moveToFirst();  **return** actualimagecursor.getString(actual\_image\_column\_index);  }  } **else if** (**"content"**.equalsIgnoreCase(uri.getScheme())) {  *// Return the remote address* **if** (*isGooglePhotosUri*(uri))  **return** uri.getLastPathSegment();  **return** *getDataColumn*(**this**, uri, **null**, **null**);  }  *// File* **else if** (**"file"**.equalsIgnoreCase(uri.getScheme())) {  **return** uri.getPath();  }  **return null**;  }   */\*\*  \* uri路径查询字段  \*  \** ***@param context*** *\** ***@param uri*** *\** ***@param selection*** *\** ***@param selectionArgs*** *\** ***@return*** *\*/* **public static** String getDataColumn(Context context, Uri uri, String selection, String[] selectionArgs) {  Cursor cursor = **null**;  **final** String column = **"\_data"**;  **final** String[] projection = {column};  **try** {  cursor = context.getContentResolver().query(uri, projection, selection, selectionArgs, **null**);  **if** (cursor != **null** && cursor.moveToFirst()) {  **final int** index = cursor.getColumnIndexOrThrow(column);  **return** cursor.getString(index);  }  } **finally** {  **if** (cursor != **null**)  cursor.close();  }  **return null**;  }    **private boolean** isExternalStorageDocument(Uri uri) {  **return "com.android.externalstorage.documents"**.equals(uri.getAuthority());  }   **public static boolean** isDownloadsDocument(Uri uri) {  **return "com.android.providers.downloads.documents"**.equals(uri.getAuthority());  }   **public static boolean** isMediaDocument(Uri uri) {  **return "com.android.providers.media.documents"**.equals(uri.getAuthority());  }   **public static boolean** isMedia(Uri uri) {  **return "media"**.equals(uri.getAuthority());  }   */\*\*  \** ***@param uri*** *The Uri to check.  \** ***@return*** *Whether the Uri authority is Google Photos.  \*/* **public static boolean** isGooglePhotosUri(Uri uri) {  **return "com.google.android.apps.photos.content"**.equals(uri.getAuthority());  }  } |

##### 其中，在当前UserInfoActivity中提供如下的saveImage()：

|  |
| --- |
| *//保存到本地* **public void** saveImage(Bitmap bitmap) {  String path = getCacheDir() + **"/tx.png"**;  Log.*e*(**"TAG"**, **"path = "** + path);  **try** {  FileOutputStream fos = **new** FileOutputStream(path);  *//bitmap压缩(压缩格式、质量、压缩文件保存的位置)* bitmap.compress(Bitmap.CompressFormat.***PNG***, 100, fos);   } **catch** (FileNotFoundException e) {  e.printStackTrace();  } } |

##### 接着MeFragment中提供onResume()的重写：

|  |
| --- |
| @Override **public void** onResume() {  **super**.onResume();  Log.*e*(**"TAG"**, **"onResume........"**);  *//读取本地的头像* String filePath = **this**.getActivity().getCacheDir() + **"/tx.png"**;  File file = **new** File(filePath);  **if**(file.exists()){  *//将文件中的图片数据加载到内存中，生成一个Bitmap对象* Bitmap bitmap = BitmapFactory.*decodeFile*(file.getAbsolutePath());  **imageView1**.setImageBitmap(bitmap);  **return** ;  } } |

##### 接着修改MeFragment中头像的显示的doUser()：

|  |
| --- |
| *//如果用户已经登录，显示登录信息* **private void** doUser() {  Login login = ((BaseActivity) **this**.getActivity()).getLogin();  **textView11**.setText(login.**UF\_ACC**);  *//读取本地的头像* String filePath = **this**.getActivity().getCacheDir() + **"/tx.png"**;  File file = **new** File(filePath);  *//防止下面通过Picasso分线程加载sp中的本地图片过慢，覆盖了CacheDir中的图片。* **if**(file.exists()){  *//将文件中的图片数据加载到内存中，生成一个Bitmap对象 // Bitmap bitmap = BitmapFactory.decodeFile(file.getAbsolutePath()); // imageView1.setImageBitmap(bitmap);* **return** ;  }  *//显示用户头像(下面还是原有的代码，不变)*  Picasso.*with*(getActivity()).load(login.**UF\_AVATAR\_URL**).transform(**new** Transformation() { |

##### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:orientation="vertical"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**include layout="@layout/common\_title"**/>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="#EDEDED"  android:orientation="vertical"**>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:paddingBottom="15dp"**>   <**ImageView  android:id="@+id/imageView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentTop="true"  android:layout\_centerHorizontal="true"  android:layout\_marginTop="10dp"  android:src="@drawable/my\_user\_default"**/>   <**TextView  android:id="@+id/textView1"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_below="@id/imageView1"  android:layout\_centerHorizontal="true"  android:layout\_centerVertical="true"  android:layout\_marginBottom="15dp"  android:text="更换头像"  android:textColor="#00baff"  android:textSize="14sp"**/>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="?android:listDivider"**/>   <**Button  android:id="@+id/loginOut"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginRight="20dp"  android:layout\_marginLeft="20dp"  android:layout\_marginTop="15dp"  android:background="@drawable/btn\_01"  android:gravity="center"  android:text="退出登录"  android:textColor="#ffffff"  android:textSize="15sp"**/>  </**LinearLayout**>  </**LinearLayout**> |

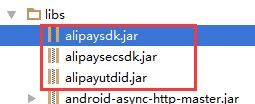
### 功能二十：退出登录

#### 在UserInfoActivity中提供退出登录的方法：

|  |
| --- |
| *//退出登录的回调* @OnClick(R.id.***loginout***) **public void** logout(View view) {  **new** AlertDialog.Builder(**this**)  .setTitle(**"退出登录"**)  .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  SharedPreferences sp = getSharedPreferences(**"user\_info"**, Context.***MODE\_PRIVATE***);  sp.edit().clear().commit();*//清除文件中的数据。   //删除本地图片文件* String filePath = UserInfoActivity.**this**.getCacheDir() + **"/tx.png"**;  File file = **new** File(filePath);  **if**(file.exists()){  file.delete();  }   *//移除栈中所有的activity* ActivityManager.*getInstance*().removeAll();  *//重新加载首页面* goToActivity(MainActivity.**class**, **null**);  }  })  .setNegativeButton(**"取消"**, **null**)  .show();  } |

### 功能二十一：集成支付宝功能

#### 导入支付宝相关的jar包，并同步



#### 在功能清单文件中注册如下activity，并导入必要的权限

##### 注册Activity：

|  |
| --- |
| *<!-- 集成支付宝组件声明-->* <**activity  android:name="com.alipay.sdk.app.H5PayActivity"  android:configChanges="orientation|keyboardHidden|navigation"  android:exported="false"  android:screenOrientation="behind"**> </**activity**> <**activity  android:name="com.alipay.sdk.auth.AuthActivity"  android:configChanges="orientation|keyboardHidden|navigation"  android:exported="false"  android:screenOrientation="behind"**> </**activity**> |

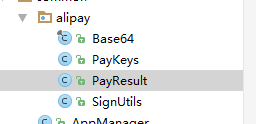
##### 权限：

|  |
| --- |
| *<!-- 配置所需权限 -->* <**uses-permission android:name="android.permission.RECEIVE\_USER\_PRESENT"** /> <**uses-permission android:name="android.permission.INTERNET"** /> <**uses-permission android:name="android.permission.WAKE\_LOCK"** /> <**uses-permission android:name="android.permission.READ\_PHONE\_STATE"** /> <**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** /> <**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** /> <**uses-permission android:name="android.permission.WRITE\_SETTINGS"** /> <**uses-permission android:name="android.permission.VIBRATE"** /> <**uses-permission android:name="android.permission.MOUNT\_UNMOUNT\_FILESYSTEMS"** /> <**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** /> <**uses-permission android:name="android.permission.SYSTEM\_ALERT\_WINDOW"** /> <**uses-permission android:name="android.permission.GET\_TASKS"** /> <**uses-permission android:name="android.permission.ACCESS\_WIFI\_STATE"** /> <**uses-permission android:name="android.permission.MANAGE\_ACCOUNTS"** /> <**uses-permission android:name="android.permission.GET\_ACCOUNTS"** /> <**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** /> <**uses-permission android:name="android.permission.RESTART\_PACKAGES"** /> |

#### 在activity下创建ChongzhiActivity：

|  |
| --- |
| */\*\*  \* Created by shkstart.  \*/* **public class** ChongzhiActivity **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***chongzhi\_text***)  TextView **chongzhiText**;  @Bind(R.id.***view***)  View **view**;  @Bind(R.id.***chongzhi\_et***)  EditText **chongzhiEt**;  @Bind(R.id.***chongzhi\_text2***)  TextView **chongzhiText2**;  @Bind(R.id.***yue\_tv***)  TextView **yueTv**;  @Bind(R.id.***chongzhi\_btn***)  Button **chongzhiBtn**;   @Override  **protected void** initData() {  **chongzhiEt**.addTextChangedListener(**watcher**);   }   **private** TextWatcher **watcher** = **new** TextWatcher() {  @Override  **public void** beforeTextChanged(CharSequence s, **int** start, **int** count, **int** after) {   }   @Override  **public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {   }   @Override  **public void** afterTextChanged(Editable s) {  String s1 = **chongzhiEt**.getText().toString();  **if** (!TextUtils.*isEmpty*(s1)) {  **chongzhiBtn**.setBackgroundResource(R.drawable.***btn\_01***);  } **else** {  **chongzhiBtn**.setBackgroundResource(R.drawable.***btn\_023***);  }  }  };   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"充值"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }   @OnClick(R.id.***iv\_title\_back***)  **public void** back(View view) {  closeCurrent();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_chongzhi***;  }   @OnClick(R.id.***chongzhi\_btn***)  **public void** clickZf(View view) {  *// 订单* String orderInfo = getOrderInfo(**"测试的商品"**, **"该测试商品的详细描述"**, **"0.01"**);  *// 对订单做RSA 签名* String sign = sign(orderInfo);  **try** {  *// 仅需对sign 做URL编码* sign = URLEncoder.*encode*(sign, **"UTF-8"**);  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }   *// 完整的符合支付宝参数规范的订单信息* **final** String payInfo = orderInfo + **"&sign=\""** + sign + **"\"&"** + getSignType();   *// 必须异步调用支付接口进行支付* Runnable payRunnable = **new** Runnable() {   @Override  **public void** run() {  *// 构造PayTask 对象* PayTask alipay = **new** PayTask(ChongzhiActivity.**this**);  *// 调用支付接口，获取支付结果* String result = alipay.pay(payInfo);  Message msg = **new** Message();  msg.**obj** = result;  **mHandler**.sendMessage(msg);  }  };  Thread payThread = **new** Thread(payRunnable);  payThread.start();  }   **private** Handler **mHandler** = **new** Handler() {  **public void** handleMessage(Message msg) {  PayResult payResult = **new** PayResult((String) msg.**obj**);   *// 支付宝返回此次支付结果及加签，建议对支付宝签名信息拿签约时支付宝提供的公钥做验签* String resultInfo = payResult.getResult();   String resultStatus = payResult.getResultStatus();   *// 判断resultStatus 为“9000”则代表支付成功，具体状态码代表含义可参考接口文档* **if** (TextUtils.*equals*(resultStatus, **"9000"**)) {  Toast.*makeText*(ChongzhiActivity.**this**, **"支付成功"**,  Toast.***LENGTH\_SHORT***).show();  } **else** {  *// 判断resultStatus 为非“9000”则代表可能支付失败  // “8000”代表支付结果因为支付渠道原因或者系统原因还在等待支付结果确认，最终交易是否成功以服务端异步通知为准（小概率状态）* **if** (TextUtils.*equals*(resultStatus, **"8000"**)) {  Toast.*makeText*(ChongzhiActivity.**this**, **"支付结果确认中"**,  Toast.***LENGTH\_SHORT***).show();   } **else** {  *// 其他值就可以判断为支付失败，包括用户主动取消支付，或者系统返回的错误* Toast.*makeText*(ChongzhiActivity.**this**, **"支付失败"**,  Toast.***LENGTH\_SHORT***).show();  }  }  }   ;  };   */\*\*  \* get the sign type we use. 获取签名方式  \*/* **public** String getSignType() {  **return "sign\_type=\"RSA\""**;  }   */\*\*  \* sign the order info. 对订单信息进行签名  \*  \** ***@param content*** *待签名订单信息  \*/* **public** String sign(String content) {  **return** SignUtils.*sign*(content, PayKeys.***PRIVATE***);  }   */\*\*  \* create the order info. 创建订单信息  \*/* **public** String getOrderInfo(String subject, String body, String price) {  *// 签约合作者身份ID* String orderInfo = **"partner="** + **"\""** + PayKeys.***DEFAULT\_PARTNER*** + **"\""**;   *// 签约卖家支付宝账号* orderInfo += **"&seller\_id="** + **"\""** + PayKeys.***DEFAULT\_SELLER*** + **"\""**;   *// 商户网站唯一订单号* orderInfo += **"&out\_trade\_no="** + **"\""** + getOutTradeNo() + **"\""**;   *// 商品名称* orderInfo += **"&subject="** + **"\""** + subject + **"\""**;   *// 商品详情* orderInfo += **"&body="** + **"\""** + body + **"\""**;   *// 商品金额* orderInfo += **"&total\_fee="** + **"\""** + price + **"\""**;   *// 服务器异步通知页面路径* orderInfo += **"&notify\_url="** + **"\""** + **"http://notify.msp.hk/notify.htm"** + **"\""**;   *// 服务接口名称， 固定值* orderInfo += **"&service=\"mobile.securitypay.pay\""**;   *// 支付类型， 固定值* orderInfo += **"&payment\_type=\"1\""**;   *// 参数编码， 固定值* orderInfo += **"&\_input\_charset=\"utf-8\""**;   *// 设置未付款交易的超时时间  // 默认30分钟，一旦超时，该笔交易就会自动被关闭。  // 取值范围：1m～15d。  // m-分钟，h-小时，d-天，1c-当天（无论交易何时创建，都在0点关闭）。  // 该参数数值不接受小数点，如1.5h，可转换为90m。* orderInfo += **"&it\_b\_pay=\"30m\""**;   *// extern\_token为经过快登授权获取到的alipay\_open\_id,带上此参数用户将使用授权的账户进行支付  // orderInfo += "&extern\_token=" + "\"" + extern\_token + "\"";   // 支付宝处理完请求后，当前页面跳转到商户指定页面的路径，可空* orderInfo += **"&return\_url=\"m.alipay.com\""**;   *// 调用银行卡支付，需配置此参数，参与签名， 固定值 （需要签约《无线银行卡快捷支付》才能使用）  // orderInfo += "&paymethod=\"expressGateway\"";* **return** orderInfo;  }   */\*\*  \* get the out\_trade\_no for an order. 生成商户订单号，该值在商户端应保持唯一（可自定义格式规范）  \*/* **public** String getOutTradeNo() {  SimpleDateFormat format = **new** SimpleDateFormat(**"MMddHHmmss"**,  Locale.*getDefault*());  Date date = **new** Date();  String key = format.format(date);  Random r = **new** Random();  key = key + r.nextInt();  key = key.substring(0, 15);  **return** key;  }    @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  *//* ***TODO: add setContentView(...) invocation*** ButterKnife.*bind*(**this**);  } } |

##### 其中，需要提供支付宝相关的API：



#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>   <**com.atguigu.p2p.ui.MyScrollview  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="44dp"  android:layout\_marginTop="20dp"  android:background="#fff"**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="#ccc"**/>   <**TextView  android:id="@+id/chongzhi\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_marginLeft="20dp"  android:text="充值金额"  android:textColor="#333"  android:textSize="14sp"**/>   <**View  android:id="@+id/view"  android:layout\_width="1dp"  android:layout\_height="19dp"  android:layout\_centerVertical="true"  android:layout\_marginLeft="18dp"  android:layout\_toRightOf="@+id/chongzhi\_text"  android:background="#ccc"**/>   *<!-- edittext如果默认看不到光标闪烁，是因为当前编译的系统版本edittext控件的光标颜色和你设置的EditText的背景颜色重叠了，所以看不到而已  android:textCursorDrawable="@null"设置光标的颜色和textColor属性保持一致  -->* <**EditText  android:id="@+id/chongzhi\_et"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_marginLeft="18dp"  android:layout\_toRightOf="@+id/view"  android:background="@color/white"  android:hint="请输入充值金额"  android:inputType="numberDecimal"  android:maxLength="19"  android:singleLine="true"  android:textColor="#333333"  android:textCursorDrawable="@null"  android:textSize="13sp"**>  </**EditText**>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="#ccc"**/>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="23dp"**>   <**TextView  android:id="@+id/chongzhi\_text2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="您的账户目前可用余额："  android:textColor="#333"  android:textSize="11sp"**/>   <**TextView  android:id="@+id/yue\_tv"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@+id/chongzhi\_text2"  android:text="0"  android:textColor="#e84768"  android:textSize="11sp"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@+id/yue\_tv"  android:text="元"  android:textSize="11sp"**/>  </**RelativeLayout**>   <**Button  android:id="@+id/chongzhi\_btn"  android:layout\_width="match\_parent"  android:layout\_height="45dp"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="22dp"  android:background="@drawable/btn\_023"  android:text="充值"  android:clickable="false"  android:textColor="#fff"  android:textSize="15sp"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginTop="45dp"  android:text="温馨提示："  android:textColor="#666"  android:textSize="14sp"**/>   *<!--text当中使用\n即可换行书写多行文本内容-->* <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="10dp"  android:text="1、所有投资保证金将有第三方平台托管。\n2、线上、线下充值手续费全免。\n3、请注意您的银行卡充值限额，以免造成不便。\n4、如果充值进的没有及时到帐，请与客服联系。\n5、禁止信用卡套现、虚假交易等行为，已经发现将予以处罚,包括但不限于：限制收款、冻结账户，永久停止服务等。\n6、如果您的账户充值金额为用于投资在本平台的理财服务,3个月之后才能进行体现申请。\n7、如果已在手机客户端充过值，则电脑端或手机端体现都只能使用该银行卡。除非原卡丢失、损坏或停用，否则一经绑定银行卡，将不得更换。此情况下，请联系客服并提供由银行提供的书面证明材料。"  android:textColor="#999"  android:textSize="13dp"**/>  </**LinearLayout**>  </**com.atguigu.p2p.ui.MyScrollview**>   </**LinearLayout**> |

#### 在MeFragment中调用：

|  |
| --- |
| @OnClick(R.id.***recharge***) **public void** cz(View view) {  ((BaseActivity) getActivity()).gotoActivity(ChongzhiActivity.**class**, **null**); } |

### 功能二十二：提现业务的实现

#### 在activity包下提供TiXianActivity类：

|  |
| --- |
| */\*\*  \* Created by shkstart.  \* <p/>  \* p2p金融项目当中的提现业务:  \* 并不是即时到账，业务流程：用户发送一个提现请求---->经过这个公司审核该请求--->审核通过--->24小时或48小时内将提现的钱款打入收款账户银行卡  \*/* **public class** TiXianActivity **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***account\_zhifubao***)  TextView **accountZhifubao**;  @Bind(R.id.***select\_bank***)  RelativeLayout **selectBank**;  @Bind(R.id.***chongzhi\_text***)  TextView **chongzhiText**;  @Bind(R.id.***view***)  View **view**;  @Bind(R.id.***input\_money***)  EditText **inputMoney**;  @Bind(R.id.***chongzhi\_text2***)  TextView **chongzhiText2**;  @Bind(R.id.***textView5***)  TextView **textView5**;  @Bind(R.id.***btn\_tixian***)  Button **btnTixian**;   @Override  **protected void** initData() {    }   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"提现"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }   @OnClick(R.id.***iv\_title\_back***)  **public void** back(View view) {  closeCurrent();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_tixian***;  }   @OnClick(R.id.***btn\_tixian***)  **public void** clicktx(View view) {  Toast.*makeText*(**this**, **"你的提现请求已被成功受理...审核通过后，24小时内，你的钱自然会到账..."**, Toast.***LENGTH\_SHORT***).show();  UIUtils.*getHandler*().postDelayed(**new** Runnable() {  @Override  **public void** run() {  closeCurrent();  }  }, 2000);  }   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  *//* ***TODO: add setContentView(...) invocation*** ButterKnife.*bind*(**this**);  } } |

#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>   <**com.atguigu.p2p.ui.MyScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_marginTop="12dp"  android:background="#ccc"**/>   <**RelativeLayout  android:id="@+id/select\_bank"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:background="#fff"**>   <**LinearLayout  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:orientation="vertical"**>   <**TextView  android:id="@+id/account\_zhifubao"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:text="支付宝: 151428109@qq.com"  android:textColor="#333"  android:textSize="14sp"**/>  </**LinearLayout**>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="#ccc"**/>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:layout\_marginTop="12dp"  android:background="#ccc"**/>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="44dp"  android:background="#fff"**>   <**TextView  android:id="@+id/chongzhi\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_marginLeft="20dp"  android:text="提现金额"  android:textColor="#333"  android:textSize="14sp"**/>   <**View  android:id="@+id/view"  android:layout\_width="1dp"  android:layout\_height="19dp"  android:layout\_centerVertical="true"  android:layout\_marginLeft="18dp"  android:layout\_toRightOf="@+id/chongzhi\_text"  android:background="#ccc"**/>   <**EditText  android:id="@+id/input\_money"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_marginLeft="18dp"  android:layout\_toRightOf="@+id/view"  android:background="@color/white"  android:hint="请输入100元以上金额"  android:inputType="numberDecimal"  android:maxLength="19"  android:singleLine="true"  android:textColor="#333333"  android:textCursorDrawable="@null"  android:textSize="13sp"**>  </**EditText**>  </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="#ccc"**/>   <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="23dp"**>   <**TextView  android:id="@+id/chongzhi\_text2"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="您的账户目前可用余额："  android:textColor="#333"  android:textSize="11sp"**/>   <**TextView  android:id="@+id/textView5"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@+id/chongzhi\_text2"  android:text="0"  android:textColor="#e84768"  android:textSize="11sp"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_toRightOf="@+id/textView5"  android:text="元"  android:textSize="11sp"**/>   </**RelativeLayout**>   <**Button  android:id="@+id/btn\_tixian"  android:layout\_width="match\_parent"  android:layout\_height="45dp"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="22dp"  android:background="@drawable/btn\_01"  android:text="提现"  android:textColor="#fff"  android:textSize="15sp"**/>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginTop="45dp"  android:text="温馨提示："  android:textColor="#666"  android:textSize="13sp"**/>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginLeft="20dp"  android:layout\_marginRight="20dp"  android:layout\_marginTop="10dp"  android:text="1、若您未在本平台理财，账户余额在3个月以后才能进行提现申请。\n2、如您已绑定过银行卡的快捷签约，绑定后，无论在电脑PC端或手机端体现都只能使用该银行卡。除非原卡丢失、损坏或停用，否侧一经绑定银行卡，将不得更换。此种情况下，请提供由银行提供的书面证明材料。"  android:textColor="#999"  android:textSize="13dp"**/>  </**LinearLayout**>  </**com.atguigu.p2p.ui.MyScrollView**>   </**LinearLayout**> |

#### 在MeFragment中调用：

|  |
| --- |
| @OnClick(R.id.***withdraw***) **public void** tx(View view) {  ((BaseActivity) getActivity()).gotoActivity(TiXianActivity.**class**, **null**); } |

### 功能二十三：折线图的分析与实现

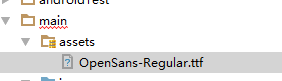
#### LineChartActivity声明为：

|  |
| --- |
| */\*\*  \* Created by shkstart  \*/* **public class** LineChartActivity **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***chart***)  LineChart **chart**;  **private** Typeface **mTf**;   @Override  **protected void** initData() {  **mTf** = Typeface.*createFromAsset*(getAssets(), **"OpenSans-Regular.ttf"**);  *// apply styling  // holder.chart.setValueTypeface(mTf);  //绘制图表右下角文字描述信息* **chart**.setDescription(**"折线图demo"**);  **chart**.setDrawGridBackground(**true**);   *//绘制图表的X轴* XAxis xAxis = **chart**.getXAxis();  xAxis.setPosition(XAxis.XAxisPosition.***BOTTOM***);  xAxis.setTypeface(**mTf**);  xAxis.setDrawGridLines(**false**);  xAxis.setDrawAxisLine(**true**);   *//绘制图表的Y轴* YAxis leftAxis = **chart**.getAxisLeft();  leftAxis.setTypeface(**mTf**);  *//false:代表值是平均分配的;* leftAxis.setLabelCount(7, **false**);   **chart**.getAxisRight().setEnabled(**false**); *// YAxis rightAxis = chart.getAxisRight(); // rightAxis.setTypeface(mTf); // rightAxis.setLabelCount(5, false); // rightAxis.setDrawGridLines(false);   // set data* **chart**.setData(generateDataLine());  *// do not forget to refresh the chart  // holder.chart.invalidate();* **chart**.animateX(750);  }    */\*\*  \* generates a random ChartData object with just one DataSet  \*  \** ***@return*** *\*/* **private** LineData generateDataLine() {   ArrayList<Entry> e1 = **new** ArrayList<Entry>();  **for** (**int** i = 0; i < 12; i++) {  e1.add(**new** Entry((**int**) (Math.*random*() \* 65) + 40, i));  }  LineDataSet d1 = **new** LineDataSet(e1, **"New DataSet 1"**);  *//指定数据集合绘制时候的属性* d1.setLineWidth(5f);  d1.setCircleSize(7.5f);  d1.setHighLightColor(Color.***BLACK***);  d1.setDrawValues(**true**);   ArrayList<LineDataSet> sets = **new** ArrayList<LineDataSet>();  sets.add(d1);   LineData cd = **new** LineData(getMonths(), sets);  **return** cd;  }   **private** ArrayList<String> getMonths() {  ArrayList<String> m = **new** ArrayList<String>();  m.add(**"Jan"**);  m.add(**"Feb"**);  m.add(**"Mar"**);  m.add(**"Apr"**);  m.add(**"May"**);  m.add(**"Jun"**);  m.add(**"Jul"**);  m.add(**"Aug"**);  m.add(**"Sep"**);  m.add(**"Okt"**);  m.add(**"Nov"**);  m.add(**"Dec"**);  **return** m;  }   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"折线图示例"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }   @OnClick(R.id.iv\_title\_back)  **public void** back(View view) {  closeCurrent();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_linechart***;  } } |

#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"**>   <**com.github.mikephil.charting.charts.LineChart  android:id="@+id/chart"  android:layout\_width="match\_parent"  android:layout\_height="250dp"**/>  </**LinearLayout**> </**LinearLayout**> |

##### 需要提供必要的字体：



#### 在MeFragment中调用：

|  |
| --- |
| @OnClick(R.id.***ll\_touzi***) **public void** line(View view) {  ((BaseActivity) getActivity()).gotoActivity(LineChartActivity.**class**, **null**); } |

### 功能二十四：柱状图的分析与实现

#### BarChartActivity声明如下：

|  |
| --- |
| */\*\*  \* Created by shkstart.  \*/* **public class** BarChartActivity **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***chart***)  BarChart **chart**;  **private** Typeface **mTf**;   @Override  **protected void** initData() {  **mTf** = Typeface.*createFromAsset*(getAssets(), **"OpenSans-Regular.ttf"**);  *// apply styling* **chart**.setDescription(**"柱状图示例"**);  **chart**.setDrawGridBackground(**false**);  **chart**.setDrawBarShadow(**false**);   XAxis xAxis = **chart**.getXAxis();  xAxis.setPosition(XAxis.XAxisPosition.***BOTTOM***);  xAxis.setTypeface(**mTf**);  xAxis.setDrawGridLines(**false**);  xAxis.setDrawAxisLine(**true**);   YAxis leftAxis = **chart**.getAxisLeft();  leftAxis.setTypeface(**mTf**);  leftAxis.setLabelCount(5, **false**);  *//绘制Y轴方向上最高的顶部的值距离整个图表的top距离* leftAxis.setSpaceTop(30f);   **chart**.getAxisRight().setEnabled(**false**); *// YAxis rightAxis = chart.getAxisRight(); // rightAxis.setTypeface(mTf); // rightAxis.setLabelCount(5, false); // rightAxis.setSpaceTop(20f);* BarData mChartData = generateDataBar();  mChartData.setValueTypeface(**mTf**);  *// set data* **chart**.setData(mChartData);  *// do not forget to refresh the chart // holder.chart.invalidate();* **chart**.animateY(700);  }   */\*\*  \* generates a random ChartData object with just one DataSet  \*  \** ***@return*** *\*/* **private** BarData generateDataBar() {   ArrayList<BarEntry> entries = **new** ArrayList<BarEntry>();   **for** (**int** i = 0; i < 12; i++) {  entries.add(**new** BarEntry((**int**) (Math.*random*() \* 70) + 30, i));  }  BarDataSet d = **new** BarDataSet(entries, **"New DataSet 1"**);  *//绘制数据集合时指定属性* d.setBarSpacePercent(50f); *//柱状块之间的相隔距离* d.setColors(ColorTemplate.***VORDIPLOM\_COLORS***);  d.setHighLightAlpha(255);  BarData cd = **new** BarData(getMonths(), d);  **return** cd;  }   **private** ArrayList<String> getMonths() {  ArrayList<String> m = **new** ArrayList<String>();  m.add(**"Jan"**);  m.add(**"Feb"**);  m.add(**"Mar"**);  m.add(**"Apr"**);  m.add(**"May"**);  m.add(**"Jun"**);  m.add(**"Jul"**);  m.add(**"Aug"**);  m.add(**"Sep"**);  m.add(**"Okt"**);  m.add(**"Nov"**);  m.add(**"Dec"**);  **return** m;  }   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"柱状图示例"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }   @OnClick(R.id.***iv\_title\_back***)  **public void** back(View view) {  closeCurrent();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_barchart***;  }  } |

#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"**>   <**com.github.mikephil.charting.charts.BarChart  android:id="@+id/chart"  android:layout\_width="match\_parent"  android:layout\_height="250dp"  android:background="@android:color/holo\_green\_light"**/>  </**LinearLayout**> </**LinearLayout**> |

#### 在MeFragment中调用：

|  |
| --- |
| @OnClick(R.id.***ll\_touzi\_zhiguan***) **public void** bar(View view) {  ((BaseActivity) getActivity()).gotoActivity(BarChartActivity.**class**, **null**); } |

### 功能二十五：饼状图的分析与实现

#### PieChartActivity声明如下：

|  |
| --- |
| */\*\*  \* Created by shkstart  \*/* **public class** PieChartActivity2 **extends** BaseActivity {    @Bind(R.id.***iv\_title\_back***)  ImageView **ivTitleBack**;  @Bind(R.id.***tv\_title***)  TextView **tvTitle**;  @Bind(R.id.***iv\_title\_setting***)  ImageView **ivTitleSetting**;  @Bind(R.id.***chart***)  PieChart **chart**;   @Override  **protected void** initData() {  *// apply styling* **chart**.setDescription(**"饼图示例"**);  *//绘制最内层的圆的半径* **chart**.setHoleRadius(42f);  *//绘制包裹最内层圆的外层圆的半径* **chart**.setTransparentCircleRadius(57f);  **chart**.setCenterText(**"资产分配"**);  **chart**.setCenterTextSize(14f);  *//是否适用百分比显示值* **chart**.setUsePercentValues(**false**);   PieData mChartData = generateDataPie(); *// mChartData.setValueFormatter(new PercentFormatter());* mChartData.setValueTextSize(14f);  mChartData.setValueTextColor(Color.***RED***);  *// set data* **chart**.setData((PieData) mChartData);   *//饼图指示器绘制* Legend l = **chart**.getLegend();  l.setPosition(Legend.LegendPosition.***RIGHT\_OF\_CHART***);  l.setYEntrySpace(0f);  l.setYOffset(20f);   *// do not forget to refresh the chart  // holder.chart.invalidate();* **chart**.animateXY(900, 900);  }   */\*\*  \* generates a random ChartData object with just one DataSet  \*  \** ***@return*** *\*/* **private** PieData generateDataPie() {   ArrayList<Entry> entries = **new** ArrayList<Entry>();   **for** (**int** i = 0; i < 4; i++) {  entries.add(**new** Entry((**int**) (Math.*random*() \* 70) + 30, i));  }   PieDataSet d = **new** PieDataSet(entries, **""**);   *// space between slices  //绘制图表数据集合的时候指定属性* d.setSliceSpace(1f);  d.setColors(ColorTemplate.***VORDIPLOM\_COLORS***);   PieData cd = **new** PieData(getQuarters(), d);  **return** cd;  }   **private** ArrayList<String> getQuarters() {   ArrayList<String> q = **new** ArrayList<String>();  q.add(**"1st Quarter"**);  q.add(**"2nd Quarter"**);  q.add(**"3rd Quarter"**);  q.add(**"4th Quarter"**);   **return** q;  }   @Override  **protected void** initTitle() {  **tvTitle**.setText(**"饼图示例"**);  **ivTitleBack**.setVisibility(View.***VISIBLE***);  **ivTitleSetting**.setVisibility(View.***INVISIBLE***);  }   @OnClick(R.id.***iv\_title\_back***)  **public void** back(View view) {  closeCurrent();  }   @Override  **public int** getLayoutId() {  **return** R.layout.***activity\_piechart***;  } } |

#### 对应的布局为：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_title"**/>    <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:gravity="center"**>   <**com.github.mikephil.charting.charts.PieChart  android:id="@+id/chart"  android:layout\_width="match\_parent"  android:layout\_height="250dp"  android:background="@android:color/holo\_green\_light"**/>  </**LinearLayout**> </**LinearLayout**> |

#### 在MeFragment中调用：

|  |
| --- |
| @OnClick(R.id.***ll\_zichang***) **public void** pie(View view) {  ((BaseActivity) getActivity()).gotoActivity(PieChartActivity2.**class**, **null**); } |

### 功能二十六：MoreFragment的布局设置

#### 布局设置

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"** >   <**include layout="@layout/common\_top"**></**include**>   <**com.atguigu.p2pinvest0714.ui.MyScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>    <**LinearLayout  android:id="@+id/ll\_more\_regist"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_regist"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="用户注册"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>    <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="60dp"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_secret"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="手势密码"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   <**ToggleButton  android:id="@+id/toggle\_more\_secret"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_centerVertical="true"  android:layout\_marginBottom="3dp"  android:layout\_marginRight="5dp"  android:background="@drawable/toggle\_selector"  android:textOff=""  android:textOn=""** />   </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_reset"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_reset"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="重置手势密码"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_contact"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_weight="1"  android:drawableLeft="@drawable/icon\_more\_contact"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="联系客服"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   <**TextView  android:id="@+id/tv\_more\_number"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_marginRight="5dp"  android:gravity="center\_vertical"  android:text="010-56253825"  android:textColor="@color/round\_red\_common"  android:textSize="22sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_sms"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_sms"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="用户反馈"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_share"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_share"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="分享给好友"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_about"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_about"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="关于硅谷理财"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>  </**LinearLayout**>    </**com.atguigu.p2pinvest0714.ui.MyScrollView**>  </**LinearLayout**> |

#### 需要drawable下创建toggle\_btn.xml文件如下：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  <**item android:state\_checked="true" android:drawable="@drawable/toggle\_btn\_checked"**/>  <**item android:drawable="@drawable/toggle\_btn\_unchecked"**/> </**selector**> |

### 功能二十七：用户注册功能

#### UserRegsiterActivity布局设置：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"** >   <**include layout="@layout/common\_top"**></**include**>   <**com.atguigu.p2pinvest0714.ui.MyScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>    <**LinearLayout  android:id="@+id/ll\_more\_regist"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_regist"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="用户注册"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>    <**RelativeLayout  android:layout\_width="match\_parent"  android:layout\_height="60dp"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_secret"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="手势密码"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   <**ToggleButton  android:id="@+id/toggle\_more\_secret"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentRight="true"  android:layout\_centerVertical="true"  android:layout\_marginBottom="3dp"  android:layout\_marginRight="5dp"  android:background="@drawable/toggle\_selector"  android:textOff=""  android:textOn=""** />   </**RelativeLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_reset"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_reset"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="重置手势密码"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_contact"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="0dp"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:layout\_weight="1"  android:drawableLeft="@drawable/icon\_more\_contact"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="联系客服"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   <**TextView  android:id="@+id/tv\_more\_number"  android:layout\_width="wrap\_content"  android:layout\_height="match\_parent"  android:layout\_marginRight="5dp"  android:gravity="center\_vertical"  android:text="010-56253825"  android:textColor="@color/round\_red\_common"  android:textSize="22sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_sms"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_sms"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="用户反馈"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_share"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_share"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="分享给好友"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>   <**View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="@color/divider\_line"**></**View**>   <**LinearLayout  android:id="@+id/ll\_more\_about"  android:layout\_width="match\_parent"  android:layout\_height="60dp"  android:gravity="center\_vertical"**>   <**TextView  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_centerVertical="true"  android:drawableLeft="@drawable/icon\_more\_about"  android:drawablePadding="10dp"  android:gravity="center\_vertical"  android:padding="5dp"  android:text="关于硅谷理财"  android:textColor="@color/more\_text"  android:textSize="20sp"** />   </**LinearLayout**>  </**LinearLayout**>    </**com.atguigu.p2pinvest0714.ui.MyScrollView**>  </**LinearLayout**> |

#### 注册功能的实现

|  |
| --- |
| **public class** UserRegisterActivity **extends** BaseActivity **implements** View.OnClickListener {    @Bind(R.id.***iv\_top\_back***)  ImageView **ivTopBack**;  @Bind(R.id.***tv\_top\_title***)  TextView **tvTopTitle**;  @Bind(R.id.***iv\_top\_setting***)  ImageView **ivTopSetting**;  @Bind(R.id.***tv\_register\_number***)  TextView **tvRegisterNumber**;  @Bind(R.id.***et\_register\_number***)  EditText **etRegisterNumber**;  @Bind(R.id.***rl\_register\_number***)  RelativeLayout **rlRegisterNumber**;  @Bind(R.id.***tv\_register\_name***)  TextView **tvRegisterName**;  @Bind(R.id.***et\_register\_name***)  EditText **etRegisterName**;  @Bind(R.id.***rl\_register\_name***)  RelativeLayout **rlRegisterName**;  @Bind(R.id.***tv\_register\_secret***)  TextView **tvRegisterSecret**;  @Bind(R.id.***et\_register\_secret***)  EditText **etRegisterSecret**;  @Bind(R.id.***tv\_register\_verify***)  TextView **tvRegisterVerify**;  @Bind(R.id.***et\_register\_verify***)  EditText **etRegisterVerify**;  @Bind(R.id.***btn\_register\_login***)  Button **btnRegisterLogin**;   @Override  **protected void** initTitle() {  **ivTopBack**.setVisibility(View.***VISIBLE***);  **tvTopTitle**.setText(**"用户注册"**);  **ivTopSetting**.setVisibility(View.***INVISIBLE***);  }   @Override  **protected void** initData() {  *//给Button设置点击事件* **btnRegisterLogin**.setOnClickListener(**this**);  }   @OnClick(R.id.***iv\_top\_back***)  **public void** back(View view){  removeCurrentActivity();  }   @Override  **protected int** getLayoutId() {  **return** R.layout.***activity\_user\_register***;  }   @Override  **public void** onClick(View v) {  String name = **etRegisterName**.getText().toString().trim();  String phone = **etRegisterNumber**.getText().toString().trim();  String password = **etRegisterSecret**.getText().toString().trim();  String verify = **etRegisterVerify**.getText().toString().trim();  **if**(TextUtils.*isEmpty*(name) || TextUtils.*isEmpty*(phone) || TextUtils.*isEmpty*(password) || TextUtils.*isEmpty*(verify)){  Toast.*makeText*(UserRegisterActivity.**this**, **"输入的信息不能为空"**, Toast.***LENGTH\_SHORT***).show();   }**else if**(!password.equals(verify)){  **etRegisterSecret**.setText(**""**);  **etRegisterVerify**.setText(**""**);  Toast.*makeText*(UserRegisterActivity.**this**, **"两次输入的密码不一致"**, Toast.***LENGTH\_SHORT***).show();  }**else**{  *//联网发送用户信息给服务器* String register = AppNetConfig.***REGISTER***;  RequestParams params = **new** RequestParams();  **try** {  *//为了避免出现乱码* name = **new** String(name.getBytes(),**"UTF-8"**);  params.put(**"name"**,name);  params.put(**"phone"**,phone);  password = MD5Utils.*MD5*(password);  params.put(**"password"**,password);  **client**.post(register,params,**new** AsyncHttpResponseHandler(){  @Override  **public void** onSuccess(String content) {  *//如果注册的电话号码的用户已经存在，提示：用户已注册* JSONObject jsonObject = JSON.*parseObject*(content);  **boolean** isExist = jsonObject.getBoolean(**"isExist"**);  **if**(isExist){  Toast.*makeText*(UserRegisterActivity.**this**, **"用户已注册"**, Toast.***LENGTH\_SHORT***).show();  **return**;  }  *//如果之前用户没有注册过，则注册成功* Toast.*makeText*(UserRegisterActivity.**this**, **"注册成功"**, Toast.***LENGTH\_SHORT***).show();  **etRegisterName**.setText(**""**);  **etRegisterNumber**.setText(**""**);  **etRegisterSecret**.setText(**""**);  **etRegisterVerify**.setText(**""**);   UIUtils.*getHandler*().postDelayed(**new** Runnable() {  @Override  **public void** run() {  removeCurrentActivity();  }  },2000);  }   @Override  **public void** onFailure(Throwable error, String content) {  Toast.*makeText*(UserRegisterActivity.**this**, **"网络异常，注册失败"**, Toast.***LENGTH\_SHORT***).show();   }  });  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  } } |

### 功能二十八：手势密码开关设置

#### 在MoreFragment中：每次进入初始化ToggleButton状态

|  |
| --- |
| *//获取设置手势密码的ToggleButton的状态* **private void** getGestureStatus() {  **sp** = **this**.getActivity().getSharedPreferences(**"secret\_protect"**, Context.***MODE\_PRIVATE***);  *//读取当前的toggleButton的状态并显示* **boolean** isOpen = **sp**.getBoolean(**"isOpen"**, **false**);  **toggleMoreSecret**.setChecked(isOpen); } |

如上方法在initData()中调用即可

#### 在MoreFragment中，设置手势密码

|  |
| --- |
| *//设置手势密码的ToggleButton状态改变的监听* **private void** setGestureListener() {  **toggleMoreSecret**.setOnCheckedChangeListener(**new** CompoundButton.OnCheckedChangeListener() {  @Override  **public void** onCheckedChanged(CompoundButton buttonView, **boolean** isChecked) {  **if** (isChecked) {  *//判断本地是否设置过手势密码* String inputCode = **sp**.getString(**"inputCode"**, **""**);  **if** (TextUtils.*isEmpty*(inputCode)) {  **new** AlertDialog.Builder(MoreFragment.**this**.getActivity())  .setTitle(**"设置手势密码"**)  .setMessage(**"是否现在设置手势密码"**)  .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  Toast.*makeText*(UIUtils.*getContext*(), **"现在设置手势密码"**, Toast.***LENGTH\_SHORT***).show(); *// toggleMoreSecret.setChecked(true);* **sp**.edit().putBoolean(**"isOpen"**, **true**).commit();  ((BaseActivity) (MoreFragment.**this**.getActivity())).goToActivity(GestureEditActivity.**class**, **null**);  }  })  .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  Toast.*makeText*(UIUtils.*getContext*(), **"取消了手势密码设置"**, Toast.***LENGTH\_SHORT***).show();  **toggleMoreSecret**.setChecked(**false**);  **sp**.edit().putBoolean(**"isOpen"**, **false**).commit();  }  })  .show();   } **else** {*//之前设置过* Toast.*makeText*(UIUtils.*getContext*(), **"开启手势解锁"**, Toast.***LENGTH\_SHORT***).show(); *// toggleMoreSecret.setChecked(true);* **sp**.edit().putBoolean(**"isOpen"**, **true**).commit();  }    } **else** {  Toast.*makeText*(UIUtils.*getContext*(), **"关闭手势解锁"**, Toast.***LENGTH\_SHORT***).show();  *// toggleMoreSecret.setChecked(false);* **sp**.edit().putBoolean(**"isOpen"**, **false**).commit();   }  }  });  } |

#### 在MeFragment的doUser()中验证进入：

|  |
| --- |
| *//判断本地用户是否开启了手势密码，如果开启，需要进入* **boolean** isSecretOpen = **this**.getActivity().getSharedPreferences(**"secret\_protect"**, Context.***MODE\_PRIVATE***).getBoolean(**"isOpen"**, **false**); **if**(isSecretOpen){    ((BaseActivity)**this**.getActivity()).goToActivity(GestureVerifyActivity.**class**,**null**); } |

#### 提供GestureEditActivity类

（直接从现有的GestureLock项目拷贝过来即可）

#### 提供GestureVerifyActivity类

（直接从现有的GestureLock项目拷贝过来即可）

### 功能二十九：联系客服

|  |
| --- |
| **private void** contactSupportStaff() {  **llMoreContact**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  **new** AlertDialog.Builder(MoreFragment.**this**.getActivity())  .setTitle(**"联系客服"**)  .setMessage(**"是否联系客服010-56253825"**)  .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  Intent intent = **new** Intent(Intent.***ACTION\_CALL***);   Uri uri = Uri.*parse*(**"tel:010-56253825"**);  intent.setData(uri);  **try** {  MoreFragment.**this**.getActivity().startActivity(intent);  } **catch** (Exception e) {  e.printStackTrace();  }  }  })  .setNegativeButton(**"取消"**, **null**)  .show();  }  }); } |

### 功能三十：用户反馈

|  |
| --- |
| **private** String **department** = **"用户未选择"**;  *//反馈信息* **private void** fanKui() {   **llMoreSms**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {   **final** View view = View.*inflate*(MoreFragment.**this**.getActivity(), R.layout.***view\_fankui***, **null**);  **final** RadioGroup rg\_fankui = (RadioGroup) view.findViewById(R.id.***rg\_fankui***);  *//获取选中的部门* rg\_fankui.setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener() {  @Override  **public void** onCheckedChanged(RadioGroup group, **int** checkedId) {  *//获取选中的部门信息* RadioButton rb\_fankui = (RadioButton) rg\_fankui.findViewById(checkedId);  **department** = rb\_fankui.getText().toString();  }  });   **new** AlertDialog.Builder(MoreFragment.**this**.getActivity())  .setView(view)  .setPositiveButton(**"发送"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  EditText et\_fankui\_content = (EditText) view.findViewById(R.id.***et\_fankui\_content***);  String fanKuiContent = et\_fankui\_content.getText().toString();  *//将反馈信息发送给服务器* AsyncHttpClient client = **new** AsyncHttpClient();  String url = AppNetConfig.***FEEDBACK***;  RequestParams params = **new** RequestParams();   **try** {  *//为了避免出现乱码，必须如下处理* **department** = **new** String(**department**.getBytes(), **"utf-8"**);  params.put(**"department"**, **department**);  fanKuiContent = **new** String(fanKuiContent.getBytes(), **"utf-8"**);  params.put(**"content"**, fanKuiContent);  client.post(url, params, **new** AsyncHttpResponseHandler() {  @Override  **public void** onSuccess(String content) {  Toast.*makeText*(MoreFragment.**this**.getActivity(), **"发送反馈信息成功"**, Toast.***LENGTH\_SHORT***).show();  }   @Override  **public void** onFailure(Throwable error, String content) {  Toast.*makeText*(MoreFragment.**this**.getActivity(), **"发送反馈信息失败"**, Toast.***LENGTH\_SHORT***).show();  }  });  } **catch** (UnsupportedEncodingException e) {  e.printStackTrace();  }  }  })  .setNegativeButton(**"取消"**, **null**)  .show();  }  }); } |

### 功能三十一：分享给好友

直接参照思维导图中提供的步骤集成即可。

### 功能三十二：关于硅谷理财

|  |
| --- |
| *//关于硅谷理财* **private void** aboutGuiguInvest() {  **llMoreAbout**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  ((BaseActivity) MoreFragment.**this**.getActivity()).goToActivity(GuiGuInvestActivity.**class**, **null**);  }  }); } |

对应的布局：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"**>   <**include layout="@layout/common\_top"**></**include**>   <**ScrollView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"**>   <**LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical"  android:paddingLeft="5dp"  android:paddingRight="5dp"** >   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:gravity="center"  android:layout\_marginTop="5dp"  android:text="硅谷金融隶属于尚硅谷IT教育品牌"  android:textSize="17sp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:text="硅谷金融(www.atguigu.com)，系尚硅谷(中国)旗下独立运营品牌。“硅谷金融”是一家基于互联网技术为用户提供金融理财服务的专业化公司。"  android:textSize="17sp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:text="硅谷金融自2014年上线运营以来，得到用户的广泛信赖和一致好评。目前“硅谷金融”服务的用户覆盖全国30余个省市、自治区，成功帮助用户获得借款并让出借用户获得稳定的高收益。"  android:textSize="17sp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:text="硅谷金融拥有完善的管理制度、卓越的管理团队、标准的运作流程、严格的风控措施，为平台两端的用户提供咨询、评估、理财方案制定、协议管理等全方位的专业化微金融与财富管理咨询服务。"  android:textSize="17sp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:text="硅谷金融是以个人对个人小额借贷为主要模式，通过为出借人提供可靠、稳健、投资回报高的理财产品，为借款人提供快捷、便利、长期稳定的低成本资金撮合渠道，从而构建透明、便捷、平等的网络金融交易服务平台。硅谷金融受益于严格的个人金融风险定价体系和安全分散有保障的投资机制，借款人和出借人的权益均可得到充分的保障和满足。"  android:textSize="17sp"** />   <**TextView  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:layout\_marginBottom="5dp"  android:text="在互联网金融蓬勃发展的今天，硅谷金融不断探索创新模式，致力于成为一家卓越的互联网金融企业，为个体金融、普惠金融发展添助力。"  android:textSize="17sp"** />   </**LinearLayout**>   </**ScrollView**> </**LinearLayout**> |

### 功能三十三：联网更新应用

#### onCreate()中判断手机是否联网：

没有联网：直接进入主界面。联网：则联网获取服务器版本

|  |
| --- |
| **private** UpdateInfo **updateInfo**; *//最新版本信息对象* **private void** initData() {  **startTime** = System.*currentTimeMillis*();  *//1.检查手机是否联网* **boolean** connected = isConnected();  **if** (!connected) {  Toast.*makeText*(**this**, **"没有联网..."**, Toast.***LENGTH\_SHORT***).show();  *//进入主界面* toMain();  } **else** {*//联网成功* String updateUrl = AppNetConfig.***URL\_UPDATE***;  AsyncHttpClient client = **new** AsyncHttpClient();   client.post(updateUrl, **new** AsyncHttpResponseHandler() {  @Override  **public void** onSuccess(String content) {  *//请求服务器得到最新版本的信息并封装为UpdateInfo对象* **updateInfo** = JSON.*parseObject*(content, UpdateInfo.**class**);  **handler**.sendEmptyMessage(***WHAT\_REQUEST\_UPDATE\_SUCCESS***);  }   @Override  **public void** onFailure(Throwable error, String content) {  *//请求失败* **handler**.sendEmptyMessage(***WHAT\_REQUEST\_UPDATE\_ERROR***);  }  });  }  } |

##### 其中，判断是否联网操作如下：

|  |
| --- |
| */\*\*  \* 判断手机是否联网  \* ConnectivityManager  \*/* **private boolean** isConnected() {  **boolean** connected = **false**;   ConnectivityManager manager = (ConnectivityManager) getSystemService(Context.***CONNECTIVITY\_SERVICE***);  NetworkInfo networkInfo = manager.getActiveNetworkInfo();  **if** (networkInfo != **null**) {  connected = networkInfo.isConnected();  }  **return** connected; } |

##### 其中，进入主界面的操作如下：

|  |
| --- |
| */\*\*  \* 进入主页面  \*/* **private void** toMain() {  **long** currentTime = System.*currentTimeMillis*();  *//得到需要延迟的时间* **int** delayTime = (**int**) (3000 - (currentTime - **startTime**));   **if** (delayTime < 0) {  delayTime = 0;  }  **handler**.sendEmptyMessageDelayed(***WHAT\_START\_MAIN***, delayTime); } |

所有消息的处理如下：

|  |
| --- |
| **private static final int *WHAT\_START\_MAIN*** = 1; **private static final int *WHAT\_REQUEST\_UPDATE\_SUCCESS*** = 2; **private static final int *WHAT\_REQUEST\_UPDATE\_ERROR*** = 3; **private static final int *WHAT\_DOWNLOAD\_SUCCESS*** = 4; **private static final int *WHAT\_DOWNLOAD\_ERROR*** = 5; **private** Handler **handler** = **new** Handler(){  **public void** handleMessage(Message msg){  **switch** (msg.**what**) {  **case *WHAT\_START\_MAIN*** :  finish();  startActivity(**new** Intent(WelcomeActivity.**this**,MainActivity.**class**));  **break**;  **case *WHAT\_REQUEST\_UPDATE\_ERROR***:  Toast.*makeText*(getApplicationContext(), **"获取最新版本信息失败"**,Toast.***LENGTH\_SHORT***).show();  toMain();  **break**;  **case *WHAT\_REQUEST\_UPDATE\_SUCCESS***:  String version = getVersion();  **if**(version.equals(**updateInfo**.getVersion())) {*//当前就是最新版本,直接进入主界面* Toast.*makeText*(WelcomeActivity.**this**, **"当前是最新版本"**, Toast.***LENGTH\_SHORT***).show();  toMain();  } **else** {*//有更新的版本  //显示下载的dialog* showDownloadDialog();  }  **break**;  **case *WHAT\_DOWNLOAD\_SUCCESS***:  *//安装apk* installApk();  **break**;  **case *WHAT\_DOWNLOAD\_ERROR***:  Toast.*makeText*(getApplicationContext(), **"下载apk失败"**, Toast.***LENGTH\_SHORT***).show();  toMain();  **break**;  }  } }; |

#### 根据响应数据，判断是否需要更新版本

##### 获取响应数据失败：登录主界面。获取响应数据成功，判断是否需要更新

|  |
| --- |
| */\*\*  \* 当前版本号  \** ***@return*** *\*/* **private** String getVersion() {  String version = **"未知版本"**;  PackageManager manager = getPackageManager();  **try** {  PackageInfo packageInfo = manager.getPackageInfo(getPackageName(), 0);  version = packageInfo.**versionName**;  } **catch** (PackageManager.NameNotFoundException e) {  *//e.printStackTrace(); //如果找不到对应的应用包信息, 就返回"未知版本"* }   **return** version; } |

##### 显示是否需要下载的Dialog:

|  |
| --- |
| */\*\*  \* 显示判断是否需要下载的dialog  \*/* **private void** showDownloadDialog() {  **new** AlertDialog.Builder(**this**)  .setTitle(**"下载最新版本"**)  .setMessage(**updateInfo**.getDesc())  .setPositiveButton(**"立即下载"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  *//启动分线程下载apk* downApk();  }  })  .setNegativeButton(**"暂不下载"**, **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  toMain();  }  })  .show(); } |

#### 下载最新的APK:

|  |
| --- |
| **private** ProgressDialog **pd**; **private** File **apkFile**; */\*\*  \* 下载最新的APK  \*/* **private void** downApk() {  Toast.*makeText*(**this**, **"下载最新的APK"**, Toast.***LENGTH\_SHORT***).show();   *//准备pd* **pd** = **new** ProgressDialog(**this**);  **pd**.setProgressStyle(ProgressDialog.***STYLE\_HORIZONTAL***);  **pd**.show();  *//准备apkFile // /sdcard/udpate.apk* File sdFile = Environment.*getExternalStorageDirectory*();  **apkFile** = **new** File(sdFile, **"update.apk"**);   *//启动分线程下载Apk* **new** Thread(){  **public void** run() {  *//下载* **try** {  downloadAPK();  *//下载成功* **handler**.sendEmptyMessage(***WHAT\_DOWNLOAD\_SUCCESS***);  } **catch** (Exception e) {*//下载失败* e.printStackTrace();  **handler**.sendEmptyMessage(***WHAT\_DOWNLOAD\_ERROR***);  } **finally** {  *//移除dialog* **pd**.dismiss();  }  }  }.start();  } |

##### 其中，downloadAPK()方法如下：

|  |
| --- |
| */\*\*  \* 下载apk文件, 并同步显示进度  \*/* **public void** downloadAPK() **throws** Exception {   InputStream is = **null**;  *// 1). 得到HttpUrlConnection对象* URL url = **new** URL(**updateInfo**.getApkUrl());  HttpURLConnection conn = (HttpURLConnection) url.openConnection();  *// 2). 设置* conn.setConnectTimeout(5000);  conn.setReadTimeout(5000);  *// 3). 连接* conn.connect();  *// 4). 请求并获取服务器端返回的数据流* **int** responseCode = conn.getResponseCode();  **if**(responseCode==200) {  *//设置最大进度* **pd**.setMax(conn.getContentLength());  is = conn.getInputStream();  *//得到apk文件的输出流* FileOutputStream fos = **new** FileOutputStream(**apkFile**);  **byte**[] buffer = **new byte**[1024];  **int** len = -1;  **while**((len=is.read(buffer))!=-1) {  *//写数据* fos.write(buffer, 0, len);  *//更新进度* **pd**.incrementProgressBy(len);  *//模拟网速慢* Thread.*sleep*(2);  }  fos.close();  is.close();  }  *//5). 关闭流, 连接* conn.disconnect(); } |

#### 安装APK：

|  |
| --- |
| */\*\*  \* 安装apk  \*/* **private void** installApk() {  Intent intent = **new** Intent(**"android.intent.action.INSTALL\_PACKAGE"**);  intent.setData(Uri.*parse*(**"file:"** + **apkFile**.getAbsolutePath()));  startActivity(intent); } |