Android笔记

## 1. 按钮的单击事件

和C#不同，比较复杂，首先一个Activity运行时都要执行OnCreate等方法，因此初始化的代码都可以写到里面。下面是按钮的单击事件

Button btn2 = (Button)findViewById(R.id.***button2***);  
btn2.setOnClickListener(**new** View.OnClickListener()  
{  
 @Override  
 **public void** onClick(View v)  
 {  
 image.setImageResource(**images**[++**currentImg**%**images**.**length**]);  
 }  
});

需要通过findViewById从xml中找到控件名。然后设置它的监听方法。在里面重写单击事件方法。

Android是用java写的，因此变量名肯定不能用数字开头。并且导入的文件名也不能以数字开头，否则会报错。

刚刚写了个显示图片的程序，为了简单，将图片名设置为1,2,3.引用的时候为R.drawable.1, R.drawable.2, R.drawable.3,就出错了，将名字改为字母开头就没有问题了。

Android的ImageView组件是显示图片的，它的setImageResource方法要传入一个整型变量。

image.setImageResource(**images**[0]);

注释的快捷键是 ctrl+/

# 2015年8月24日23:29:15

## 1. 修改anctionbar的颜色

<**resources**>  
  
 <!--<style name="AppBaseTheme" parent="Theme.Holo.Light"></style>-->  
 <!-- Base application theme. -->  
 <**style name="AppTheme" parent="Theme.AppCompat"**>  
  
 <**item name="colorPrimary"**>@drawable/BackColor</**item**>  
 <!--Status bar color-->  
 <**item name="colorPrimaryDark"**>@drawable/BackColor1</**item**>  
 <!--Window color-->  
 <**item name="android:windowBackground"**>@drawable/BackColor2</**item**>  
 </**style**>  
 <!--</style>-->  
 <**drawable name="BackColor"**>#20ff1f</**drawable**>  
 <**drawable name="BackColor1"**>#ff42b1</**drawable**>  
 <**drawable name="BackColor2"**>#2c36ff</**drawable**>  
</**resources**>

# 2015年8月25日15:05:36

1. Android 安装程序的时候回出现复制一些东西到手机的错误，需要修改gradle脚本，在module：app里面添加下面的代码

packagingOptions  
 {  
 exclude **'META-INF/LGPL2.1'** exclude **'META-INF/LICENSE'** exclude **'META-INF/NOTICE'** }

# 2015年8月27日23:01:33

在制作抽屉界面的时候用用到DrawerLayout

在里面添加一个ListView

其中为了防止这个抽屉菜单挡住标题栏，需要设置layout\_marginTop属性

<**ListView  
 android:id="@+id/navdrawer"  
 android:layout\_width="200dp"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="start"  
 android:layout\_marginTop="?attr/actionBarSize"  
 android:background="#fa5c32"  
 android:choiceMode="singleChoice"  
 android:divider="@android:color/transparent"  
 android:dividerHeight="0dp"** />

# 2015年9月23日20:30:00

1. 最近做安卓又遇到了低级错误，而且导致很久都没解决问题。

先是加载离线tpk地图，程序没报错，但是一片漆黑，找不到原因。然后试了试加载在线地图，服务器使用arcgis官方的，还是一片漆黑，真是郁闷。然后找到以前写的能正常显示例子。结果在手机里正常显示，这就奇葩了，代码一模一样，怎么会呢。突然想起了在清单文件manifest里面没有设置权限的允许，没有允许网络访问，么有允许访问本地存储。当然不能加载地图了。真是烦了个低级错误。

记住显示地图要在清单文件里设置权限的允许。

<**uses-permission android:name="android.permission.INTERNET"** />  
<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />  
<**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />

# 2015年9月25日11:56:50

## 1. 去掉Android的anctionbar

在values下的styles.xml里面添加

<item name="windowNoTitle">true</item>

其实就是修改主题xml文件，主题名为AppTheme,这个主题的在manifest清单文件里被引用。

2.设置drawerlayout的旋转按钮

主界面使用DrawerLayout布局，里面添加一个lisview来显示抽屉的列表。再用个toolbar来显示按钮标题。

要使用旋转按钮，不用自己添加按钮图标。应该是系统自带了。

在style里面修噶主题。让actionbar不可见。在添加drawerarrowstyle主题。

<resources>  
 <style name="AppTheme" parent="AppTheme.Base"/>  
  
 <style name="AppTheme.Base" parent="Theme.AppCompat">  
 <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="windowNoTitle">true</item>  
 <item name="windowActionBar">false</item>  
 <item name="drawerArrowStyle">@style/DrawerArrowStyle</item>  
  
 </style>  
  
 <style name="DrawerArrowStyle" parent="Widget.AppCompat.DrawerArrowToggle">  
 <item name="spinBars">true</item>  
 <item name="color">@android:color/white</item>  
 </style>  
</resources>

在主界面java代码里面添加变量**private** ActionBarDrawerToggle **drawerToggle**;

用于抽屉状态的切换

drawerToggle = new ActionBarDrawerToggle(this, mDrawerLayout, toolbar, R.string.*app\_name*, R.string.*app\_name*);  
mDrawerLayout.setDrawerListener(drawerToggle);

添加重写函数

@Override  
protected void onPostCreate(Bundle savedInstanceState) {  
 super.onPostCreate(savedInstanceState);  
 drawerToggle.syncState();  
}

效果是没打开抽屉时，按钮是平行的三条线，在打开过程中逐渐旋转到最后的一个箭头。

1. 让activity全屏

在style里面添加

<item name="android:windowFullscreen">true</item>

# 2015年11月25日15:16:06

## 1.关于menu

在menu的xml文件中每个菜单为一个item，这些菜单显示在actionbar上面，如果在actionbar上的按钮(菜单项)有展开菜单，即点击后有个列表，要在下面设置group，在写item

<item android:title="@string/font\_size"  
 android:showAsAction="always|withText"  
 android:icon="@drawable/font">  
 <menu>  
 *<!-- 定义一组单选菜单项 -->* <group android:checkableBehavior="single">  
 *<!-- 定义多个菜单项 -->* <item  
 android:id="@+id/font\_10"  
 android:title="@string/font\_10"/>  
 <item  
 android:id="@+id/font\_12"  
 android:title="@string/font\_12"/>  
 <item  
 android:id="@+id/font\_14"  
 android:title="@string/font\_14"/>  
 <item  
 android:id="@+id/font\_16"  
 android:title="@string/font\_16"/>  
 <item  
 android:id="@+id/font\_18"  
 android:title="@string/font\_18"/>  
 </group>  
 </menu>  
</item>

关于在actionbar上面的显示，actionbar本身空间有限，显示的菜单数量有限。需要设置android:showAsAction属性。如果设置为never，即表示不直接显示，而是通过三个点的隐藏菜单展开后找到菜单，这三个点在小米平板上是显示在右上角，在魅族是显示在下面的。

如果设置为android:showAsAction="always|withText"，则表示一直显示，如果空间不够再隐藏。

2． 程序设置**actionBar**.setDisplayShowHomeEnabled(**true**);

后会闪退，程序是用AppCompatActivity，参考里面是Activity。

# 2015年11月26日18:02:31

1. 将toolbar添加到actionbar，

加入v7包compile **'com.android.support:appcompat-v7:22.2.0'**

在布局文件中加入toolbar

<android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="fill\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="#77db93"  
 app:popupTheme="@style/ThemeOverlay.AppCompat.Light"  
 app:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"/>

其中要使用app，需要添加秘密空间**xmlns:app=**[**http://schemas.android.com/apk/res-auto**](http://schemas.android.com/apk/res-auto)

然后在java代码里面定义一toolbar

Toolbar toolbar;

这个toolbar来自于support.v7.widget.toolbar

import android.support.v7.widget.Toolbar;

然后找到toolbar，并设置到actionbar中

toolbar = (Toolbar) findViewById(R.id.*toolbar*);

setSupportActionBar(toolbar);

这些代码在drawelayout的参考代码里面是一样的。

# 2015年11月27日14:45:08

1.昨天写的笔记还有漏洞，之前一直奇怪为什么自己写的程序总是闪退，计算自己是一句一句抄的斗殴不行，抄自己以前的也不行。经过多次尝试，发现问题在toolbar上面，我在xml里面添加toolbar没有问题，运行时会在actionbar下面添加toolbar，为了把toolbar放到actionbar里面，需要在代码里面添加setSupportActionBar(**toolbar**);我一旦添加这句话，程序就闪退，编译无措，实在是太难解决了。然后网上找了专门关于toolbar的教程，总算找到原因了。需要在style里面设置windactionbar和windownotitle的属性，分别为false和true。

<resources>  
  
 *<!-- Base application theme. -->* <style name="AppTheme" parent="AppTheme.Base">  
 *<!-- Customize your theme here. -->* </style>  
  
 <style name="AppTheme.Base" parent="Theme.AppCompat">  
 <item name="windowActionBar">false</item>  
 <item name="windowNoTitle">true</item>  
 </style>  
</resources>

这下才恍然大悟，发现我原来写的有toolbar的程序都是设置了这两个属性的，要用toolbar需要取消actionbar和title。

在添加好drawerlayout和toolbar、navigationview后，就要设置点击toolbar左上角的按钮来展开drawer。

需要ActionBarDrawerToggle来实现。并且监听。

actionBarDrawerToggle =new ActionBarDrawerToggle(this,drawerLayout,toolbar,R.string.*app\_name*,R.string.*action\_settings*);  
drawerLayout.setDrawerListener(actionBarDrawerToggle);

@Override  
public void onPostCreate(Bundle savedInstanceState)  
{  
 super.onPostCreate(savedInstanceState);  
 actionBarDrawerToggle.syncState();  
}

加了这些代码后，左上角自动回多出一个三横的图标，点击后会自动旋转为箭头。

# 2015年12月5日16:05:34

1. android:uiOptions="splitActionBarWhenNarrow"

当您的应用程序上运行Android 4.0系统（API 14级）或更高级别时，有一个额外的模式可称action bar为“split action bar”。当在一个狭窄的屏幕运行启用split action bar时，会在屏幕的底部出现一个action bar显示所有action item。分裂action bar用来分开action item，确保分配合理数量的空间来在一个狭窄的屏幕上显示所有的action item，而空间留给顶端的导航和标题元素。 使用 split action bar，只需添加uiOptions=“splitActionBarWhenNarrow”，到你的<activity>或<application>清单元素。

# 2015年12月6日22:45:31

1. 在菜单的单击事件里面用switch选择菜单项时注意switch要return true；

2. 向第二个activity传递字符串，用intent

发送：

Intent intent2 = **new** Intent(getActivity(),OtherMap.**class**);  
intent2.putExtra(**"path"**,**othermap2\_path**);  
startActivity(intent2);

接收：

Intent intent =getIntent();  
String res= intent.getStringExtra(**"path"**);

# 2015年12月7日10:15:43

# 快捷键

Ctrl+E，可以显示最近编辑的文件列表

Shift+Click可以关闭文件

Ctrl+F12，可以显示当前文件的结构

Ctrl+Alt+T可以把代码包在一块内，例如try/catch

Alt+Up and Alt+Down可在方法间快速移动

Ctrl+/和Ctrl+Shift+/可以注释代码

Ctrl+O可以选择父类的方法进行重写

通过菜单code下的generator可以快速生成代码，包括构造方法，getter和setter等。快捷键是alt+insert，但好像没用。

2015-12-12 09:28:14

1. 在android studio中格式化代码，也就是使代码布局更规范，比如每行开头对齐，快捷键是ctrl+alt+L,但是这个快捷键会和qq的锁定冲突。

2.Fragment的使用

要使用fragment必须写一个类继承至fragment，然后重写oncreateview方法，返回一个view，在这个方法里面要指定一个布局文件xml来填充fragment，这个布局可以是任意类型的，使用R.layout来找到它，**return** inflater.inflate(R.layout.***fragement\_second***,container,**false**);

**public class** SecondFragment **extends** Fragment {  
 @Nullable  
 @Override  
 **public** View onCreateView(LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {  
*// return super.onCreateView(inflater, container, savedInstanceState);* return inflater.inflate(R.layout.fragement\_second,container,false);  
 }  
}

定义好fragment子类和布局xml文件后，就可以在主activity中添加这个fragment了，可以在主布局xml文件中添加，也可以用java代码添加。

a.在主布局文件中添加

在主布局文件中添加一个fragment标签，用来填充前面的fragment，指明宽度高度等，可以有多个fragment，可以把它看成不用的元件来布局，有两个特殊的属性要设置，一是name，指明fragment子类名称，要加上包名。二是要指明布局文件tools:layout,指向该fragment的布局文件。但是我看视频教程的时候都没有设置第二个属性，我写的程序如果没有设置第二个属性，那么程序闪退，添上后没有问题。

<fragment  
 android:id="@+id/first\_frg"  
 android:name="com.example.guozhang.myapplication.FirstFragment"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="5"  
 tools:layout="@layout/fragment\_first" />

这样在主activity的java代码中就不用添加任何代码就能把fragment添加上了。

使用java代码添加

这种方法比较常用，而且比较灵活。

要用到两个对象FragmentManager和FragmentTransaction。

注意着两个对象，在android.app包和android.support.v4.app包里面都有，前者适用于android3.0系统以上，后者是兼容包，使用与更早的系统。一般使用前者，但不要和后者混用。

要特别注意写这两个类的时候导入的包名不要写错了，否则就是闪退。

还是要定义fragment的子类和xml布局文件，和前面一样。区别就在主activity方法里面。

使用java代码添加fragment一样要在主布局文件里面站位，用于填充fragment。这是一般用framelayout来站位。

<FrameLayout  
 android:id="@+id/first\_frg"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="15"/>

有多个fragment 就可以有多个framelayout

然后就在主activity的oncreate方法里面用java代码添加了。

首先用getFragmentManager()方法获取一个FragmentManager对象，然后用该对象的beginTransaction方法获取FragmentTransaction对象。这两部都是通用的，就得这么写。

也可以连着写，不写对象名。

然后用add方法来添加fragment，注意参数，第一个是站位容器framlayout的id，第二个就是fragment的对象了，使用new来创建。

最后还要用commit来提交。

FragmentManager fragmentManager = getFragmentManager();  
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();  
  
fragmentTransaction.add(R.id.first\_frg,new FirstFragment());  
fragmentTransaction.add(R.id.second\_frg,new SeconFragment());  
fragmentTransaction.commit();

Fragment的替换和删除，和添加一样了都要用到FragmentManager和FragmentTransaction对象了。只是把add方法换成replace和remove方法就可以了。

RadioButton的使用

要用RadioButton，即单选按钮，必须要用RadioGroup容器来包裹它。

在java代码中读取每个按钮是否选中，不直接通过RadioButton,而是通过这个group，有多种方法，可以通过RadioGroup对象的getCheckedRadioButtonIdff来获取，也可以通过选择改变事件来获取，setOnCheckedChangedListener().

<RadioGroup  
 android:id="@+id/radio\_grp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
  
 <RadioButton  
 android:id="@+id/rd1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="RadioButton1"/>  
 <RadioButton  
 android:id="@+id/rd2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="RadioButton2"/>  
 <RadioButton  
 android:id="@+id/rd3"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="RadioButton3"/>  
  
</RadioGroup>

**btnOK**= (Button) findViewById(R.id.***OK***);  
**radioGroup**= (RadioGroup) findViewById(R.id.***radio\_grp***);  
  
**btnOK**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 **switch** (**radioGroup**.getCheckedRadioButtonId())  
 {  
 **case** R.id.***rd1***:  
 Toast.*makeText*(MainActivity.**this**,**"1"**,Toast.***LENGTH\_SHORT***).show(); **return**;  
 **case** R.id.***rd2***:  
 Toast.*makeText*(MainActivity.**this**,**"2"**,Toast.***LENGTH\_SHORT***).show(); **return**;  
 **case** R.id.***rd3***:  
 Toast.*makeText*(MainActivity.**this**,**"3"**,Toast.***LENGTH\_SHORT***).show(); **return**;  
 **default**: **return**;  
 }  
 }  
});

1. 一些注意事项

用final修饰的变量是常量，不能重新赋值

2. DialogFragment的使用

还是要先创建一个DialogFragment的子类，重写onCrateDialog方法，返回一个Dialog对象 。

写一个getInstance方法来传递参数，创建对象，而不是 直接在主程序new一个对象。启动一个DialogFragmet时传递一个参数，用到Bundle和setArguments方法。

**public static** MyDialogFragment getInstance(**int** type)  
{  
 MyDialogFragment myDialogFragment= **new** MyDialogFragment();  
 Bundle bundle = **new** Bundle();  
 bundle.putInt(**"DialogType"**,type);  
 myDialogFragment.setArguments(bundle);  
 return myDialogFragment;  
  
}

这里传递一int参数进去，放到bundle里面，这个参数的名称是**DialogType**，到时候获取参数的时候要用到这个名称。注意用到myDialogFragment.setArguments(bundle);

接着写onCreateDialog方法，这里返回一个AlertDialaog

**return new** AlertDialog.Builder(getActivity())  
 .setIcon(R.mipmap.***ic\_launcher***)  
 .setTitle(getTag())  
 .setMessage(**"DialogType:"**+Dialog\_Type)  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(getActivity(),**"你点击了确定按钮"**,Toast.***LENGTH\_SHORT***);  
 }  
 })  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
 Toast.*makeText*(getActivity(),**"你点击了取消按钮"**,Toast.***LENGTH\_SHORT***);  
 }  
 })  
 .create();

然后在主程序里面定义一个DiaglgFragment子对象

MyDialogFragment **mydialog** = **null**;

在按钮的点击事件或其它事件里实例化这个dialogfragment并显示。

使用getInstance来实例化对象，并传递参数

**mydialog**= MyDialogFragment.*getInstance*(***DialogType1***);

其中***DialogType1***是一个int变量

**if**(**mydialog**!=**null**)  
{  
 **mydialog**.show(getFragmentManager(),**"信息"**);  
}

显示dialog，第一个参数是FragmentManager，第二个参数是一个TAG。在前面用getTAG来设置title，那么对话框的标题就为 “信息”，这也可以传递参数。

# 2015年12月13日20:22:12

在RelativeLayout 布局中

让一个Veiw放在最底部，可以给view赋一个layout\_alignParentBottom属性

**android:layout\_alignParentBottom="true"**

从activity访问fragment

分两种情况：一是在通过xml添加的fragment，主界面有个<fragment/>标签

二是通过java代码添加的fragment

Xml添加的fragment

在主程序里先通过id找到fragment，然后在通过fragment找到view元素，然后就可以方便的操作fragment里面的view了。

找到fragment要用FragmentManager，然后用findFragmentById方法，也可以用FindFragmentByTag，id就是主activity里面fragment标签的id。最后强制类型转换赋值给Fragment子对象，方便后面调用。

BottomFragment bottomFragment= (BottomFragment) getFragmentManager().findFragmentById(R.id.***fragment\_bottom***);

找到后主要通过getView和findViewByid来找到子元素

TextView res= (TextView) bottomFragment.getView().findViewById(R.id.check\_result);

接下来操作就很容易了。

用java代码添加的fragment

这个更方便了，找fragment里面的view是和前面一样的，不一样的是前面要通过findFragmentByI来找到fragment，而这个是在添加fragment到主界面时就添加了fragment对象的，这个对象要先new出来，再用getfragmentmanager.begintrasaction.add来添加。

添加好后就可以用这个fragment对象的getview().findviewbyid(R.id.xxx)来找到子view了。

定义全局变量BottomFragment **bottomFragment**;

在oncreate里实例化**bottomFragment**= **new** BottomFragment();

添加fragment

FragmentManager fragmentManager = getFragmentManager();  
fragmentManager.beginTransaction().add(R.id.***frame\_container***,**bottomFragment**).commit();

找到子view

TextView res=(TextView) **bottomFragment**.getView().findViewById(R.id.***check\_result***);

# 2015年12月14日21:01:25

在fragment中的EditText的键盘事件必须写到oncreateView

中，不能写到onactivitycreated中，写到这里就闪退。不知道问什么。之前写按钮的访问都是些到这里的。

EidtText 的键盘事件

下面的例子是回车按键的事件

**final** EditText editText =(EditText) view.findViewById(R.id.***input\_edit***);  
 editText.setOnKeyListener(**new** View.OnKeyListener() {  
 @Override  
 **public boolean** onKey(View v, **int** keyCode, KeyEvent event) {  
  
 **if**(event.getAction() == KeyEvent.***ACTION\_DOWN***)  
 {  
 **if**(keyCode == KeyEvent.***KEYCODE\_ENTER***)  
 {  
 String content = editText.getText().toString();  
// onAddContentListenner.addContent(content);  
 return true;  
  
 }  
 }  
 return false;  
 }  
 });

Fragment向activity传递值，可以用getactivity找到附着的activity，再找到activity中的view，就比较容易从fragment操作activity了，但是这样不好，fragment和activity的耦合太紧了，不利于代码重用，如过acitivity变了，fragment就得重新修改，这样不好。

改进的办法是利用接口，利用回调函数。定义一个接口，写一个传递值的方法，在定义一个这个接口的引用变量，在fragment的onattach方法中将它附着到activity中，让activity继承这个接口，实现接口中的方法，真正实现的过程就放到了activity中了。

在fragment中调用接口变量的这个还没有实现的方法，就可以传值了。

**public interface** AddContentToFragmentListenner {  
 **public void** addContent(String context);  
}  
**public** AddContentToFragmentListenner **onAddContentListenner**;  
@Override  
**public void** onAttach(Activity activity) {  
 **super**.onAttach(activity);  
 **try** {

//附着到activity中  
 **onAddContentListenner** =(AddContentToFragmentListenner)activity;  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
}

Fragment中调用

**onAddContentListenner**.addContent(content);

activity中的实现：

@Override  
**public void** addContent(String context) {  
  
 Toast.*makeText*(**this**,context,Toast.***LENGTH\_SHORT***).show();  
}

向listframg传入数据

Fragment和fragment通信都是通过activity作为媒介的。

Activity找到fragment通过fragmmanager，

Fragment找到activity通过getactivity或者接口回调，onattach中绑定方接口方法到主activity中。

接下来要通过activity向fragment发送数据

Listfragment自带了一个listview，自己不用定义，非得自定义也可以。

向listview中传递数据需要arraylist 和 arrayadapter 对象。

前者是数据源，后者是适配器

**private** ArrayList<String> **data**;  
**private** ArrayAdapter<String> **adapter**;

**data** = **new** ArrayList<String>();  
**adapter** = **new** ArrayAdapter<String>(**this**,android.R.layout.***simple\_list\_item\_1***,**data**);

arraylist通过add方法添加数据。

Arrayadapter添加arraylist的数据，指定布局方式为***simple\_list\_item\_1***

最后如果arraylist的数据有更新，不用重新neww arrayadapter，而是用notifyDataSetChanged方法

data.add(context);  
adapter.notifyDataSetChanged();

要写listfragment的每一项点击事件，需要在子类中重写onListItemClick方法

@Override  
**public void** onListItemClick(ListView l, View v, **int** position, **long** id) {  
 **super**.onListItemClick(l, v, position, id);  
 Toast.*makeText*(getActivity(),**""**+l.getItemAtPosition(position),Toast.***LENGTH\_SHORT***).show();  
}

# 2015年12月15日00:00:21

此时终于写出了自定义内容的listview了

这下方便属性编辑了。主体是一个listview，每一项包含一个textview和edittext，内容都是通过arraylist传入的，而不是向黄江的在一个fragment里面全部写死了。

Listview通过setAdapter方法添加一个ArrayAdapter对象，可以是子对象，这个对象在添加一个arraylist对象，前者是适配器，后者是数据源。

数据源arraylist是一个泛型，即可以添加任意类型。

Arrayadapter的数据类型要和arraylist一样。

通过定义ArrayAdapter的子类，来自定义内容。需要作为主界面的内部类

**class** ListTypeArrayAdapter **extends** ArrayAdapter<ListTypeData>

ListTypeData类定义了两个字符串，string name和string value;

自定义的ArrayAdapter要构造方法，和重写getview方法。

构造方法用于传递数据源

Getview方法用于每一项的界面布局

下面的布局是java代码来添加线性布局的，也可以用xml来定义布局。

**class** ListTypeArrayAdapter **extends** ArrayAdapter<ListTypeData> {

**public** ListTypeArrayAdapter(Context context, ArrayList<ListTypeData> datatypes) {  
 **super**(context, 0,datatypes);  
 }  
  
 @Override  
 **public** View getView(**int** position, View convertView, ViewGroup parent) {  
*// return super.getView(position, convertView, parent);* LinearLayout line = **new** LinearLayout(MainActivity.**this**);  
 line.setOrientation(1);  
 TextView name = **new** TextView(MainActivity.**this**);  
 EditText value = **new** EditText(MainActivity.**this**);  
 name.setText(**data**.get(position).getName());  
 value.setText(**data**.get(position).getValue());  
 line.addView(name);  
 line.addView(value);  
 **return** line;  
  
 }  
 }

写好ArrayAdapter子类后，就可以导入数据和给listview设置数据了。

**listView** = (ListView) findViewById(R.id.***list\_exp***);  
  
 **data**= **new** ArrayList<>();  
 **data**.add(**new** ListTypeData(**"姓名"**, **"JACK"**));  
 **data**.add(**new** ListTypeData(**"年龄"**, **"18"**));**listTypeArrayAdapter** = **new** ListTypeArrayAdapter(**this**,**data**);  
 **listView**.setAdapter(**listTypeArrayAdapter**);

# 2015年12月15日01:39:16

Activity向fragment传值的时候，一定要等fragment的上的view都创建完了才能调用，否则view为null，就给它传值，或出错，导致闪退。

一般要在dialogfragment show方法完了后在传值，也可以在之前传值，但不要直接给控件传值，因为它还没实例化，非得这样看，可以在fragment里面定义几个变量，给这几个变量传值，然后在oncreateview方法里面控件初始化完了后把这几个变量值赋值给控件。这样不会出错。

2015年12月15日20:41:37

1. 又遇到一个比较重要的问题

到现在显示要素的属性的功能已经实现了，但是至少显示，编辑的还没有实现。

就在这个显示属性的地方有个小问题我是把属性表添加到dialogfragment的lisview中的，属性一开始能正常显示，但是listview往下滑到某个位置的时候就程序就闪退了。不知道是什么原因，但很明显是给直接操作了某个值为null的变量，几乎所有的闪退都是操作了没有实例化的对象。

我知道自己刚刚只写了属性显示的部分，因此在这部分找原因，调试程序设置断点，发现读取属性值到Map<String,Object>中的有个字段是null的，在arcgis中打开属性表，这个字段是没有赋值的。但是把它添加到Arraylist中的时候直接把他tostring了，但没有报错啊，我尝试不把值为null的字段添加到属性表中，结果就没有闪退了。

2. 直接用Feature接口下的Map<String, Object> getAttributes();方法虽然能获取属性表，但仅仅是读取，不能修改设置啊。

GeodatabaseFeature 继承至Feature，不仅能获取属性表，还能修改设置，应该用它。黄江的代码就是这样用的。

FeatureLayer featureLayer = (FeatureLayer) *mapView*.getLayerByID(getTemplatePicker().getSelectedLayer().getID());

GeodatabaseFeature gdbFeatureSelected = (GeodatabaseFeature) featureLayer.getFeature(**currentFeatureID**);

这样可以直接获取getAttributes，而不用先获取FeatureTable，再通过这个表找到Feature，这样走弯路了。

Map<String, Object> attributes\_res= gdbFeatureSelected.getAttributes();  
**attributedata** = **new** ArrayList<>();  
**for**(String s:attributes\_res.keySet())  
{  
 **attributedata**.add(**new** AttributeDataType(s,attributes\_res.get(s)));  
}  
  
**attributeDialogFragment** = **new** AttributeDialogFragment();  
**attributeDialogFragment**.setList\_Adapter(**new** AttributArrayAdapter(OfflineEditorActivity.**this**,**attributedata**));  
**attributeDialogFragment**.show(getFragmentManager(),**"属性表"**);

# 2015年12月16日13:10:35

1. 关于android 的listview获取子view的问题

我是通过自定义arrayadapter传递了每项包括textview和edittext的view到listview中的。想在edittext中修改后再保存，问题就出在如何获取listview中的某一项的edittext，找了很多地方，发现可以用getCount方法来获取listview的总的item的数量，再用getChildAt获取子view，注意这个子view只是包含textview和Edittext的容器是个Linearlayout，昨天就直接把这个子view强制转换为Textview了，结果就是闪退。应该先强制转换为LinearLayout在用它的getChildAt方法获取textview或edittext。这里textview和edittext是通过java代码创建并添加到LinearLyaout中的，因此没有id，不能通过findviewbyid找到，我尝试在arrayadapter中通过加载xml来布局，就有id了，但是程序闪退，还是直接用java代码来添加。由于没有id，但是LinearLayout的getchildAt（）方法参数是index，直接写0或1就能找到textview和edittext了。

**for**(**int** i =0;i<**attribute\_list**.getChildCount();i++)  
{  
 LinearLayout layout = (LinearLayout) **attribute\_list**.getChildAt(i);  
 TextView tx = (TextView) layout.getChildAt(0);  
 String res1 = tx.getText().toString();  
}

注意for循环中getChildCount，这里不是getCount，两者有区别，getCount是获取listview中的所有item数量，包括可见和不可见内容。GetChildcount只能获取可见部分的item数量，listview如果数量太多是可以上下滑动的，就存在可见和不可见部分。如果用getchildcount，再用getchildat就只能找到可见内容。上下滑动，可见内容就会变化，这时用上面的代码读取的内容就会不一样。如果for循环是用的getcount下面用的是getchildat也会出错，因为getchildat只能读取可见内容，这样会导致索引超出界限，结果就是闪退。

要想获取listview所有的item，还得用getcount，但下不能用getchildat了，得先找到adapter，在用adapter的getview（）方法，这个getview方法有个索引参数，通过索引就能找到任何想要的子view了。

**for**(**int** i =0;i<**attribute\_list**.getCount();i++){  
*// LinearLayout layout = (LinearLayout) attribute\_list.getChildAt(i);* LinearLayout layout1 = (LinearLayout) **attribute\_list**.getAdapter().getView(i,**null**,**null**);  
 EditText ed = (EditText) layout1.getChildAt(1);  
  
 res += ed.getText().toString();  
 res +=**" "**;  
 }

getView方法本来就在创建arrayadapter的时候重写了，目的就是为了找到想要的子view。

现在有又新的问题了，子view是找到了，编辑EditText后输出新的内容没变，重新上下滑动listview，刚输入的内容变成别的字段的内容了，非常混乱。经过分析，在输出所有内容的时候，是通过getview来找到每一个子view的，getview又是通过数据源arraylist来获取数据的，编辑edittext后如果没有更新这个arraylist，读出来的始终是旧的数据。

现在的问题是怎么在edittext编辑后就更新这个arraylist呢，可以给每个edittext设置一个输入改变事件addTextChangedListener，在arrayadapter的getview方法中设置这个方法。事件里面要处理这个edittext，那么这个edittext必须用final修饰为全局的，之前没有这么弄，虽然数据有更新，但是都是乱的。

**public class** AttributArrayAdapter **extends** ArrayAdapter<AttributeDataType>  
 {  
**public** AttributArrayAdapter(Context context,ArrayList<AttributeDataType> data) {  
 **super**(context, 0,data);  
 }  
  
 @Override  
 **public** View getView(**final int** position, View convertView, ViewGroup parent) {  
Log.*e*(**"Tag"**,**"getView方法"**);  
  
 LinearLayout line = **new** LinearLayout(OfflineEditorActivity.**this**);  
 line.setOrientation(LinearLayout.***VERTICAL***);  
  
 *//必须是final，否则出错* **final** TextView key = **new** TextView(OfflineEditorActivity.**this**);  
 **final** EditText value = **new** EditText(OfflineEditorActivity.**this**);  
  
  
 key.setText(**attributedata**.get(position).getKey());  
  
 *//有些字段的值为空，直接让null tostring会闪退* **if**(**attributedata**.get(position).getValue()==**null**)  
 {  
 value.setText(**""**);  
 }  
 **else** {  
 value.setText(**attributedata**.get(position).getValue().toString());  
 }  
value.addTextChangedListener(**new** TextWatcher() {  
 @Override  
 **public void** beforeTextChanged(CharSequence charSequence, **int** i, **int** i1, **int** i2) {  
  
 }  
  
 @Override  
 **public void** onTextChanged(CharSequence charSequence, **int** i, **int** i1, **int** i2) {  
  
 }  
  
 @Override  
 **public void** afterTextChanged(Editable editable) {  
  
 Log.*e*(**"TAG"**,**"position:"**+position);  
 Log.*e*(**"TAG"**,**"value:"**+value.getText().toString());  
 **attributedata**.get(position).setValue(value.getText().toString());  
 }  
 });line.addView(key);  
 line.addView(value);  
**return** line;  
}

设置输入改变事件后，原始数据就更新了，再读取就是新的了。

# 2015年12月18日00:25:42

要使用toolbar，有几点得注意

activity的主题必须是noactionbar的，或者application的主题设置为没有ationbar

activity必须继承自AppCompatActivity,来自support.v7包，因此得加入v7包，如果没有继承自这个appcompatactivity，那么setSupportActionbar方法就不识别。

# 2015年12月18日21:49:26

在java代码中获取xml资源要用getResource方法

比如获取color里面的颜色

**toolbar**.setTitleTextColor(getResources().getColor(R.color.***textColor***));

又发现一个重要问题，这个toolbar根本就没有掌握啊。之前写的drawlayout里面用到了toolbar，和今天看到的写法不一样。区别在toolbar上加载的按钮。

首先都是要把主题设置为noactionbar，然后在主布局里定义一个toolbar，在java里面findviewbyid找到toolbar，接下来就不一样了。

之前的方法：用setSupportActionBar(toolbar);把toolbar加载到actionbar位置，然后drawerlayout的相关设置可以包含toolbar。然后就没有了。Toolbar上的按钮来自menu，和没有定义toolbar一样，原来菜单是怎么加载的，现在就是怎么加载的。onCreateOptionsMenu实现了菜单按钮的加载，所以这样toolbar和actionbar没有多大区别。点击事件还是写到onOptionsItemSelected里面。

今天新看到的方法：不调用onCreateOptionsMenu方法来加载按钮。定义和找到toolbar都是一样的，但是没有用setSupportActionBar(toolbar)方法，然后是初始化的时候加入**toolbar**.inflateMenu(R.menu.***menu\_main***);

点击事件写到toolbar的setOnMenuItemClickListener方法里面

**toolbar**.setOnMenuItemClickListener(**new** Toolbar.OnMenuItemClickListener() {  
 @Override  
 **public boolean** onMenuItemClick(MenuItem item) {  
 **int** menuItem = item.getItemId();  
 **switch** (menuItem) {  
 **case** R.id.***action\_search***:  
 IsAdapterEmpty();  
 **initiateSearch**.*handleToolBar*(MainActivity.**this**, **card\_search**, **toolbar**, **view\_search**, **listView**, **edit\_text\_search**, **line\_divider**);  
 **break**;  
 **default**:  
 **break**;  
 }  
 **return false**;  
 }  
});

两种方法的xml菜单布局都是一样的。包要用android.support.v7.widget.Toolbar;

还有一个包里面也有toolbar，是android.widget.toolbar

# 2015年12月19日00:50:45

最近在学android的SearchView,我是在网上看到别人的代码里面有用到searchview，效果比较好，想学一下，用来写到arcgis for android上查询要素，想法还挺好。

我打算把searchview写到一个dialogfragment里面。参考例子里面的SearchVeiw是用java代码来加载的，而不是用xml来定义的，并且它是附着到toolbar上面的，一开始不明白为什么这个searchveiw就放到了toolbar的位置上。例子里面还定义了两个toolbar，只不过把需要用的toolbar用过setsupportactionbar把它附着到actionbar上面了，那么这个toolbar就相当于actionbar了，然这个例子的searchview是默认通过actionbar上的menu来加载的，因此就加载到了这个toolbar上了。由于有两个toolbar，这两个toolbar位置还不一样，又让我明白了一个道理，网上说toolbar比actionbar灵活，原来是在这里体现啊，数量可以多个，位置可以任意，内容可以自定义，但actionbar好像一activity只能有一个，可以把一个toolbar附着到actionbar上面。使用actionbar传统的menu加载方式来给这个toolbar加载按钮。

回到searchview，这个例子里的searchview是在onCreateOptionsMenu方法

里面加载的，直接绑定了一个menu菜单

@Override  
**public boolean** onCreateOptionsMenu(Menu menu) {  
 getMenuInflater().inflate(R.menu.***menu\_main***, menu);  
 MenuItem searchItem = menu.findItem(R.id.***action\_search***);  
 **mSearchView** = (SearchView) MenuItemCompat.*getActionView*(searchItem);  
 **mSearchView**.setOnQueryTextListener(**this**);  
 **if** (**searchRequest** != **null**)  
 **mSearchView**.setQuery(**searchRequest**, **true**);  
 **return true**;  
}

这个menu菜单有个比较特殊的属性

<**item  
 android:id="@+id/action\_search"  
 android:orderInCategory="100"  
 android:title="@string/action\_search"  
 app:showAsAction="always"  
 app:actionViewClass="android.support.v7.widget.SearchView"**/>

加载就完成了，关于事件方面的要继承接口，详细看代码。

然后我试着按照他的来写，刚开始就出问题了，我是要写到dialogfragment中，但是dialogfragment没哟菜单啊，我怎么加载啊，只好网上找searchviwe的例子，github上搜了半天，要么是超级复杂，看不懂，要么是用按钮假编辑框自己写的，很无语。又在百度上搜索，资料有限，找到一个比较简单的，是在xml里面直接定义一个searchview，然后在java代码里面用findviewbyid来找到，我把它写到dialogfragment的布局中了，然后在java代码中定义了一searchview变量，再用findviewbyid找到它，编译没错，代码一运行就闪退，真是无语了，然后我把java代码中searchview相关的都删掉，保留xml中的searchview，再运行，结果就可以了，不闪退了，但是这样不行啊，我不能再java代码中找到它我怎么设置它的事件，就关根本不能继续往下写了。 想了下，刚看到的例子是直接在activity中定义searchview的，我的是在dialogfragment中，是不是不一样啊。于是我重新建立了一工程，只在activity中定义searchview，用xml定义，java找到，结果悲剧重演，还是闪退。代码跟网上的一样啊，我的怎么就不行呢，然后网上搜索未名searchviwe加入后程序闪退，还真找到了原因。是searchview用的包名不对。Searchview存在于两个包中android.support.v7.widget.SearchView和android.widget.SearchView;

第一个用java代码和menu加载的用的是support包，只能加载到activity中的toolbar上。而android.widget.SearchView包是通用的，用xml定义，可放到任意位置。我的错就出在本应该用android.widget.SearchView包的，却用到了

android.support.v7.widget.SearchView包，在定义SearchView的时候，IDE回提示要用哪个包，由于没有注意直接选择了第一个不对的包，才导致错误。

就像之前的fragment也有不同的包，一个是support的兼容包，一个是高版本的包

总结：使用控件的时候一定要注意使用正确的包，程序闪退，除了对象没有实例化外还有就是包用错了。

# 2015年12月19日10:50:43

1. 关于style

<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">  
 *<!-- Customize your theme here. -->* <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
</style>

<style name="AppTheme" parent="android:Theme.Holo.Light">  
 *<!-- Customize your theme here. -->* <item name="colorPrimary">@color/colorPrimary</item>  
 <item name="colorPrimaryDark">@color/colorPrimaryDark</item>  
 <item name="colorAccent">@color/colorAccent</item>  
  
</style>

<**style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar"**>  
 *<!-- Customize your theme here. -->* <**item name="colorPrimary"**>@color/colorPrimary</**item**>  
 <**item name="colorPrimaryDark"**>@color/colorPrimaryDark</**item**>  
 <**item name="colorAccent"**>@color/colorAccent</**item**>  
  
</**style**>

本来想自定义actionbar，通知栏的颜色的，可是在主题里面设置**colorPrimary后，发现不起作用，一样是黑色的，后来才发现用的主题不对，之前一直用的是holo主题，这是比较早的了，后来多了个5.0的主题AppCompat，在这个主题里面是可以修改颜色的。**

# 2015年12月19日17:35:182

Toolbar本身继承与ViewGroup，是个布局容器，里面可以放很多东西。之前的方法都是什么设置title，icon，加载menu，这些都还不够，否则就和actionbar没有区别了。在xml 的toolbar的定义里添加别的view，可以扩展内容，比如添加一个TextView，布局设置为剧中，可以让它显示在toolbar的中间，之前的方法设置titile都只能显示在左边。这下明白了，为什么网上常用软件设置界面左上角有个向左的图标，紧接着是“返回”汉字，点击任何一个都可以返回。之前的方法给toolbar设置一个icon，并设置icon的点击事件，功能上能实现，但是界面效果太差，icon占用的控件太大和title不协调。

现在可以在toolba下面加一个imageview和Textview，来实现这个功能。

**要用**setSupportActionBar，那么activity必须是appcompatactivity，否则不识别。

现在的问题是，我想修改主界面actionbar颜色，它没哟使用toolbar，用的是activity而不是appcompatactivity，因此要自己添加一个toolbar，并在style里面设置为noactionbar，这样toolbar是成功加载上了，颜色也修改了，但是软件有个编辑的功能用到了actionmode功能，就是在actionbar上面临时放置菜单按钮，由于现在是没有actionbar，而是toolbar，因此加载actionmode后在toolbar上面多出了临时的actionbar，非常不好看

2． 使用compat主题很容定制actionbar等在颜色，使用holo主题，默认只有黑白两种颜色，我把自己的程序写成compat的主题，可是程序总是崩溃，找不到原因，只好改回holo主题，网上找到了使用holo主题定制颜色的方法。

<**style name="CustomActivityTheme" parent="@android:style/Theme.Holo.Light.DarkActionBar"**>  
 <**item name="android:actionBarStyle"**>@style/MyActionBar</**item**>  
</**style**>  
<**style name="MyActionBar" parent="@android:style/Widget.Holo.ActionBar"**>  
 <**item name="android:titleTextStyle"**>@style/TextAppearance.Holo.Widget.ActionBar.Title.Own</**item**>  
 <**item name="android:background"**>@color/colorPrimary</**item**>  
</**style**>  
<**style name="TextAppearance.Holo.Widget.ActionBar.Title.Own"  
 parent="@android:style/TextAppearance.Holo.Widget.ActionBar.Title"**>  
 <**item name="android:textColor"**>#FFFFFF</**item**>  
</**style**>

修改actionmode的颜色，actionmode的颜色和主题有关，但可以自定义，不自定义要么白色，要么黑色。自定义的方法是在主题中添加

<**item name="android:actionModeBackground"**>@color/yellow</**item**>

还可以定制actionmode的关闭按钮，就是左边默认为勾的按钮，点击会关闭actionmode

<item> name="android:actionModeCloseDrawable">@drawable/ic\_launcher</item>

# 2015-12-20 22:19:00

Drawerlayout中默认在左边可以通过右滑动打开抽屉，可以通过代码关闭。

**drawer**.setDrawerLockMode(DrawerLayout.***LOCK\_MODE\_LOCKED\_CLOSED***);

Drawerlayout中可以不要toolbar，直接将总布局设置为drawerlayout，下面只能有两个view，第一个是主界面，第二个drawer的内容。

总布局drawerlayout要设置一个id，在oncreate中用findviewbyid找它，然后在菜单或按钮事件中打开就可以，也不用写别的代码就可以了。

*drawer.openDrawer(Gravity.LEFT);*

但是我在我的离线编辑程序中想添加drawerlayout，但是总是程序崩溃，然后写了个测试程序，都可以正常运行，我的就不行，主题，库版本，编译版本都一样啊，不断测试，不知道怎么回事，进入主程序不崩溃了，但是一旦点击按钮，执行打开drawer代码，程序就崩溃了，整的我都快崩溃了，这么简单的代码，和正常运行的也没有区别啊。接着我试着在主界面加一个button，然后通过这个button的点击事件来打开抽屉，没想到，结果就成功了，然后我把这个button去掉，用菜单打开抽屉，又成功了，真是无语，莫名其妙。也许是我在drawerlayout中加一个linearlayout吧。

原来是NavigationView作怪，我刚刚测试的能行的是用的listview，现在我换回NavigationView去不行了，

原来NavigationView要用appcompat的主题，我的程序不能改成compat的主题，一用就崩溃，只能用holo的主题。那么还不能用NavigationView了。

Actionbar左边有个应用程序图标和文字，默认点击没有反应，可以设置它能点击，也可是设置带一个向左的箭头图标，用于返回主界面的按钮上。

设置能点击的方法是getActionBar().setHomeButtonEnabled(true);

然后在菜单的点击事件里找到id 为android.R.id.***home***:

**case** android.R.id.***home***:  
 **drawer**.openDrawer(Gravity.***LEFT***);  
 **return true**;

如果是setDisplayHomeAsUpEnabled（true）就是使能并返回。

在菜单的事件中打开和关闭抽屉

**case** android.R.id.***home***:  
 **if**(!**drawer**.isDrawerOpen(**list\_drawer**))  
 {  
 **drawer**.openDrawer(Gravity.***LEFT***);  
 }  
 **else** {  
 **drawer**.closeDrawer(Gravity.***LEFT***);  
 }  
  
 **return true**;

其中list\_drawer是抽屉容器的id，这里是个listview

# 2015年12月21日11:52:58

Drawerlayout的抽屉部分可以不是listview，之前尝试navigationnview，不行因为主题不是appcompat，现在尝试抽屉用relativieLayout，发现程序闪退，然后用fragment，还是闪退，多次尝试都是闪退，只好还是用回listview，我都怀疑是不是只能用listview，如果是，那岂不是太不行了。网上找资料，说的是可以任何布局，有的用的是Linearlayout，有的用的是Framelayout，但是人家都设置了gravity属性，我之前好像也设置了，但是还是不行啊。

现在我用framelayout，并且设置了gravity属性为left，没想到成功了。

<FrameLayout  
 android:id="@+id/left"  
 android:layout\_width="240dp"  
 android:layout\_height="match\_parent"  
 android:layout\_gravity="left"  
 >  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:text="hello"/>  
  
</FrameLayout>

这个效果还是半透明的，效果很好。

程序默认打开drawer后在按后退键无效，不能关闭drawer，也不能退出程序。重写onbackpress也不行，只能手动关闭drawer后用后退键退出程序了。

先在遇到一个问题，地形图是加载上了，但是一编辑的时候就闪退，原因就是软件只能编辑geodatabase的数据，这里多了个shape，也是FeatureLayer，不能编辑，因此出错。还得修改。如果初始时没有加载地形图shape文件，那么程序是不会崩溃的。

# 2015年12月31日21:12:59

让android程序不出现在最近应用程序列表里面

要在manifest.xml里面配置activity

 android:excludeFromRecents="true"

2016年1月5日22:07:10

Android studio svn 新添加一文件后，会变成红色，标示这个文件没有添加到svn，这死提交commit是没有用的，需要把它添加到svn，右键它，选择subversion下的add to vcs，然后文件名会变成绿色，这是标示添加到了svn，但没有提交，提交后会变成黑色，这是正常颜色。已经提交的 文件，如果修改后会变成蓝色，提交后会变成黑色。

# 2016年1月6日09:50:57

刚刚写程序，加载百度地图，遇到了太多错误了，都是些不该犯的低级错误

Sdk开房包是正确导入了，库也添加了。百度地图要联网，manifest里面没有添加允许权限的代码。没有申请开发AK密钥。

刚刚又犯了个低级错误，添加toolbar，也设置了不要actionbar和title的主题，但是toolbar就不不显示，后来添加setsupportactionbar方法也不行，最后才发现是主布局是Relativelayout，先写了toolbar，再写了baid的mapview，后者会将前者覆盖啊，应该用linearlayout，或者设置布局位置啊。这个错误不应该。

实践表明，用了setsupportactonbar（toolbar）方法后toolbar上回显示应用程序名称，不用这个方法就不会。

# 2016年1月7日14:29:57

哎，无语了啊，最近把官方的鹰眼轨迹服务程序移植到我自己的程序里面，可是点击开始按钮，理论上应该启动轨迹service，可是死活不能启动，一点反应都没有，仔细检查代码，也没有错啊，难道是servieid不能和之前的冲突，然后我新申请了一个serviceid，依然没有用。难道了密钥ak不行，也不应该啊，百度地图都能正常显示，ak应该没有问题。然后还是怀疑代码没有弄明白，于是我直接拷贝所有代码过来，基本上不怎么修改，只是去掉了不要的功能，包名修改了，可以说轨迹服务的代码和官方的一模一样，可是结果很惨，还是不行。

最后我只好打开百度官网的鹰眼帮助，一步步看，才发现了问题所在，没有在manifest里面配置service，要添加百度的轨迹服务service才行，

<service  
 android:name="com.baidu.trace.LBSTraceService"  
 android:enabled="true"  
 android:exported="true" >  
</service>

加上后，果然成功了。

android:launchMode="singleInstance"

设置activity的启动模式为上面的模式后，在主activity中打开这个activity，然后后退能退出，也能重新进入，但是当打开这个activity后，按home键退到主界面，然后打开最近任务，直接进入这个程序，打开的是这个activity，接下来就是奇怪的地方了，再按后退键，程序直接退出了，不回到主activity了。

# 2016年1月8日17:22:15

不容易啊，这个问题困扰我好几了。一直在想activity的变量退出后怎么保存，再次进入是不是又重新回到最初状态了，还是上次保留状态，或者是后台更新的值，我需要activity再次进入时打开上次退出时的状态。不知道怎么弄，一直研究activity的生命周期，网上说onCreate方法只调用一次，可以我调试是每次都调用了，我在onCreate方法里面实例化对象了，每退出后重新进入，是不是又要重新实例化啊，那样是不行的啊，对象变成新的对象了，不能控制之前的对象了。我每次重新进入activity

后，点击关闭百度鹰眼服务的按钮，都提示没有开启服务，如果是第一次进入，开启后，没退出activity点击关闭是正常关闭的，但是退出后再进入关闭就不行了，关闭是不行，但是开启是可以的，第一次进入开启成功，退出后重新计入开启会提示已经启动，如果是又实例化了一点对象，那么会提示开启成功。非常奇怪。

我又重新写了个测试程序，写了几个变量，点击按钮自增，输出值，退出后再进入，又回到初始值了，那么应该是程序又开辟了新的空间来存储值，而不是用原来空间的变量，想到这里就会想到static了，把变量定义为static全局变量，就不会实例化多次了。我尝试着在变量前添加static，在按钮点击事件里自增，退出后，重新进入，输出这个变量值，是最新的值，而不是初始值。

一个类中static修饰的变量具有全局性，这个变量不属于某个对象，这个类实例化对象的时候，不会为它开辟新的空间，无论new多少个对象，

# 2016年1月9日09:28:15

百度鹰眼轨迹服务有个bug，就是activity退出后，再次进入，按停止按钮不能停止，回调消息提示服务未开启，一开始我怀疑是重新进入后stopTraceListener又被实例化了一次，并不是上次退出前的对象了，才导致不能停止的。很快我就否定了这个想法，因为，我退出activity后，重新进入，点击开始按钮，收到已经开启的消息，而不是重新开启的“success”消息，说明startTraceListener是完全正常的，那么我的startTraceListener和stopTraceListener变量定义是不是又说明区别呢，我发现都是一样的，访问修饰符一样，都有用static，但都在onCreateView里面重新new了。

我读取了这两个对象的hashcode，发现，每次重新进入后它们的hashcode都变了，因此，这个两个对象实例化多次应该都是能用才行，可实际上，只有start能用，stop不行，通过尝试，发现重新进入后，不能stop，但是这时再次点击开始，重新启动服务，然后点停止按钮，就能成功停止了。

# 2016年1月13日00:48:31

我他妈的也是醉了

写个百度的定位程序，从下午搞到现在，总算完成了，这么简单东西，废了这么大的劲，哎。

首先出现的问题是点击启动定位按钮，没反应，根本就没定位，怀疑是不是按钮点击事件没写好，又在点击里面写了log，发现按钮是写对了的，在接收定位的listener里面的回调方法里面打断点，发现根本就进不了这个方法，无语了。只好看官方的帮助文档，忽然就发现，官方程序在maniifest里面都配置了service，而我的没有配置，和之前轨迹服务一样，忘了配置servie导致痛苦了很久。把它配置好久能定位了

<service  
 android:name="com.baidu.location.f"  
 android:enabled="true"  
 android:process=":remote" >  
 <intent-filter>  
 <action android:name="com.baidu.location.service\_v2.2" >  
 </action>  
 </intent-filter>  
</service>

这样我在回调函数里就能接收到数据了，非常兴奋啊，总算弄出来了，可是转眼间就发现了问题，为什么定位的数据大部分都是null，定位失败，经纬度要么是科学计数法要么是0，其他的都是null，无语了，又仔细研究官方程序，没发现有什么区别啊，就差一字不差的抄代码了。死活找不到原因。一开始我怀疑过sdk包是不是全的，但没有太在意，我只是看了我的so那些文件的总大小，发现还比官网的例子里面的大，我很快就否定了开发包没拷全的想法了，实在是找不到原因，开发指南也没有，只好逛论坛，突然发现一个帖子，是说回到消息里面错误代码的含义，比如61的gps定位成功，我的大部分是61，但是经纬度都是0，有是162，论坛里说是服务端定位失败，不知道什么原因，下面网友很多也是162，有人就说了是so库不正确，看到这里，我马上检查我的库，这下才恍然大悟，原来是我根本就没有导入定位的so库，官方的该例子只是定位的功能，当让so总大小要比我的包含轨迹的要小了。把定位的so库导入后，重新编译下载，立马成功，这时都到深夜了。

# 2016年1月13日16:22:12

今天遇到一个新的问题，关于context

我是在百度添加overlay的marker的infowindow里面点击一按钮，然后弹出一个alertdialog，来判断是否删除该记录，这里alertdialog有个参数是context，我不知道怎么设置，设置为this，直接报错，因为这个点击事件已经是内部类里面的内部类了，this表示点击的listenner对象，不行 。用getapplicationcontext没报错，但是程序运行闪退。

经过一番尝试后发现要用到该activity的类，比如我的该段代码是在TrackActivity.java里面，应该给context赋TrackActivity.this,这时就没错了。

AlertDialog dialog = **new** AlertDialog.Builder(TrackActivity.**this**)  
 .setTitle(**"提示"**)  
 .setMessage(**"是否删除该记录"**)  
 .setPositiveButton(**"确定"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
*// markerlist.remove(marker);  
// marker.remove();* }  
 })  
 .setNegativeButton(**"取消"**, **new** DialogInterface.OnClickListener() {  
 @Override  
 **public void** onClick(DialogInterface dialog, **int** which) {  
  
 }  
 })  
 .create();  
 dialog.show();

# 2016年1月14日19:34:16

制作缩略图可以用Thumbnails类，不用自己写。

2016年1月14日23:15:57

当scrollview中有edittext的时候，进入activity会自动弹出键盘，需要禁止它自动弹出。

在manifest中的对应activity中添加

android:windowSoftInputMode="stateHidden|stateUnchanged"

在Edittext的属性中添加

**android:focusable="true"  
android:focusableInTouchMode="true"**

这样就可以了，如果只是把foucusable设置为false，那么是不弹出键盘，也是一直都吧弹出，无法输入文字。

# 2016年1月15日00:35:30

将contextmenu的实现由长按变为短按，首先要设置该view的oncliecklistener方法，在这个方法里调用openContextMenu(v);方法

# 2016年1月15日11:19:38

1. Dialogfragment点击屏幕空白处，不退出的方法

在dialogfragment的onCreate方法里写：

@Override  
**public void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 **this**.setCancelable(**false**);  
 **int** style = DialogFragment.***STYLE\_NORMAL***, theme = 0;  
 setStyle(style,theme);  
  
}

只能在这里写，不能再onCreateView和onCreatedialog方法里面写，

这时不仅点击屏幕不能退出，后退键也不能退出，需要在dialogfragment里面手动退出，在里面添加个按钮，在点击事件里面写dismiss();

# 2016年1月17日01:30:13

1. 到现在，总算把sqlite基本框架搭上了，能增删改查了。

我是在github上找到了一个例子，仿照它来写的，百度上搜索的关于sqlite的博客都写得太垃圾了，要么长篇大幅，不知所云，要么只是贴代码，什么都不解释，还有就是简单讲几句，又没有代码，也没法参考。然后我在github上搜索，先是找到了一个，看上去比较好的例子，但是打开代码一看，吓了一跳，太多了，光是代码文件数量就达到了19个之多。完全没法看。只好重新搜索，总算找到一个比较不错的，好像和之前看到某博客上讲得差不多，是继承SQLiteOpenHelper来写的，主要代码只有这一个，非常适合参考。

继承SQLiteOpenHelper，需要重写onCreate方法，在这里面创建表，不是数据库，在这个子类的构造方法里面创建数据库，使用super调用了父类的创建方法

public DatabaseHandler(Context context) {  
 super(context, DATATBASE\_NAME, null, DATABASE\_VERSION);  
}

这个子类里面很多方法都要自己写，完全依赖于自己要往数据库里面写的数据的结构，要在这里定义表名，数据库名，所有的列名，还有一个arraylist来保存这个表，用于查询时输出。然后写添加一项，查找所有项，删除，更新的方法，所有这些都写到这一个文件里面，紧跟数据类型有关，比如我保存一个采集点的属性，封装了一个类Collectpoint\_Attribute，这个类里面的所有属性字段就喝表相关了，唯一区别是数据库表有个主键，是int的自动增长的，参考例子是用它作为标示来查询的，我不用他，我是用采集点属性里面的一个uuid来匹配查询，因此我不用重新修改Collectpoint\_Attribute类给它添加一个id字段，根本不管这个id。给表添加一项的时候，ContentValues也不用put这个id字段，因为它是自动生成的。

还有一点是，定义了DatabaseHandler这个类之后，怎么使用的问题，我看到参考例子里面，是想什么时候使用，就什么时候new一个它的对象出来，也不拍冲突，或者会因为多个对象儿导致多个表的问题。这个问题应该不存在，不管new多少次，这个类就只能操作一个表，因此可以多次实例化，事实也是如此。

数据库写完了，把相关操作放到采集点上，如添加一个点后，往数据库里面添加一条数据，删除点，就删除一条数据，还有编辑点，在onActivityResult里面跟新表，现在是移动点的地方还没有修改数据库。

接下来就是测试了，我在主界面一个按钮里面调用了获取表中记录数量的方法，想输出一下一共记录 了多少次数据，但是一调用这个方法，程序就闪退，提示说，使用了一个已经关闭的数据库，刚开始弄，不太明白，网上查，说的是不应该关闭，不关闭就行了，由于代码是抄的，自己也不太明白是怎么回事，当时还在想，是不是不能实例化多个这个类啊，但是例子里面都是可以的啊，然后我直接在例子里面修改，添加了一个按钮，调用输出表中项目数量的方法，也是闪退，不应该，然后仔细看代码，这才恍然大悟，当时抄的时候就有点怀疑，它用

String coutnQuery ="SELECT \* FROM "+TABLE\_POINTATTR;  
Cursor cursor = db.rawQuery(coutnQuery,null);

来查询，返回的是cursor，然后立马就用了cursor.close()方法，然后输出cursor.getCount();

应该是先获取了数量，再close，改过来后就对了，原来是例子也有bug啊。

昨天下午写，昨天半夜时留下的bug，弄了一下午，晚上才写数据库，写那个编辑界面返回时判断是否修改过数据，都写了好久。

# 2016年1月17日12:30:03

昨晚还有一点忘记写了，就是sqlite里面创建表的语句，在列名的后面要加字段类型和主键标识等，由于是先定义了一个字符串名字来代替列名，在语句里面直接用+连接列名和数据类型名，没有加空格，导致创建表失败，

private static final String KEY\_ID = "id";

String CREATE\_ATTRIBUTE\_TABLE = "CREATE TABLE " + TABLE\_POINTATTR + "("  
 + KEY\_ID + " INTEGER PRIMARY KEY,"  
 + KEY\_ATTR\_LATITUDE + " TEXT,"  
 + KEY\_ATTR\_LONGITUDE + " TEXT,"  
 + KEY\_TIME + " TEXT,"

注意一定要加空格

# 2016年1月17日20:49:20

1. 设置spinner的弹框位置，默认会把文字遮挡不好看，需要把弹框放到文字的正下方，通过两个属性来设置水平和竖直上的偏移。

**android:dropDownHorizontalOffset="50dp"  
android:dropDownVerticalOffset="100dp"**

# 2016年1月17日23:22:34

1. 现在已经能在listview里面显示表中项目了，但是在数据量比较少的情况还好，太多就卡了，现在才三条数据就有点卡了，应为我加载数据都是写到主线程里面的。

# 2016年2月24日16:25:25

1. 给activity的actionbar设置返回按钮。

首先要找到actionbar，

要用android.support.v7.app里面的actionbar，而不要用其他地方的，否则会失败。

设置显示主按钮home，然后在menu的点击方法里面找到这个菜单id为home，然后finish即可。

ActionBar actionBar = getSupportActionBar();  
**if** (actionBar != **null**) {  
 *// Show the Up button in the action bar.* actionBar.setDisplayHomeAsUpEnabled(**true**);  
}

@Override  
 **public boolean** onOptionsItemSelected(MenuItem item) {  
*// return super.onOptionsItemSelected(item);* **int** id = item.getItemId();  
 **if** (id == android.R.id.***home***) {  
 **this**.finish();  
 **return true**;  
 }  
 **return super**.onOptionsItemSelected(item);  
 }

# 2016年2月25日09:48:58

之前做的百度轨迹的android程序，在定位上会不受控制的多次定位。从主界面进入地图界面后，马上退回，就会增加一个采集点，而明明没有点击采集按钮，这样的效果是不行的。主界面和地图界面都用到了application类里面定义的定位类的对象，因此是全局的，只是实例化了一个对象，百度建议就是这样做的，因为定位功能是一个service。单问题就出现在这里了，两个接地定位冲突了，主界面和地图界面用的定位监听器对象是不一样的，都在各自的界面进入退出方法onstart，onstop里面注册和注销，register和unregister，理论上从地图界面退出后就已经注销了地图界面的定位监听器，重新进入主界面的又会注册新的监听器，程序应给没有错才对。但是地图界面退出后，定位类仍然调用了地图界面的监听器，用的地图界面的监听器，定位成功就会增加采集点，主界面定位成功是不会增加采集点的，这是不合理的。

虽然没有搞清楚出现问题的根本原因是什么，但是还是修改正确了。之前在地图界面的 onCreate方法和采集按钮点击方法里面都初始化了定位，绑定监听器，只不过按钮里面会有start开始定位方法的调用。因此每次进入地图界面，都会绑定新的监听器到定位类，onStop方法里面虽然用了unregister方法，但不一定真正的解除了绑定。因此我决定不在地图界面的 onCreate方法里面初始化定位了，只是在按钮点击里面初始化，如果只是这样，在进入地图界面后如果没有点击采集按钮，就退出，在地图界面的onStop方法里面调用注销方法是会出错的，因此在地图界面的onStop方法里面也不调用注销方法了，在定位成功后直接促销，也就是在监听器mLocationListener

里面定位成功后直接注销。这下就没有前面的问题了。

# 2016年2月25日20:18:36

1. Android studio真正的下载地址

<http://tools.android.com/recent>

<http://tools.android.com/download/studio/canary>

<https://dl.google.com/dl/android/studio/ide-zips/2.0.0.14/android-studio-ide-143.2609919-windows.zip>

直接在百度或谷歌搜索的android studio下载地址是developer.android.com里面的，这里的版本太低，而且是exe格式的，正确的下载地址是前面写的，是zip的压缩包，解压后打开就可以用。

最近应该是android studio升级了，gradle版本不对，刚新建的工程就报错，说插件版本太低，改成别的版本又有别的错误，导入别人的工程就没有一个是真确的，全是gradle版本错误。

# 2016年2月26日12:01:24

setContentView方法所指定的View，只有在onCreate方法返回后才会显示在界面上。因此，如果调用了两次setContentView方法，只有最后一次才是有效的。

# 2016年2月26日23:08:40

搞了一天终于把启动界面搞出来了，太不容易了，这真是如果做过，那么只要10分钟，如果没有做过，一天甚至几天都有可能。一个启动界面，这么简单的部分没想到太特殊了。

最初的思想是，启动界面为一个单独的activity，这个启动界面运行完就启动主界面，好处是启动界面和主界面分开，可以单独设置主题，那么启动界面全屏，没有title和actionbar都和主界面无关，因为主界面需要title和actionbar，这是好处，但是缺点也很明显。启动界面为最先运行的界面，那么每次退出程序后再次进入都要进入一次启动界面，即使程序刚退出还在缓存中保留，启动界面本来就有延迟，那么每次进入程序都先打开启动界面，效果很不好，网上的常用程序，比如新浪微博、酷狗等，都有启动界面，但是人家的是如果程序彻底退出后进入就就先进入启动界面，再进入主界面，如果程序刚运行了，没有彻底退出，再次进入是直接进入主界面。效果很好，因此我也要做到它们这样。

要这样，就一定不能用两个activity，原来的启动界面不能要了。只能保留主界面，关键是setContentView可以调用多次，因此除了主界面的xml布局外，再写一个启动界面的xml布局，先调用setcontentview设置启动解密的布局，然后延迟在设置主界面的布局。主要思路是这样的，但是如果是直接这样设置是看不到效果的，setcontview只有最后一次才有效，网上有解决方法，思路是差不多的，但有区别，我还没有看懂。

透明主题要卸载后重新安装才有效，替换的话程序就不能运行，点击图标没有反应，这也不知道为什么，这个问题困扰了我很久，直到后来我偶然卸载了一次程序，再设置透明主题重新安装，就可以了。

# 2016年3月3日11:05:03

修改dialogplus，成功在viewholder中添加了radiobutton，我只是修改了dialogplus和dialogplusbuilder，而viewholder，gridholder和listholder没有修改，只是在viewholder的布局xml文件里面添加了radiobutton。参考代码了给view添加clicklistener的方法，添加checkedchangedlistener监听。首先添加接口OnCheckedChangedListener，

**public interface** OnCheckedChangedListener {  
 **void** onCheckedChanged(RadioGroup group, **int** checkedID);  
  
}

在dialogplus的createview方法里面找到不同holder的view，如果这个view是viewgroup，那么用getchild和getChildAt找到子view，然后用递归的方法继续判断，如果view是RadioGroup，那么给它添加setOnCheckedChangeListener，然后用前面定义的接口对象来实现。

**private void** assignCheckedChangedListenerRecursively(View parent) {  
 **if** (parent == **null**) {  
 **return**;  
 }  
 **if** (parent **instanceof** ViewGroup) {  
 ViewGroup viewGroup = (ViewGroup) parent;  
 **int** childCount = viewGroup.getChildCount();  
 **for** (**int** i = childCount - 1; i >= 0; i--) {  
 View child = viewGroup.getChildAt(i);  
 assignCheckedChangedListenerRecursively(child);  
 }  
  
 }  
 setCheckedChange(parent);  
}  
  
**private void** setCheckedChange(View view)  
{  
 **if** (view **instanceof** RadioGroup) {  
 ((RadioGroup) view).setOnCheckedChangeListener(**new** RadioGroup.OnCheckedChangeListener() {  
 @Override  
 **public void** onCheckedChanged(RadioGroup group, **int** checkedId) {  
 **if**(**onCheckedChangedListener**==**null**)  
 {  
 **return**;  
 }  
 **onCheckedChangedListener**.onCheckedChanged(group,checkedId);  
 }  
 });  
  
 }  
}

接下来就是raidobutton的样式了，默认的是一个圆圈里面有个点，我需要做到像百度地图一样，是两个圆角矩形，点击不同的按钮，边框颜色不一样。

# 2016年3月21日11:31:18

1. 沉浸式状态栏

**if**(Build.VERSION.***SDK\_INT***>=Build.VERSION\_CODES.***KITKAT***)  
{  
 getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_TRANSLUCENT\_STATUS***);  
 getWindow().addFlags(WindowManager.LayoutParams.***FLAG\_TRANSLUCENT\_NAVIGATION***);  
}

第一设置让状态栏透明，也就是通知栏透明，通知栏的颜色和activity的背景颜色一致。第二个让导航栏透明，华为的手机底部有导航栏，如果这样设置，导航栏的背景颜色和activity一致。

除了这样设置还要设置没有actionbar，在主题里面设置

<**item name="windowActionBar"**>false</**item**>  
<**item name="windowNoTitle"**>true</**item**>

否则，会多出actionbar，影像效果。

这些设置以后，activity里面的控件就是从手机屏幕的最左上角开始布局了，从原来状态栏左上角开始布局，而不是从状态栏下面开始布局。也就是说只如果放一个textview，没有设置边距，那么它会显示在状态栏文字的下方。

# 2016年3月31日18:00:12

StateListDrawable是表示不同状态下的drawable，已经在很多地方用过了，但都不知道是StateListDrawable，比如按钮的状态，点击和不点击背景颜色的不同，形状图片的不同。这些都是在xml里面设置的，只要给某个view的背景或颜色设置这个drawable资源就可以。StateListDrawable资源也是一个xml文档，用selector和item来表示。

*<?***xml version="1.0" encoding="utf-8"***?>*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
 <**item android:drawable="@drawable/ic\_sync\_black\_24dp1" android:state\_pressed="true"**/>  
 <**item android:drawable="@drawable/ic\_sync\_black\_24dp2" android:state\_pressed="false"**/>  
  
</**selector**>

*<?***xml version="1.0" encoding="UTF-8"***?>*<**selector xmlns:android="http://schemas.android.com/apk/res/android"**>  
 *<!-- 指定获得焦点时的颜色 -->* <**item android:state\_focused="true"  
 android:color="#f44"**/>  
 *<!-- 指定失去焦点时的颜色 -->* <**item android:state\_focused="false"  
 android:color="#ccf"**/>  
</**selector**>

StateListDrawable不仅可以用xml来实现，还可用java语句来实现，之前写的轨迹调查的主界面，类似gridview的九宫格界面就是用StateListDrawable来实现的，其中个图标点击和不点击的颜色是不一样的，就是加载不同的颜色的图片。

**final** StateListDrawable firstDrawable = **new** StateListDrawable();  
firstDrawable.addState(**new int**[]{android.R.attr.***state\_pressed***}, getResources().getDrawable(firstDrawablePressed));  
firstDrawable.addState(**new int**[]{}, getResources().getDrawable(firstDrawableNormal));

其中**final int** firstDrawableNormal = getResources().getIdentifier(String.*format*(**"ic\_department\_%02d\_normal"**, firstIndex + 1), **"drawable"**, getActivity().getApplicationContext().getPackageName());

View.*inflate*(**view**.getContext(), R.layout.***layout\_line\_vertical***, linearLayout)

# 2016年4月2日16:50:06

每隔一段时间就会提示gradle版本太旧，然后怎么改都不不行，也不知道最新版本是多少，正确办法不是去下载最新的gradle版本，android studio会自动下载，自己下载的重复了，也没用，是要配置真确的版本

<https://jcenter.bintray.com/com/android/tools/build/gradle/>

到这个网站看最新的版本编号，比如此时最新的是2.1.0-alpha5

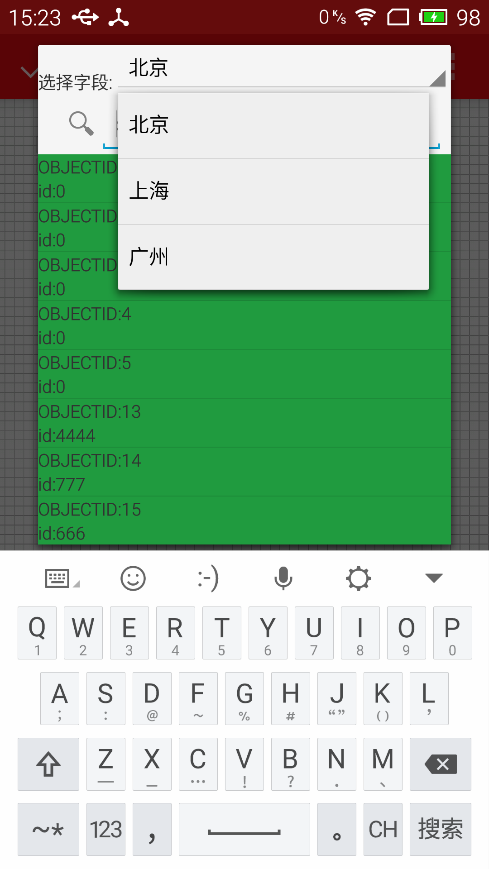
改为这个classpath **'com.android.tools.build:gradle:2.1.0-alpha5'**

就好了

# 2016年4月3日15:23:39

设置spinner为一种比较好看的样式

**spinner** = (Spinner) **view**.findViewById(R.id.***spinner2***);  
  
ArrayList<String> datalist= **new** ArrayList<>();  
datalist.add(**"北京"**);  
datalist.add(**"上海"**);  
datalist.add(**"广州"**);  
  
ArrayAdapter<String> arrayAdapter = **new** ArrayAdapter<String>(getActivity(), android.R.layout.***simple\_spinner\_item***,datalist);  
arrayAdapter.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
**spinner**.setAdapter(arrayAdapter);



# 2016年4月3日18:30:12

Arcgis for android 查找属性表找到要素

最近在写arcgis for android 程序，因为之前的程序虽然完成了领导的要求，但是还不完善，没有查找功能。最近古树认种的事情出了问题，通过照片，陈杰有很多都不认识，反馈回来的表有一大半是空白的，他建议重新去现场核对，这件事我都还没有跟领导汇报，领导知道了肯定要疯的。我想如果去现场核对，那么怎么找到每棵树呢，因此必须添加一个搜索功能。由于能得到一个图层中所有要素的字段属性，那么得到属性表应该是没有问题的，从这个表中查找就能解决问题，现在问题的关键是怎么查。

# 2016年4月4日20:38:39

Android仿dialogplus制作侧面展开对话框



设计基于github上的开源项目dialogplus，地址<https://github.com/orhanobut/dialogplus>

看到这个开源项目的源代码，不得不佩服原作者的编程能力。程序设计结构非常巧妙，易于扩展，效果完美。

首先，拿到这个库，不需要完全弄明白每一步的原理，就可以很容易的将其添加到自己的项目里面。只需要自定义一个布局的xml文件，给它赋值到你对应的holder里面，设置几个监听事件，然后就可以用DialogPlusBuilder来创造DialogPlus了。程序巧妙就巧妙在用了builder设计模式，将复杂的实现过程用一个设计器来实现，不需要了解具体过程，只需要给它赋值就行。

第二点就是关于界面了，设计的非常好。动画界面非常华丽，可从上面、下面和中间展开和消失，除此之外，最让我感到惊奇的是DialogPlus没有继承Fragment或者View，它就是一个单独的类。以往看到的比较华丽的界面，大多都是自定义View，让后将这个viwe当做普通的控件一样使用。如果写一个 普通的自定义对话框，一般的做法是重写Dialogframent，然后在onCreateView方法里面绑定布局的xml文件。如果只是自定义一个Fragment的话，也是要重写Fragment。重写Fragment有个很大的弊端，如果Fragment少的话还比较好办，如果一旦多起来，不同的fragment之前传值就非常困难。在我了解到DialogPlus之前只知道fragment，看到别人的非常复杂的界面就想到是不是每个小块都是用fragment做的呢，如果是的话，那fragment里面嵌套子fragment怎么传递值呢。想了很久也不行。Fragment传值要用到它所依附的Activity，activity找的fragment里面的要素要用fragmentmanager的getview方法，fragment传递值到activity要用到接口和onAttch方法。因此fragment多了就不好弄了。

在了解了DialogPlus后，发现原来精美的界面根本不用fragment。DialogPlus里面用到的关键是LayoutInflater，用这个类可以将xml的布局文件解析为View，然后找主activity的ViewGroup，将解析后的子view添加到activity的viewgroup就可以了，然后设置一些布局参数和动画效果就非常绚丽了。发现用LayoutInflater可以非常方便的添加新的布局到想要的位置。

LayoutInflater layoutInflater = LayoutInflater.from(builder.getContext());

decorView = (ViewGroup) activity.getWindow().getDecorView().findViewById(android.R.id.content);  
  
rootView = (ViewGroup) layoutInflater.inflate(R.layout.base\_container,decorView,false);  
  
rootView.setLayoutParams(builder.getOutmostLayoutParams());

接下来步入正题，讲讲我基于DialogPlus做的改进。

首先，我做了个从左边展开的动画，使用translate动画就能实现

<**translate  
 android:duration="300"  
 android:fromXDelta="-100%"  
 android:fromYDelta="0%"  
 android:toXDelta="0%"  
 android:toYDelta="0%"** />

在构造DialogPlus的时候设置.setGravity(Gravity.LEFT)即可。原来的代码上是没有从左边展开的。

2.原来的代码里面，默认是占满整个宽度的，设置Gravity.TOP或Gravity.BOTTOM的时候，都是从边界处开始展开的，不能从想要的位置展开，不能满足我的要求，因此我在DialogPlusBuilder里面添加了个设置边距的方法setContentMargin

**public** DialogPlusBuilder setContentMargin(**int** left, **int** top, **int** right, **int** bottom) {  
 **this**.**margin**[0] = left;  
 **this**.**margin**[1] = top;  
 **this**.**margin**[2] = right;  
 **this**.**margin**[3] = bottom;  
  
 **return this**;  
}

在给FrameLayout.LayoutParams params赋值的的时候，将这四个边距赋给params

**public** FrameLayout.LayoutParams getContentParams() {**this**.**params**.setMargins(**margin**[0],**margin**[1],**margin**[2],**margin**[3]);  
 **return params**;  
 }

需要注意的是这个布局参数params所要用到的地方，原来的代码里面是没有**this**.**params**.setMargins(**margin**[0],**margin**[1],**margin**[2],**margin**[3]);  
这句话的，也就是说params的边距默认是0。

在DialogPluse里面调用

**contentContainer** = (ViewGroup) **rootView**.findViewById(R.id.***dialogplus\_content\_container***);  
**contentContainer**.setLayoutParams(builder.getContentParams());

那么contentContainer默认就从rootView的左上角开始布局，

然后原代码又在initContentView(  
 layoutInflater,  
 builder.getHeaderView(),  
 builder.getFooterView(),  
 builder.getAdapter(),  
 builder.getContentPadding(),  
 builder.getContentMargin()  
);

方法里面调用了builder.getContentMargin()方法，这个边距不是赋值给contentContainer的，儿是赋值给contentContainer里面的contentview，即我们自定义的holder里面的内，那么holder里面的view距离contentContainer的边距设置了，看起来的效果确实成功设置了边距，因为我们最后看到的就是holder里面的view，contentContainer只是个容器，看不见的(但摸得着)。按照原来的代码，设置了margin边距后，效果看到了，但是发现点击界面左边和上面的空白不能取消对话框，只有点击下面和右面的空白才行。仔细研究代码才发现问题所在，原代码设计的也是非常巧妙的，它给rootview设置了个Touch事件，一旦触摸，就会调用dismiss()方法关闭对话框，但不是rootview上的任何地方点击都会关闭，里面有控件的，当然执行控件的事件，只有点击空白处才会关闭。这个rootview是占据了整个activity，然后contentContainer包含在里面，contentContainer再包含contentView。原程序的contentContainer在设置布局参数params的时候，margin属性没有，默认为0，那么就从rootview 的左上角布局，contentview又设置了margin，因此content距contentContainer有边距，contentContainer距rootview的左边和上边距是0，导致最终问题是点击左边和上边的空白不能关闭对话框，正确的做法是给contentContainer设置边距，contentview不设置边距，默认用Match\_Parent充满contentContainer，这样点击任何空白处就能关闭对话框了。

注意在构造对话框的时候不要调用setOutMostMargin(100,200,200,200)来设置边距，这个是给rootview设置边距的，而且rootview还有个设置背景的方法，用来设置蒙版效果，这个边距设置后，蒙版就不能充满整个屏幕，看起来效果不好。如果要取消蒙版效果需要在DialogPluse里面注销掉*outmostView.setBackgroundResource(builder.getOverlayBackgroundResource()*

语句。

然后在构造对话框的时候调用

.setContentMargin(0,250,0,0)  
.setContentHeight(900)  
.setContentWidth(500)

这三句就可以设置对话框的位置和大小了，注意单位是px，不是dp。

然后是关于设置监听事件的，原代码里面设置了很多关于监听的方法，然后赋值给dialogplusbuilder，在传递给dialogplus，在里面再用递归的方法给每个view设置事件。不同类型的事件还需要单独传递，比如原来里面只有按钮或图片，只有点击事件，值需要传递一个点击事件的接口变量即可，然后dialogplus里面指定给每个view分配点击事件。但后来我添加了radiobutton，事件又不同了，我需要重新修改dialogplusbuilder和dialogplus的方法，添加radiogroup选择改变事件，非常不方便，如果后来又有listveiw，那岂不是疯了。我找到了个很简单的方法，根本不需要往dialogplus里面传递view的事件，如果有需要只需要传递对话框关闭或取消等的事件。要给dialogplus里面的view赋事件只需在构造完对话框后，找到想要的view，然后给它赋事件就可以了。由于传递给dialogplus的内容来自holder，因此从holder里面用getInflateView找到相应的view，然后设置事件即可。

ListView listView = (ListView) **holder**.getInflateView().findViewById(R.id.***testList***);

listView.setOnItemClickListener(**new** AdapterView.OnItemClickListener() {  
 @Override  
 **public void** onItemClick(AdapterView<?> parent, View view, **int** position, **long** id) {  
 Toast.*makeText*(MainActivity.**this**,names[position],Toast.***LENGTH\_SHORT***).show();  
 }  
});

# 2016年4月4日22:43:12

找到activity所在的窗体

**decorView** = (ViewGroup) activity.getWindow().getDecorView().findViewById(android.R.id.***content***);  
**rootView** = (ViewGroup) layoutInflater.inflate(R.layout.***base\_container***,**decorView**,**false**);

layoutInflater只是解析xml的资源文件为View对象，并不是添加到父容器里面，需要用父容器的addview方法来添加

decorview.addView(rootview)

# 2016年4月9日23:22:07

最近想模仿微博头条客户端的导航界面，就是viewpager加自定义的viewpagerindicator来实现，它的选择项的标题有个动画效果，经过几天的苦苦研究，总算弄的差不多了。但是今天就剩最后一个问题了，困扰我一个下午加晚上到现在，最后用了一个非常简单的方法解决了。就是之前每个选项块离的太近，看着很臃肿，想把每块的宽度设置大一些，无奈怎么都弄不好。每一个选项块是一个自定义的ViewGroup，继承自RelativeLayout,里面要重写onMeasure方法，如果是直接继承ViweGroup的话，还要重写onLayout，就太复杂了。OnMeasure方法就是用来设置子view的大小的。当父容器要加载该view的时候，就要知道这个view或viewgroup的大小。这里在布局参数设置的时候，在设置宽度和高度的时候，有设置Wrap\_Content和Match\_Content,或者具体的值，这个参数的设置可以在xml里面设置，也可以是java代码里面设置。问题就在这里，如果是具体是值，父容器就不用计算子view的大小了。OnMeasure会调用，但也没有多大意义了。但是如果是设置的warp\_content或者match\_parent,那么就要通脱onMeasure来设置子view的大小了，如果没有特殊要求，直接重写它，调用super方法即可，如果还要设置大小，如果是继承自view，调用setMeasuredDimension(int measuredWidth, int measuredHeight)方法，如果是继承自viewGroup方法，调用super.onMeasure(widthMeasureSpec, heightMeasureSpec)方法。这两个方法的参数就是要设置的新的值，根据出入的值计算得到新的值传入，即可。我试过super.onMeasure确实能改变大小，但最终都不是我想要的结果，一开始如果没有写默认的super.onMeasure(widthMeasureSpec, heightMeasureSpec)方法，就直接写改变过后的值，就是错误的，onMeasure方法不知道为什么会调用两次，widthMeasureSpec解析出来的大小有时候是0，有时候又不是，在调用默认的方法后，再次调用这个方法，只是参数变了，貌似就可以了，应为我看到网上的参考代码就是这么写的，问题又出现了，网上的是直接继承自TextView，而我是继承自RelativeLayout的，网上的代码再次调用super.onMeasure的方法是传递一个更小的宽度进去，它的程序没有问题。而我是需求是要加宽view，需要设置更大的宽度，因此我在super.onMeasure方法里面传递了一个更大的宽度值，一开始看起来没有问题，确实宽度增加了，但是仔细一看，滑动到最右边，发现现实不全了，最后一项又一大半显示不了，也滑动不过去，我以为是线性布局的父容器装不下了呢，将数组个数减小，发现问题依旧，这就是困扰我最久的问题，实在高不明白，为什么就显示不全了呢，也许是第二次调用super.onMeasure只能设置更小的值，大的就显示不全了，可能是第一次调用super.onMeasure的时候，系统就已经定死了最大宽度，不能再设置更大的了。那么我的问题还是没有解决啊，我只是想增大每一选项块的宽度啊。

后来看到都23点了，就打算不在实验室了，在回来的路上也在思考这个问题，突然灵机一动，发现，如果我在每个选项块里面的textview的布局参数里面设置margin较大会怎样，因为每个选项块默认都是wrap\_content或match\_parent,都只是把内容包裹完就了事，我想如果我把textview的大小设置更大，那么么个选项块是不是宽度就要增加，也就是textview占据的范围更大而已，进一步，那我可以设置更大的边距啊，也会增加textview占据的空间。回来后我试了下，只是在xml布局里面讲margin\_left和margin\_right设置的更大，效果一下就出来了，每个选项块的宽度就增加了，而文字的大小也不变。

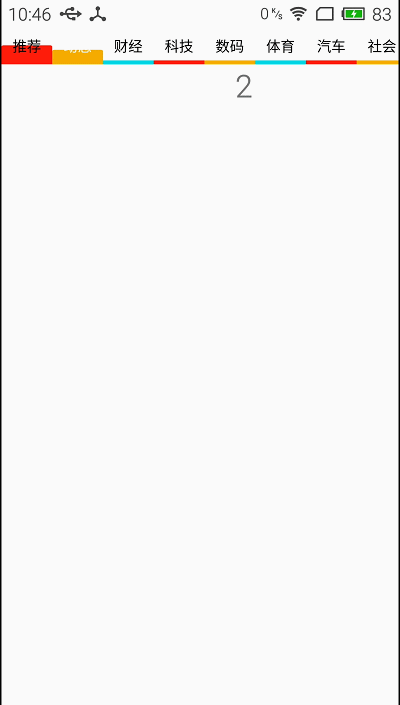
# 2016年4月10日10:42:36

记录下自己写的模仿微博头条客户端导航界面的的过程与经验，然后在写一篇博客发到网上。

Android使用viewpager和自定义viewpagerindicator仿微博头条客户端

1. 微博头条客户端界面设计非常精美，第一次看到就被吸引住了，也许是在此之前还从来没有看到别的那个APP用过这种设计风格。就像qq客户端一样，当初应该是首次使用侧滑展开隐藏内容，然后主界面缩小的效果。使用别人没有的好的设计当然能最先吸引用户。qq侧滑的效果的成功是不用质疑的，从后来网上很多模仿他的侧滑效果就可以看出，网上有很多这种效果的源代码。因此能有个别人没有的创意是多么重要。

而本文要讲的就是模仿微博头条客户端的导航界面。先看看效果。



它的界面非常好看，顶部是导航块，下面是用viewpager做的fragment。首先最惊艳的是它的切换动画效果，每一个选项块的背景都是一个高度可变的圆角矩形，在viewpager切换的时候，当前选项块的背景矩形高度下降，下一个选项块的背景矩形高度升高，动画效果很好。第二就是当前选中块的背景是充满的，没有选中的只有下面的一条横线。第三，每个选项块的颜色不同，使界面更绚丽。第四，顶部导航栏的最外层容器使用HorizontalScrollView，可以水平滚动，当选项块太多时，可以滑动它展开隐藏内容。

1. 接下来就要讲讲它是设计过程了

有过android编程经验的人都能猜到，它用了viewpager来实现。只有它当然还是不行的，以前使用过viewpager，网上讲解的相关资料都是，里面使用viewpagertabstrip来做导航条的，使用起来确实简单方便，但是达不到我要的效果，它的默认效果是一个页面只显示三个选项块，而且颜色都不能随意定，还有滑动的时候选项块也要跟着滚动到中间，留下一边空白的，总之效果不好。于是就在网上找新的解决办法，发现了viewpagerindicator，用它来代替pagertabstrip作为导航条。

Viewpagerindicator只是一个自定义的viewpgroup，由于要实现滑动展开隐藏的内容，使它集成HorizontalScrollView。

# 2016年5月22日09:08:43

dispatchTouchEvent的用法

这个方法一般用在重写viewgroup类，重写它的意思是重新分派屏幕触摸事件。

屏幕触摸事件主要有：

MotionEvent.***ACTION\_DOWN***:

MotionEvent.***ACTION\_MOVE***:

MotionEvent.***ACTION\_CANCEL***:

MotionEvent.***ACTION\_UP***:

Viewgroup下面可以有viewgroup和view，子viewgroup下面又可以有这些，层层递进。触摸事件是一层一层往下分派的，这个方法返回值是boolean类型的，如果为true，则不往下分派，下层的view或viewgroup接收不到触摸事件。如果为false，则继续往下分派。

# 2016年5月22日20:16:28

1. 对自定义viewgroup的理解

开发自定义viewgroup的时候，往往要重写onMeasure和onLayout两个方法。

1. **protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec)

这个方法用来测量子view的宽度和高度，也可以设置宽度和高度，

调用view的measure方法设置宽度和高度

**protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  
  
**int** width = MeasureSpec.*getSize*(widthMeasureSpec);  
 **int** height = MeasureSpec.*getSize*(heightMeasureSpec);  
 setMeasuredDimension(width, height);  
  
 **int** childCount = getChildCount();  
   
 **for** (**int** i = 0; i < childCount; i++) {  
 View child = getChildAt(i);  
 child.measure(getWidth(), MultiScreenActivity.*scrrenHeight*);  
 }  
}

1. **protected void** onLayout(**boolean** changed, **int** l, **int** t, **int** r, **int** b) {

这个方法用来布局子 view或viewgroup，即设置它们的位置。

@Override  
**protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  
 **super**.onMeasure(widthMeasureSpec, heightMeasureSpec);  
 **middleMenu**.measure(widthMeasureSpec, heightMeasureSpec);  
 **middleMask**.measure(widthMeasureSpec, heightMeasureSpec);  
 **int** realWidth = MeasureSpec.*getSize*(widthMeasureSpec);  
 **int** tempWidthMeasure = MeasureSpec.*makeMeasureSpec*(  
 (**int**) (realWidth \* 0.8f), MeasureSpec.***EXACTLY***);  
 **leftMenu**.measure(tempWidthMeasure, heightMeasureSpec);  
 **rightMenu**.measure(tempWidthMeasure, heightMeasureSpec);  
}  
  
@Override  
**protected void** onLayout(**boolean** changed, **int** l, **int** t, **int** r, **int** b) {  
 **super**.onLayout(changed, l, t, r, b);  
 **middleMenu**.layout(l, t, r, b);  
 **middleMask**.layout(l, t, r, b);  
 **leftMenu**.layout(l - **leftMenu**.getMeasuredWidth(), t, r, b);  
 **rightMenu**.layout(  
 l + **middleMenu**.getMeasuredWidth(),  
 t,  
 l + **middleMenu**.getMeasuredWidth()  
 + **rightMenu**.getMeasuredWidth(), b);  
}

调用View的scrollTo()和scrollBy()是用于滑动View中的内容，而不是把某个View的位置进行改变。如果想改变莫个View在屏幕中的位置，可以使用如下的方法。

调用public void offsetLeftAndRight(int offset)用于左右移动方法或public void [offsetTopAndBottom](http://developer.android.com/reference/android/view/View.html#offsetTopAndBottom(int))(int offset)用于上下移动。

                 如：button.offsetLeftAndRignt(300)表示将button控件向左移动300个像素。

# 2016年5月23日09:36:56

LayoutInflater layoutInflater = LayoutInflater.*from*(getContext());  
View view = layoutInflater.inflate(R.layout.***title\_bar\_tool***,**this**,**false**);

这段代码用于从xml布局文件中解析出view对象，方便在java代码中用addview添加视图。

注意第二个方法layoutInflater.inflate有三个参数，最后一个是false，它也有两个参数的重载方法，没有最后一个boolean的参数，这个参数boolean attachToRoot表示是否添加到根节点，不知道什么意思，但如果没有这个参数，那么会发生内存溢出异常，java.lang.StackOverflowError: stack size 8MB

加了这个参数，并且为false就不会发生这个异常了。

# 2016年6月3日00:59:59

关于float类型的运算

刚刚在做textview的透明度的设置，调用setAlpha方法传入一个float的参数进入，范围是0到1.0，数值越小越透明。

我是用两个int整数相除得到小数的，可是发现其中一个数从小变到大，结果这个float的数值只有两种值：0.0和1.0，根本就没有连续变化。

**float** alpha = (355-currentPos)/255;

currentPos从0逐渐变大，一开始alpha为1.0，当currentPos从100变化到101的时候，alpha就突然变成了0.0，真是奇怪。

修改为：**float** alpha = (355-currentPos)/255f;

添加了一个f就好了。

# 2016年6月5日22:40:59

Java的String类结

(1). String是一个类，首字母大写，C#中大小写都可以

(2). String对象的初始化有两种方式，一是直接赋值，二是用new来实例化。

两者有区别。

String str1=”hello”

Sting str2 = new String(“hello”)

第二种方式在构造方法里面传递一个字符串变量，这种方法在C#里面是不可以的，C#中的String构造方法可以传入char的数组和其他类型，但不能直接传递一个字符串。

两者的区别在于，直接赋值只在堆内存中开辟一个空间，而用new的方法会开辟两个空间，一个的废弃了的内存空间，因此用new的方法会浪费空间。只要有一个双引号括起来的字符串，那它就是String类的一个匿名对象，就一定会开辟一个堆内存空间，因此用new的方法是先开辟一个空间来存储hello字符串，然后又用new方法开辟了一个空间也存储hello，所以会开辟两个内存空间。

使用new的方法一定会开辟新的空间，而使用直接赋值的方法不一定会开辟新的空间。定义一个String对象的时候，会从堆中查找是否有相同内容的字符串，如果堆中已经有相同内容的字符串，新定义的String对象是不会开辟新的空间，直接指向该相同内容的空间。

String str1= “hello”

String str2=“hello”

这两句话实际上只开辟了一个内存空间，str1和str2都指向堆中同一个空间，该空间内容为“hello”

(3). String对象的内容不可变性

这个特点和C#相同，一旦一个String变量赋值了，该堆内存的空间就不能修改了，如果重新赋值就会开辟新的空间，指向新的地址。由于这个特点，因此不能够给string变量大量重复赋值，否则会浪费内存，解决办法是用stringbuffer，在C#中是用stringbuilder。

(4). String的比较

有两种方式：==和equal

在值类型中==比较的是内容大小，而在String中比较的是内存地址，equal比较的是内容。

String str1=“hello”

String str2 =“world”

String str3=“hello”

那么用==，str1和str2不相等，str1和str3相等

String str1= “hello”

String str2= new String（“hello”）

Str1和str2不相等

# 2016年6月9日13:06:11

刚犯了一个很低级的错误，还想了很久，调试了很久的程序才发现原因。

有个两个arraylist：lists和oldlists，目的是点击按钮后给oldlists赋lists的值，

一开始的做法是自己用=：在点击事件里写：

Oldlists= lists；

发现当点击了按钮后，后续操作改变里lists的值，但没有再次点击按钮，读取oldlists的值，发现内容变了，变成改变后的lists的内容了，找了很久，怀疑是不是哪里又给oldlists赋值了，可是没有找到。

后来才突然想到list是引用传递，lists和oldlists都指向同一个堆内存空间，其中一个的内容变了，另一个的内容跟着变。

为了不让oldlists的内容自动随lists的内容变化，一开始oldlists就不应该指向lists的地址，也就是不能用=，而要用new，在点击事件里面改为：

oldlists = new ArrayList<>(lists);

这样就好了，每次点击就new一个oldlists，lists的变化不会影响oldlist的变化，只有点击按钮才赋值给oldlists。

# 2016年6月21日23:18:25

1. HorizontalScrollView中添加元素

最近给viewpager的indicator添加内容，一开始条目比较多，总宽度超过了屏幕宽度，因此每个都是wrapcontent，布局比较紧密，可是后来发现当数目比较少的时候，比如只有两个，那么每个条目都被拉宽了，直到填充满整个屏幕，真的很奇怪，我的布局参数设置的都是warpcontent啊，不仅是父容器，每个子容器都是wrapcontent啊，后来发现根容器类型为HorizontalScrollView，它比较特殊，如果在onMeasure中调用了setFillViewport(true),那么所有内容将被拉伸来填满HorizontalScrollView，比如HorizontalScrollView中一般添加一个LinearLayout，且只能添加一个子元素，更多的就要在LinerLayout中添加了，如果调用了setFillViewport(true)，那么LinearLayout就要被拉伸，不管LinearLayout对象调用setLayoutParams时参数是matchparent还是wrapcontent，都要被拉伸。

如果不设置setFillViewport(true)，那么子元素如果设置的即使是matchparent，也会变成wrapcontent，根本不会充满HorizontalScrollView，好像也应该是这样，因为HorizontalScrollView的宽度本来就是不确定的，怎么会充满呢。

不仅设置matchparent和wrapcontent没用，就连设置一个固定的宽度都没用，全都会变成wrapcontent。

总结：HorizontalScrollView的元素宽度要么是wrpcontent，要么是fillparent，不是用params来设置的，而是用setFillViewport方法来设置，如果没有调用或者参数为false，那么就是wrapcontent，这是默认效果，如果参数是true，那么所有元素就要被拉伸来充满HorizontalScrollView，当然是内容元素没有充满父容器的宽度的时候才拉伸。

因此要想达到我目的：HorizontalScrollView中部分内容充满，另一部分不充满，是实现不了的。我需要indicator的条目较少的是时候就排在左边，不拉伸，但是需要一条下划线充满整个HorizontalScrollView的底部，这样是实现不了的，要么都拉伸，要么都不拉伸。我认为根本原因就是HorizontalScrollView的宽度是不定的，从它的名字就可以看出，水平滚动，当内容超过屏幕宽度的时候还可以滚动来展开隐藏内容，隐藏内内部可以加任意多的元素，宽度不定。

要解决在indicator的底部加一条充满屏幕的下划线，就不能再indicator内部添加了，只能在外部和indicator并排添加到一个relativelayout中了。

# 2016年6月24日11:25:01

1. android源码查看地址

<http://grepcode.com/project/repository.grepcode.com/java/ext/com.google.android/android/>

2016年6月27日21:33:41

1. java线程同步

(1). 使用徒步代码块synchronized，参数是当前对象

class Mythread implements Runnable

{

private int tickets=5;

public void run()

{

**synchronized(this){**

for(int i=0;i<100;i++)

{

if(tickets>=0)

{

System.out.println("剩余票数："+tickets--);

}

}

**}**

}

}

public class ThreadDemo04{

public static void main(String[] args)

{

Mythread ru = new Mythread();

new Thread(ru).start();

new Thread(ru).start();

}

}

**输出为：**

剩余票数：5

剩余票数：4

剩余票数：3

剩余票数：2

剩余票数：1

剩余票数：0

如果不加synchronized，那么每个线程都可以同时访问tickets变量，导致输出结果混乱，加上后，那么synchronized所包裹的代码块任何时候只能有一个线程访问，其余要访问的线程只能等待。

(2) 同步方法

即在方法前加上synchronized关键字

class Mythread implements Runnable

{

private int tickets=5;

public void run()

{

this.sale();

}

public synchronized void sale()

{

for(int i=0;i<100;i++)

{

if(tickets>=0)

{

System.out.println("剩余票数："+tickets--);

}

}

}

}

public class ThreadDemo05{

public static void main(String[] args)

{

Mythread ru = new Mythread();

new Thread(ru).start();

new Thread(ru).start();

}

}

结果和前面一样

**加上synchronized同步代码块后执行效率降低。**

程序中需要资源共享的时候需要同步。

程序中过多的同步会产生死锁。

# 2016年8月18日16:21:04

1. 使用butterknife简化程序

一般的程序需要大量使用findViewById()和setonclicktListener()，使用butterknife后就可以不使用这些方法就能直接使用控件了。普通程序需要使用findViewById从xml中找到控件，否则控件为null，就不能使用。

butterknife是一个注解包，里面有个Bind的注解，其定义如下：

@Retention(***CLASS***) @Target(***FIELD***)  
**public** @**interface** Bind {  
 */\*\* View ID to which the field will be bound. \*/* **int**[] value();  
}

从其定义来看，说明它的作用对象是字段，即类中的变量，接收参数是整型数组。Annotation的使用都是通过反射来实现的。不过我们不需要去管反射怎么实现的。我们只要在对象定义前使用这个注解，就可以不要findviewbyid了。

butterknife的原理也是反射，不过它是在编译的时候，通过注解生成新的类文件，再调用findviewbyid，然后再编译成class文件。不是在程序运行的时候通过反射来找到控件的，因此它的效率和不用注解是一样的。

使用方法：

加入依赖：compile 'com.jakewharton:butterknife:7.0.1'

绑定控件：

@Bind(R.id.***test\_img***)

ImageView **test\_iamgeview**;

在oncreate中调用ButterKnife.bind(this);

然后就可以随便使用**test\_iamgeview这个对象了。**

2. 使用Glide加载图片

加入依赖：compile **'com.github.bumptech.glide:glide:3.6.0'**

加载图片：Glide.*with*(**this**)  
 .load(**"http://www.wallcoo.com/animal/v195\_Lively\_Dogs/wallpapers/1280x800/Lively\_Dogs\_wallpaper\_MIX88041\_wallcoo.com.jpg"**)  
 .into(**test\_iamgeview**);

load不仅可以加载网络图片，还可以加载本地图片，注意在manifest中添加网络允许

2016年8月19日11:30:45

1. Andorid的多线程除了AsyncTask和Handler之外，还有如下：

Activity.runOnUiThread（Runnable）

　　 View.post（Runnable）

　　 View.postDelayed（Runnable， long）

　　 Handler.post

　　 Handler.postDelayed（Runnable， long）

Runnable 并不一定是新开一个线程，比如下面的调用方法就是运行在UI主线程中的：

     Handler mHandler=new Handler();   
     mHandler.post(new Runnable(){   
        @Override public void run()   
        { // TODO Auto-generated method stub   
         }   
     });

android在ui线程是没有做线程同步的，它强制让只能在UI线程才能改变ui，

# 2016年8月20日15:24:49

1.真是气死我了，浪费了我一天的时间，本想写个简单的程序的，可是发现colorPrimaryDark无论怎样在我的魅族手机上就是不起作用，在华为手机上就是好好的，在小米平板上更是不行，状态栏一直都是黑色的。以前要想改通知栏颜色都是修改colorPrimaryDark的值，可是现在怎么不行了呢，是不是android studio升级了有bug，我立马又去下载了最新的beta版的android studio，还是一样，后来又回到用几个月前版本的android studio也不行。那么是不是魅族手机自己的问题呢，最近也升级了系统，以前没问题，要不重刷到旧版本的系统，于是我又开始下载旧的rom包。搞了一天都没有搞好，最后偶然打开了魅族手机设置里面的通知栏设置，发现有个沉浸式状态栏选项，发现是勾选的，于是我取消勾选，一切正常，又回到了和之前一样能用colorPrimaryDark修改状态栏颜色的时候了，问题就出在这里，如果勾选了，那么状态栏的颜色就会自动和最顶部颜色一致，使之成为一种颜色，这就是沉浸式状态栏，取消勾选后，就能自由设置颜色了，真是无语啊。

# 2016年8月20日22:55:14

1、onInterceptTouchEvent()是用于处理事件（重点onInterceptTouchEvent这个事件是从父控件开始往子控件传的，直到有拦截或者到没有这个事件的view，然后就往回从子到父控件，这次是onTouch的）（类似于预处理，当然也可以不处理）并改变事件的传递方向，也就是决定是否允许Touch事件继续向下（子控件）传递，一但返回True（代表事件在当前的viewGroup中会被处理），则向下传递之路被截断（所有子控件将没有机会参与Touch事件），同时把事件传递给当前的控件的onTouchEvent()处理；返回false，则把事件交给子控件的onInterceptTouchEvent()

2、onTouchEvent()用于处理事件（重点onTouch这个事件是从子控件回传到父控件的，一层层向下传），返回值决定当前控件是否消费（consume）了这个事件，也就是说在当前控件在处理完Touch事件后，是否还允许Touch事件继续向上（父控件）传递。返回false，则向上传递给父控件，详细一点就是这个touch事件就给了父控件，那么后面的up事件就是到这里touch触发，不会在传给它的子控件。如果父控件依然是false，那touch的处理就给到父控件的父控件，那么up的事件处理都在父控件的父控件，不会触发下面的。

返回true，如果是子控件返回true，那么它的touch事件都在这里处理，父控件是处理不了，因为它收不到子控件传给他的touch，被子控件给拦截了。(这里啰嗦了这么多就是为了加深记忆，这个两个事件理解起来都这么麻烦了，更何况去记，记我肯定是一下子就忘的了^0^)

（注：可能你会觉得是否消费了有关系吗，反正我已经针对事件编写了处理代码？答案是有区别！比如ACTION\_MOVE或者ACTION\_UP发生的前提是一定曾经发生了ACTION\_DOWN，如果你没有消费ACTION\_DOWN，那么系统会认为ACTION\_DOWN没有发生过，所以ACTION\_MOVE或者ACTION\_UP就不能被捕获。）

# 2016年8月25日02:25:51

1.在RelativeLayout中居中添加子元素

**private** RelativeLayout **mHeaderView**;

**mHeaderView** = (RelativeLayout) view.findViewById(R.id.*header\_view*);

RelativeLayout.LayoutParams layoutParams = **new** RelativeLayout.LayoutParams(LayoutParams.***WRAP\_CONTENT***,LayoutParams.***WRAP\_CONTENT***);  
layoutParams.addRule(***CENTER\_IN\_PARENT***);  
**mHeaderView**.addView(header,layoutParams);

2016年8月25日22:41:17

dp和px的相互转换：

**public int** Dp2Px(Context context, **float** dp) {  
 **final float** scale = context.getResources().getDisplayMetrics().**density**;  
 **return** (**int**) (dp \* scale + 0.5f);  
}  
  
**public int** Px2Dp(Context context, **float** px) {  
 **final float** scale = context.getResources().getDisplayMetrics().**density**;  
 **return** (**int**) (px / scale + 0.5f);  
}

# 2016年8月26日17:47:21

1. fitsSystemWindows属性

该属性表示是状态栏是否占位，可以用在activity的主题上，也可用其他view上面。如果用在activity上，且该属性为true，那么状态栏一直占据屏幕，不管是否是透明主题。

如果是用在其他view上，比如toolbar，coordinatelayout，那么当它滑动到屏幕顶端的时候，会自动增加一个状态栏的高度，也就是设置它的paddingTop为状态栏的高度，为的就是不让它和状态栏的内容重叠，影像使用。这样效果并不好，如果我的view是一张图片，就是要它作为状态栏的背景，在设置主题为透明状态后，如果该属性为true，就没有效果了。

在style的设置方法是<item name="android:fitsSystemWindows">true</item>

在其他view的设置方法是：android:fitsSystemWindows="true"

设置透明状态栏：

*<!-- No ActionBar -->*<**item name="windowActionBar"**>false</**item**>  
<**item name="windowNoTitle"**>true</**item**>  
<**item name="android:windowNoTitle"**>true</**item**>  
<**item name="android:windowContentOverlay"**>@null</**item**>  
  
*<!-- 全屏显示界面-->*<**item name="android:windowFullscreen"**>false</**item**>  
*<!-- 状态栏透明-->*<**item name="android:windowTranslucentStatus"**>true</**item**>  
*<!-- 全屏显示-->*<**item name="android:fitsSystemWindows"**>false</**item**>

# 2016年8月27日23:22:54

1. 自定义viewgroup中onMeasure的用法

在这里首先是获取子view的测量结果，用getchildat()来获取子view，然后用measureChild来测量大小，然后才能决定本viewgroup的大小，如果本viewgroup的宽度或高度设置的是wrapcontent，那么它实际的宽度和高度就和子view有关系了，需要获得子view的总宽度，在加上margin等就是本viewgroup的实际宽度，高度一样。如果本viewgroup的宽度设置的是精确值，或者matchparent，那么它的宽度就和子view没关系了，直接调用super.onMeasure(widthMeasureSpec, heightMeasureSpec);方法即可，或者用setMeasuredDimension方法设置大小。

测量子view的大小，并不能对子view的大小产生影响，只是获取它的宽度和高度，方便viewgroup设置大小，可以用measureChildWithMargins和measureChild方法

super.onMeasure(widthMeasureSpec, heightMeasureSpec);方法就是设置自己的大小，如果自己的大小受到子view的影响，就要先获取子view的大小，在设置自己的大小

在viewgroup中的onMeasure方法中：

// 获得它的父容器为它设置的测量模式和大小  
int sizeWidth = MeasureSpec.getSize(widthMeasureSpec);  
int sizeHeight = MeasureSpec.getSize(heightMeasureSpec);  
int modeWidth = MeasureSpec.getMode(widthMeasureSpec);  
int modeHeight = MeasureSpec.getMode(heightMeasureSpec);

设置viewgroup的宽度和高度

setMeasuredDimension((modeWidth == MeasureSpec.EXACTLY) ? sizeWidth  
 : width, (modeHeight == MeasureSpec.EXACTLY) ? sizeHeight  
 : height);

**onMeasure方法在View.java中的实现**

**protected void** onMeasure(**int** widthMeasureSpec, **int** heightMeasureSpec) {  
 setMeasuredDimension(*getDefaultSize*(getSuggestedMinimumWidth(), widthMeasureSpec),  
 *getDefaultSize*(getSuggestedMinimumHeight(), heightMeasureSpec));  
}

# 2016年8月31日23:16:07

1. 自定义view中动画的执行

先定义一个动画Animation mAnimation;

配置完动画后，不能直接使用mAnimation.start()方法运行，这样是没有效果的，要用view的startAnimation方法。

**this.startAnimation(mAnimation);**

2017年7月9日11:53:20

1. 好久没写安卓程序了，什么都忘了。

Android studio中添加文件夹和普通的是不一样的，右键选择添加文件夹只能添加res的资源文件夹。若想添加java源代码的文件夹，是不一样的，这里不叫文件夹，而是叫包，即package。右键选择new，再选择package就可以了。

2017年7月20日00:42:08

1.要在22版本的编译器下使用support.design，需要在gradle里面设置compile 'com.android.support:design:22.2.0'

2018年2月11日21:09:22

1. 获取Android开发的sha1值

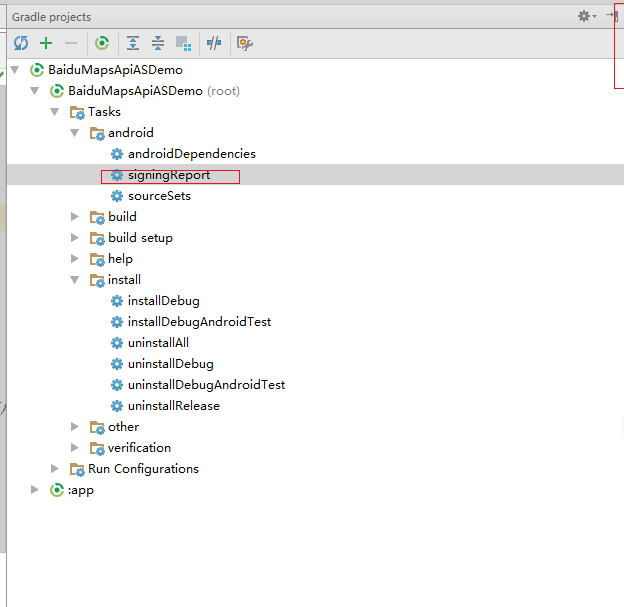
Sha1值与不同的开发环境有关，如果你的软件使用了sha1值，并且换了开发环境（比如换了电脑），就得重新查询sha1值。百度地图使用了sha1来加密。需要获取本地的sha1和包名来申请ak。

获取方式有两种：（1） 命令行进入.android目录，比如我的：C:\Users\Administrator\.android

输入 keytool -list -v -keystore即可

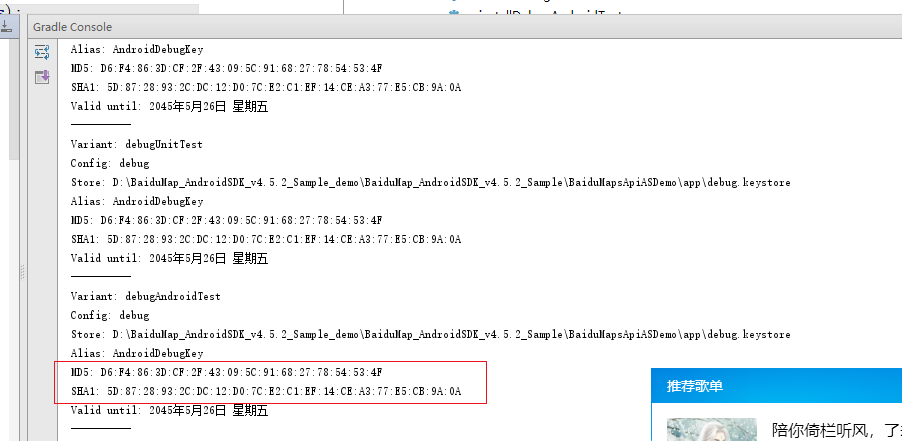
（2）使用Android studio来获取

展开右边的gradle



双加signingReport

打开gradle project就可以看到sha1值了



1. 关于百度地图官方demo报230错误的问题

我已经弄了好几天了，一直都是密钥获取错误，代码为230。理论上是sha1或者包名没有弄正确。

一般是通过debug.keystore文件来获取sha1值，sha1值存储在这个文件里面。换了这个文件sha1就不一样。比较奇怪的是官方demo的app文件夹下也有这个文件。如果把这个文件删了或者换了，就根本不能下载程序到手机里面，更别说230错误了。若是自己建立的工程，在app文件夹下就没有这个文件。之前我是使用的.android目录下的debug.keystore来获取sha1值的。但也用了论坛提供的一个获取sha1的app来获取的，两个结果不一样。后者获取的是工程下的sha1值，和在Android studio里面获取结果是一样的，这两种sha1我都试过了，一样不行。后来我在申请的时候将开发板sha1和发布版sha1都填上了，重新申请了一个密钥。就成功了，也不知道是不是这个的问题。

2018年2月17日18:45:01

1. 让Android studio自动加载so库，

在app/src/main/下创建jniLibs文件夹，将so文件放入这个文件夹下，Android studio会自动加载这个文件夹下的so库。

放入其他文件夹也可以，需要在gradle里面配置路径。

sourceSets{

main{

jniLibs.srcDir 'libs'

//说明so的路径为该libs路径，关联所有地图SDK的so文件

}

}

2018年2月27日19:37:03

1. 实现自动获取权限功能

华为的高版本的手机默认不会提示获取权限，如果访问定位等系统资源，则会出现异常。以前的低版本的手机就会自动获取权限。

然后我在百度轨迹的demo里面看到了自动获取权限的供。只需要在mainacitivity的onStart方法里面获取就可以了。

**protected void** onStart() {  
 **super**.onStart();   
 List<String> permissions = **new** ArrayList<>();  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M*** && isNeedRequestPermissions(permissions)) {  
 requestPermissions(permissions.toArray(**new** String[permissions.size()]), 0);  
 }  
  
}  
**private boolean** isNeedRequestPermissions(List<String> permissions) {  
 *// 定位精确位置* addPermission(permissions, Manifest.permission.***ACCESS\_FINE\_LOCATION***);  
 *// 存储权限* addPermission(permissions, Manifest.permission.***WRITE\_EXTERNAL\_STORAGE***);  
 *// 读取手机状态* addPermission(permissions, Manifest.permission.***READ\_PHONE\_STATE***);  
 **return** permissions.size() > 0;  
}  
**private void** addPermission(List<String> permissionsList, String permission) {  
 **if** (Build.VERSION.***SDK\_INT*** >= Build.VERSION\_CODES.***M*** && checkSelfPermission(permission) != PackageManager.***PERMISSION\_GRANTED***) {  
 permissionsList.add(permission);  
 }  
}

2018年2月28日14:57:02

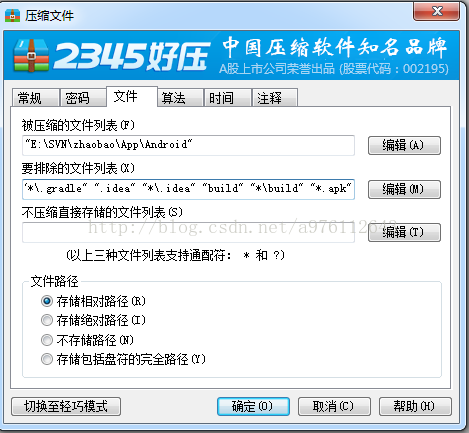
1. 新版本的Android studio截图功能

3.0版本的Android studio的截图功能和以前不一样，不好找了。新版本的在logcat里面，有一个照相机的图标是截图的，还有一个视频播放的图标是录屏的。很恶心的是如果你对logcat的界面的高度拉的很低，这两个图标就看不到了，需要往上拉才能看到。

2018年3月1日02:17:08

1. Android studio源码打包技巧

一般开发时，会在整个工程里面输出很多与代码无关的数据，当需要移植代码给别人时，数据量会非常大，动不动就几百兆，实际代码去很少。尤其是build文件夹下的文件都是没用的。一个非常好的打包技巧是使用压缩工具，比如好压。设置它的压缩文件忽略规则，排除掉不想要的文件。



".gradle" "\*\.gradle" ".idea" "\*\.idea" "build" "\*\build" "\*.apk"

这样可以大大减轻打包后文件的大小。并且移植后打代码，重新编译完全不影响运行。

2018年3月1日12:16:32

1. 百度地图生成apk后，无法显示的问题

由于开发的时候使用的是debug.keystore这个来加密的生成sha1的，百度地图key的申请需要sha1这个字符串。而自己生成apk的时候是要创建一个加密文件的，后缀是jks，而且要设置密码。这个jks里面包含了新的sha1值，用这个jks文件生成的apk的sha1和用Androidstudio开发时写入的sha1是不一样的。因此需要在百度申请一个和jks文件

相同的sha1值的key，这样生成的apk的sha1才能和key匹配，这样才能正常显示地图。

获取jks文件的sha1的方法

keytool -list -v -keystore jks文件名.jks

2018年3月18日16:22:12

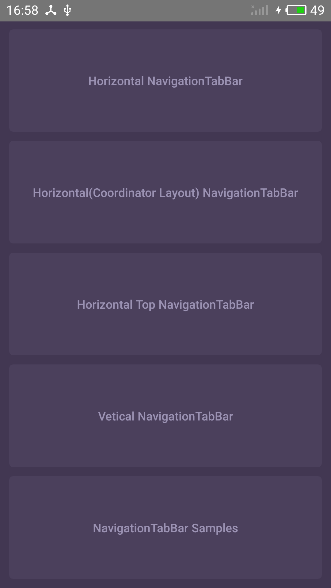
1. Android自定义标题栏，使用了安卓自带的一个方法

**mActivity**.getWindow().setFeatureInt(Window.***FEATURE\_CUSTOM\_TITLE***, R.layout.***toolbar\_home\_login***);

1. 可以用glide来加载图片

<https://github.com/bumptech/glide>

1. 给Android的textview设置圆形的背景



这里面的每一个元素都是一个textview。主要设置它的background属性

<**TextView  
 android:id="@+id/btn\_horizontal\_coordinator\_ntb"  
 android:layout\_width="match\_parent"  
 android:layout\_height="0dp"  
 android:layout\_weight="1"  
 android:text="Horizontal(Coordinator Layout) NavigationTabBar"  
 android:textStyle="bold"  
 android:gravity="center"  
 android:textColor="#9b92b3"  
 android:layout\_marginStart="10dp"  
 android:layout\_marginLeft="10dp"  
 android:layout\_marginEnd="10dp"  
 android:layout\_marginRight="10dp"  
 android:layout\_marginBottom="10dp"  
 android:background="@drawable/bg\_round\_rect"**/>

bg\_roun\_rect.xml的内容如下：

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape xmlns:android="http://schemas.android.com/apk/res/android"  
 android:dither="true"  
 android:shape="rectangle"**>  
 <**corners android:radius="4dp"**/>  
 <**solid android:color="#4b405c"**/>  
</**shape**>

这个例子里面还有特别好的效果，点击按钮后，按钮会有缩小的效果。

设置了它的监听事和动画效果。

@Override  
**public void** onClick(**final** View v) {  
 ViewCompat.*animate*(v)  
 .setDuration(200)  
 .scaleX(0.9f)  
 .scaleY(0.9f)  
 .setInterpolator(**new** CycleInterpolator())  
 .setListener(**new** ViewPropertyAnimatorListener() {  
 @Override  
 **public void** onAnimationStart(**final** View view) {  
  
 }  
  
 @Override  
 **public void** onAnimationEnd(**final** View view) {  
 **switch** (v.getId()) {  
 **case** R.id.***btn\_horizontal\_ntb***:  
 startActivity(  
 **new** Intent(MainActivity.**this**, HorizontalNtbActivity.**class**)  
 );  
 **break**;  
 **case** R.id.***btn\_horizontal\_top\_ntb***:  
 startActivity(  
 **new** Intent(MainActivity.**this**, TopHorizontalNtbActivity.**class**)  
 );  
 **break**;  
 **case** R.id.***btn\_horizontal\_coordinator\_ntb***:  
 startActivity(  
 **new** Intent(MainActivity.**this**, HorizontalCoordinatorNtbActivity.**class**)  
 );  
 **break**;  
 **case** R.id.***btn\_vertical\_ntb***:  
 startActivity(  
 **new** Intent(MainActivity.**this**, VerticalNtbActivity.**class**)  
 );  
 **break**;  
 **case** R.id.***btn\_samples\_ntb***:  
 startActivity(  
 **new** Intent(MainActivity.**this**, SamplesNtbActivity.**class**)  
 );  
 **break**;  
 **default**:  
 **break**;  
 }  
 }  
  
 @Override  
 **public void** onAnimationCancel(**final** View view) {  
  
 }  
 })  
 .withLayer()  
 .start();  
}

**private class** CycleInterpolator **implements** android.view.animation.Interpolator {  
  
 **private final float mCycles** = 0.5f;  
  
 @Override  
 **public float** getInterpolation(**final float** input) {  
 **return** (**float**) Math.*sin*(2.0f \* **mCycles** \* Math.***PI*** \* input);  
 }  
}

2018-3-19 11:11:12

1. 设计带图片的textview，比如微博底部的导航条按钮，图片下面是文字，实际上只用了一个textview，设置了它的drawableTop属性

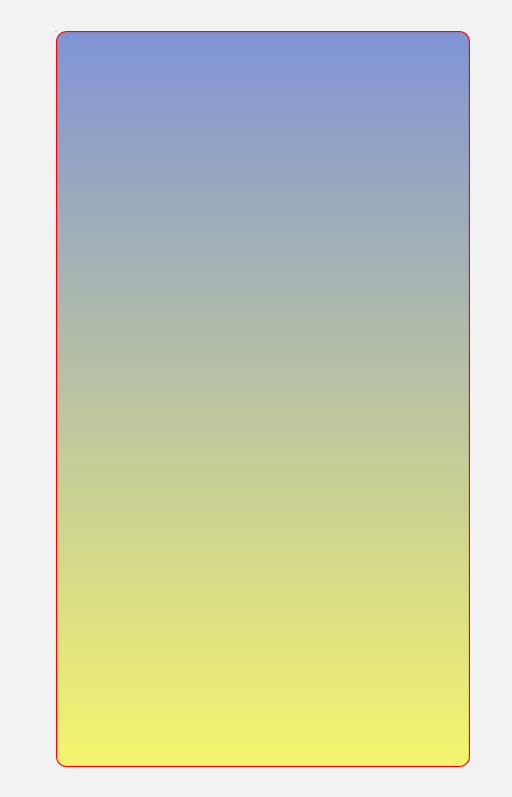
<**TextView  
 android:id="@+id/tv\_home"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:drawableTop="@drawable/tabbar\_home\_auto"  
 android:gravity="center"  
 android:text="首页"  
 android:textSize="10sp"** />

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

2018年3月20日11:09:42

1. Android在自定义shape里面实现背景渐变效果，即阴影效果。使用gradient属性

*<?***xml version="1.0" encoding="utf-8"***?>*<**shape xmlns:android="http://schemas.android.com/apk/res/android"**>  
 <**stroke  
 android:width="1dp"  
 android:color="#ff0101"**/>  
 <**corners android:radius="10dp"** />  
 <**solid android:color="#985d5a68"**/>  
 <**gradient  
 android:startColor="#f5f513"  
 android:endColor="#3152c1"  
 android:angle="90"**/>  
  
</**shape**>

这个效果比原来好多了。

2018年3月20日15:26:23

1. 在imageview中设置layout\_margin属性，可以设置图片的缩放效果。这个值越大，图片越小，应为它是设置图片的外边距的，而padding是设置内边距的。

2018年3月21日16:15:02

1. 使用$来快速绑定view

在activity里面定义如下的方法：

public <T> T $(int id) {  
 return (T) findViewById(id);  
}

调用方法：

mAdd = $(R.id.activity\_add);

2018年3月21日21:57:36

1. 再次回顾android的动画

最近要实现一个功能，点击一个按钮，下面弹出一个信息框，并且这个按钮上移，使之不会和对话框重叠。之前考虑到用dialogplus，实在是大材小用了。直接添加一个普通的viewgroup，比如LinearLayout，然后设置它的visibility为visible或者view.gone,（前提是button是和对话框从上到下排列在LinearLayout里面的），这样控制对话框的visibility属性，button就永远不会和它重叠。但是这样是没有动画效果的。每一个view都有个startAnimation（）方法，参数为Animation对象，这个对象可以从xml里面获取，比如：

private Animation getOutAnimation() {  
 // int res = (outAnimation == INVALID) ? Utils.getAnimationResource(this.gravity, false) : outAnimation;  
 int res= R.anim.slide\_out\_bottom;  
 return AnimationUtils.loadAnimation(mContext, res);  
}

<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.com/apk/res/android"  
 android:shareInterpolator="false"  
 >  
  
 <translate  
 android:duration="300"  
 android:fromXDelta="0%"  
 android:fromYDelta="100%"  
 android:toXDelta="0%"  
 android:toYDelta="0%"  
 />  
</set>

然后调用方法是

*ctrView.startAnimation(outAnim);*

并且，outAnim这个动画对象还可以设置对动画的监听，比如开始，结束等。

*outAnim.setAnimationListener(new Animation.AnimationListener()*

这样就实现了动画效果，但是这有个很大的缺陷，就是不能同时设置多个动画。

比如传递多个view，并且同时调它们的startAnimation方法。结果是只能实现一个动画效果。

解决办法是使用AnimatioSet

AnimatorSet **animatorSet**= **new** AnimatorSet();

ObjectAnimator animator1 = ObjectAnimator.*ofFloat*(**this**,**"translationY"**,100f,0f);  
ObjectAnimator animator2 = ObjectAnimator.*ofFloat*(**ctrView**,**"translationY"**,100f,0f);  
 **animatorSet**.play(animator1).with(animator2);  
 **animatorSet**.setDuration(300);  
 **animatorSet**.start();

这样就能实现多个view同时实现动画效果了。

现在看看这个方法：

ObjectAnimator.*ofFloat*(**this**,**"translationY"**,100f,0f);  
第一个参数设置要控制哪个对象，

第二个参数指定这个动画要操作这个控件的哪个属性

第三个参数是可变参数，指这个属性值从哪个值变化到哪个值。

控制旋转

1. ObjectAnimator animator = ObjectAnimator.ofFloat(tv,"rotation",0,180,0);
2. animator.setDuration(2000);
3. animator.start();

2018年3月22日10:48:37

* 1. 百度地图缩放供的实现

最近写了个自定义的控件，来操作地图的缩放效果，没有用地图自带的缩放按钮，它的按钮位置不合适，且样式也不能修改。

MapStatusUpdate zoomin= MapStatusUpdateFactory.*zoomIn*();  
*// mBaiduMap.setMapStatus(zoomin); //没有平滑的效果  
 mBaiduMap*.animateMapStatus(zoomin);

之前用setMapStatus，没有平滑效果，直接就放大道某个级别了，效果不好，使用animateMapStatus就可以了。

2018年3月22日22:52:40

1. 动画集合还可以用AnimationSet

AnimationSet animationSet = **new** AnimationSet(**true**);  
ScaleAnimation scaleAnimation = **new** ScaleAnimation(0, 3f, 0, 3f,  
 Animation.***RELATIVE\_TO\_SELF***, 0.5f,  
 Animation.***RELATIVE\_TO\_SELF***,  
 0.5f);  
AlphaAnimation alphaAnimation = **new** AlphaAnimation(1, 0);  
animationSet.setDuration(300);  
animationSet.addAnimation(scaleAnimation);  
animationSet.addAnimation(alphaAnimation);  
animationSet.setFillAfter(**true**);  
child.startAnimation(animationSet);

2018年3月23日09:37:23

1. Android实现带文字的图片

使用textview

<**TextView  
 android:id="@+id/tv\_message"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_centerInParent="true"  
 android:drawableTop="@drawable/tabbar\_message\_auto"  
 android:gravity="center"  
 android:text="消息"  
 android:textSize="10sp"** />



1. 要想将子元素放到父容器的右边，需要这样设置

**android:layout\_alignParentRight="true"**

但是父容器必须是Relativelayout，不能是LinearLayout。

如果非得在LinearLayout里面放到右边，可以将子元素用Relativelayotu包裹，再放到LinearLayout里面。

2018年3月23日21:14:36

1. 今天写代码真是遇到鬼了，遇到的bug毫无道理可言。虽然我现在解决了，实现了功能，但我仍然不知道到底是怎么回事，真不想深入研究了。

本来是想基于dialogplus设计一个个侧面弹出的对话框的，用于显示地图的图层控制工具。

我原来的 guijidiaocha的项目里面有个dialogplus的库，这是一个比较完善的弹框，非常完善，我之前用他做了一个简单的顶部弹出窗体来控制图层显示。

现在同样的移植，就报了莫名其妙的错误。

在viewholder的getView方法里面，

View view = inflater.inflate(R.layout.*dialog\_view*, parent, **false**);  
View outMostView = view.findViewById(R.id.*dialogplus\_outmost\_container*);  
outMostView.setBackgroundResource(**backgroundResource**);

调试发现在遇到第三句就出错了，提示outMostView是空的。也就是说findViewById没有找打对应的View，怎么可能呢，明明id都是正确了，也没有改过什么别的地方啊。一直都弄不出来。

网上也就解释findViewByid返回null的解决办法，现在我知道应该重新建立以布局的xml文件，而不是从原来的项目里复制过来。

实在是找不到怎么办了，然后我想到了我以前写了一个简单的基于dialogplus的customdialog，去掉了很多没用的功能，比如只有一viewholder，没有什么listholder和gridholder，holder里面也去掉了header和footer，这些根本不需要，如果用户需要的话，只需要指定一个contentView，加载到viewholder里面就可以了。

然后我就开始移植精简版的dialogplus，同样遇到了奇葩的问题，但是这次出现问题的地方却不一样。

**contentView**=inflater.inflate(**viewResourceId**,parent,**false**);

这句话又返回null了，也就是说没有解析到这个layout，可是这个**viewResourceId**也是正常的啊，并且解析的过程所涉及的类也与viewholder无关啊。是不是inflater来自fragment，而不是activity的原因啊。于是我把mainfrag的onCreateView方法里面接卸layout的方法里面的id换成了这个id，也就是myholder\_content.xml，发现程序完全崩溃了，也就是在mainfragment的oncreateView里面也解析不到myholder\_content.xml。于是我新建了一个布局的xml文件，解析新的布局问题，没有问题，并且将之前的viewholder里面解析xml的换成新的xml，一切正常。问题就出现在myholder\_content.xml这个文件。可是这是为什么呢，不就是一个xml文件吗，想不通。

1. 原来我还在dialogplus上增加了好多功能，比如，增加了outmost的边距，outmost是最外面的容器，点击空白处可以取消对话框，默认它是整个屏幕的范围，如果设置了它的边距，点击它外面的部分是不会关闭对话框的。

除此之外还设置了内部容器的高度宽度和边距。主要是设置framelayout的params。这些都是原来的dialogplus不具备的功能。

DialogPlus dialog = DialogPlus.*newDialog*(**mContext**)  
 .setContentHolder(**holder**)  
 .setCancelable(**true**) *//是否能点空白或后退键取消* .setGravity(Gravity.***RIGHT***)  
 *// .setOutMostMargin(100,200,200,200)  
// .setPadding(20,20,20,20)  
 // .setContentMargin(0,250,0,0)  
 // .setContentHeight(900)* .setContentHeight(ViewGroup.LayoutParams.***MATCH\_PARENT***)  
 .setContentWidth(700)  
  
 .setOnCancelListener(cancelListener) *//可以不要* .create();  
  
 dialog.show();

2018年3月25日21:29:32

1. 在自定义ViewGroup里面实现智能添加一个子View

通常在自定义viewgroup里面使用addView方法添加子元素，但有时候不希望添加过多的子元素，因此就需要重写父类的addVeiw方法，在里面使用getChildCount方法来判断子元素的个数。

@Override  
**public void** addView(View child) {  
 *// super.addView(child);  
 //注销掉自带的addView方法，使用addContent方法，避免重复添加*}  
  
**public void** addContent(View v)  
{  
 **if**(getChildCount()>=1)  
 {  
 **return**;  
 }  
 **super**.addView(v);  
}

我的做法是定义了一个新的添加元素的方法addContent，里面判断如果子元素个数大于等于1，就不添加，但是如果该自定义viewgroup的对象直接调用addView方法就会出现添加多个元素的方法，因此需要复写addView方法，然后在里面注销掉父类的addView方法。如果仅仅是这样，那么addContent方法调用addView方法，也会调用复写过的addView方法，这样就一个元素都不能添加了。因此在addContent方法里面需需要写成super.addView(v)的形式。

2018年3月26日21:03:10

1. Android中使用handler在子线程中更新ui

Android中有些功能只能在子线程或者异步任务中执行，比如http请求之类的。

在主线程中顶一个Handler子类，重写handleMessage方法。

然后在子线程中发送消息。消息的传递使用Message类，并且message类中包含了bundle对象，也就是说使用bundle来包裹数据，让后封装到message里面，然后发送给handler。

**private** Handler **myhandler**;

**myhandler** = **new** Handler(){  
 @Override  
 **public void** handleMessage(Message msg) {  
 **super**.handleMessage(msg);  
  
 Bundle bundle= msg.getData();  
 String res= bundle.getString(**"responseMsg"**);  
 **resTv**.setText(res);  
  
 }  
};

子线程中发送消息的过程：

Message msg= **myhandler**.obtainMessage();  
Bundle bundle = **new** Bundle();  
bundle.putString(**"responseMsg"**,stringBuffer.toString());  
msg.setData(bundle);  
msg.sendToTarget();

1. Android使用HttpurlConnection实现get请求，得到返回结果。
2. 首先是服务器端后台程序的设计，我使用了playframework。

在routes定义了一个方法的映射关系

GET /login @controllers.Application.login()

然后再Application类里面定义了一个login方法，这样如果客户端使用http向服务器发起了get请求，请求的方法是login，然后就会执行Application中的login方法，在这个方法里面返回数据给客户端。

**public** Result login()

{

DynamicForm requestData= Form.*form*().bindFromRequest();

String getsex= requestData.get("sex");

**return** *ok*("hello:"+getsex);

//return ok("hello:"+getsex).as("text/html");

}

这里在方法里面获取了请求的参数，使用form请求的方式。

比如客户端发起的请求url是<http://localhost:9000/login?sex=girl>

那么这个函数就能得到请求的参数sex的值为gril。然后通过判断做出具体响应。

1. 接下来是Android客户端通过httpurlconnetion发起get请求，get请求比post请求简单多了。post请求要设置很多请求头信息，很容易出错。

要注意Android中网络的请求不能再主线程中运行。

**private void** doGet(String s) {  
 String geturl =**urlAddress**+**method**+**"?sex="**+s;  
 **try** {  
 URL url= **new** URL(geturl);  
 HttpURLConnection httpURLConnection= (HttpURLConnection) url.openConnection();  
 httpURLConnection.connect();  
 **if**(httpURLConnection.getResponseCode()==200)  
 {  
 InputStream is =httpURLConnection.getInputStream();  
 BufferedReader bufferedReader= **new** BufferedReader(**new** InputStreamReader(is));  
 StringBuffer stringBuffer = **new** StringBuffer();  
 String readline=**""**;  
 **while**((readline=bufferedReader.readLine())!=**null**)  
 {  
 stringBuffer.append(readline);  
 }  
 is.close();  
 bufferedReader.close();  
 httpURLConnection.disconnect();  
 *// res.setText(stringBuffer.toString());  
 //子线程不能直接访问ui* Log.*e*(**"http"**, **"doGet: "**+ stringBuffer.toString());  
  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,stringBuffer.toString());  
 msg.setData(bundle);  
 msg.sendToTarget();  
  
 }**else** {  
 *// res.setText("请求失败");* Log.*e*(**"http"**, **"doGet: "**+ **"访问失败"**);  
 }  
  
 } **catch** (MalformedURLException e) {  
 e.printStackTrace();  
 }**catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
}

首先定义一个URL对象，也就是请求的地址。

然后定义一个HttpURLConnection对象，但是这个对象是通过url的openConnection方法得到的。

URL url= new URL(geturl);  
HttpURLConnection httpURLConnection= (HttpURLConnection) url.openConnection();

然后通过connet方法就实现了get请求

httpURLConnection.connect();

接下来就是得到相应信息了。

httpURLConnection.getResponseCode()==200

200是请求成功的响应码。

然后通过输入流来获取响应信息。使用BufferedReader来读取数据。读到的结果存储在StringBuffer里面。

读完后也要关闭输入流、读入流，并关闭httpurlconnection连接。

is.close();  
bufferedReader.close();  
httpURLConnection.disconnect();

2018年3月26日21:48:29

1.在API 23中，Google已经移除了移除了Apache HttpClient相关的类 。谷歌推荐使用HttpUrlConnection，如果要继续使用需要Apache HttpClient，有两种方法。

Eclipse下libs里添加org.apache.http.legacy.jar，

Android studio里在相应的module下的build.gradle中加入即可。

**android** {

useLibrary 'org.apache.http.legacy'

}

1. Android中使用httpclient方法比使用httpurlconnection方法更简单，httpurlconnection更底层，需要知道很多细节原理。可以httpclient是apache的，不是Android自带的。

需要在gradle里面添加useLibrary 'org.apache.http.legacy'

如果没有添加，下面这句话就会报错，

HttpGet httpGet = new HttpGet(geturl);

添加后HttpGet类就能通过包导入了。

**private void** doGet2(String s)  
{  
 String geturl =**urlAddress**+**method**+**"?sex="**+s;  
 HttpGet httpGet = **new** HttpGet(geturl);  
 **try** {  
 HttpResponse httpResponse= **new** DefaultHttpClient().execute(httpGet);  
 **if**(httpResponse.getStatusLine().getStatusCode() == 200)  
 {  
 String result = EntityUtils.*toString*(httpResponse.getEntity()) ;  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,result);  
 msg.setData(bundle);  
 msg.sendToTarget();  
 }  
 **else** {Log.*e*(**"http"**, **"doGet: "**+ **"访问失败"**);  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

可以看出使用httpclient方法非常简单。注意还是需要在子线程里面使用，并且用handler更新ui。

1. 使用httpclient的post方法请求服务器

**private void** doPost(String s)  
{HttpPost httpPost= **new** HttpPost(**urlAddress**+**method**);  
 List<NameValuePair> params = **new** ArrayList<>();  
 params.add(**new** BasicNameValuePair(**"sex"**,s));  
 **try** {  
 httpPost.setEntity(**new** UrlEncodedFormEntity(params, HTTP.***UTF\_8***));  
 HttpResponse httpResponse = **new** DefaultHttpClient().execute(httpPost);  
 **if**(httpResponse.getStatusLine().getStatusCode() == 200)  
 {  
 String result = EntityUtils.*toString*(httpResponse.getEntity()) ;  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,result);  
 msg.setData(bundle);  
 msg.sendToTarget();  
 }  
 } **catch** (UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

和get方法相比，除了请求方法设置稍微不一样外，请求的执行和响应的处理是一模一样的。

HttpPost方法请求需要使用键值对来存储请求参数。

然后还有一个非常关键的地方要修改：服务器端。之前爱服务器端都是设置为GET方式请求的，而没有设置POST方式。如果没有试着POST方式，却用POST方式请求，当然无法请求，因此需要在routes里面添加：POST /login @controllers.Application.login()

2018年3月26日23:04:30

1. OKHttp

OKHttp是支持SPDY协议的高效的HTTP库

官网：<http://square.github.io/okhttp/>

Android中只用OKHttp，需要在gradle中添加依赖

compile 'com.squareup.okhttp3:okhttp:*(insert latest version)*'

版本号：比如：3.2.0

**maven**中添加的方法是<dependency>

<groupId>com.squareup.okhttp3</groupId>

<artifactId>okhttp</artifactId>

<version>*(insert latest version)*</version>

</dependency>

1. POST TO A SERVER

This program posts data to a service. [Full source](https://raw.github.com/square/okhttp/master/samples/guide/src/main/java/okhttp3/guide/PostExample.java).

public static final MediaType JSON

= MediaType.parse("application/json; charset=utf-8");

OkHttpClient client = new OkHttpClient();

String post(String url, String json) throws IOException {

RequestBody body = RequestBody.create(JSON, json);

Request request = new Request.Builder()

.url(url)

.post(body)

.build();

Response response = client.newCall(request).execute();

return response.body().string();

}

1. 使用OKHttp实现get请求

先提前声明client对象

**private** OkHttpClient **okHttpClient**;

**okHttpClient**= **new** OkHttpClient();

*//使用OKhttp***private void** doGet4(String s) {  
 **final** String geturl = **urlAddress**+**method**+**"?sex="**+s;  
 Request request = **new** Request.Builder().url(geturl).build();  
 **try** {  
 Response response=**okHttpClient**.newCall(request).execute();  
 **if**(response.isSuccessful()) {  
 String result= response.body().string();  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,result);  
 msg.setData(bundle);  
 msg.sendToTarget();  
 }  
 **else** {  
 *// res.setText("请求失败");* Log.*e*(**"http"**, **"doGet: "**+ **"访问失败"**);  
 }  
  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
}

2018年3月27日09:56:01

1. 使用OKHttp发送JSON请求，服务器解析JSON数据并返回结果。
2. 服务器端，参考play官网解析JSON请求的例子

//解析json请求

@BodyParser.Of(BodyParser.Json.**class**)

**public** Result say()

{

JsonNode json = *request*().body().asJson();

ObjectNode result = Json.*newObject*();

String name = json.findPath("name").textValue();

**if**(name == **null**) {

result.put("status", "KO");

result.put("message", "Missing parameter [name]");

**return** *badRequest*(result);

} **else** {

result.put("status", "OK");

result.put("message", "Hello " + name);

**return** *ok*(result);

}

}

1. Android客户端

先设置content-type为json

**public static final** MediaType ***JSON*** = MediaType.*parse*(**"application/json; charset=utf-8"**);

*//使用okhttp，发送json* **private void** doPost4() {String json=**"{\"status\":\"OK\",\"message\":\"Hello jjj\",\"name\":\"guozhang\"}"**;  
 RequestBody body = RequestBody.*create*(***JSON***,json);  
  
 Request request = **new** Request.Builder()  
 .url(**urlAddress**+**"say"**)  
 .post(body)  
 .build();  
  
 **try** {  
 Response response = **okHttpClient**.newCall(request).execute();  
 **if**(response.isSuccessful()) {  
  
 String result= response.body().string();  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,result);  
 msg.setData(bundle);  
 msg.sendToTarget();  
 }  
 **else** {  
 *// res.setText("请求失败");* Log.*e*(**"http"**, **"doGet: "**+ **"访问失败"**);  
 }  
  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
  
  
 }

Android studio有个比较智能的地方就是，如果粘贴的字符串有特殊字符，它能自动地给你添加转义符。比如有引号，粘贴过去后会自动添加斜杠。

1. JSON

JSON语法规则：是javascript对象表示语法的子集

1. 数据在键值对中
2. 数据由逗号分隔
3. 花括号保存对象
4. 方括号保存数组

JSON值可以是：

1. 数字（整数或者浮点数）
2. 字符串（在双引号中）
3. 逻辑值（true或false）
4. 数组（在方括号中）
5. 对象（在花括号中）
6. null

*//解析JSON数据***private void** parseData(String result)  
{  
 List<Member> list = **new** ArrayList<>();  
  
 **try** {  
 JSONObject jsonObject= **new** JSONObject(result);  
 JSONArray jsonArray = jsonObject.getJSONArray(**"list"**);  
 **for**(**int** i=0;i<jsonArray.length();i++)  
 {  
 JSONObject object= jsonArray.getJSONObject(i);  
 Member member= **new** Member();  
 member.setName(object.getString(**"name"**));  
 member.setSex(object.getString(**"sex"**));  
 list.add(member);  
 }  
 **for**(Member member :list)  
 {  
 member.toString();  
 }  
  
 } **catch** (JSONException e) {  
 e.printStackTrace();  
 }  
  
}

Java下常见的Json类库有Gson、JSON-lib和Jackson等，Jackson相对来说比较高效，在项目中主要使用Jackson进行JSON和Java对象转换，下面给出一些Jackson的JSON操作方法。

之前百度轨迹查询的解析就用的是Gson。play官网上的JSON解析用的是jackson。

上面的都是现成的框架，而麦子学院的视频教程里面用的是原生的json对象。JSONObject，这种方式就比较复杂了。

jackson官网：<https://github.com/FasterXML/jackson>

android端常用的json解析工具是GSON,和Fast-json

2018年3月27日11:38:27

1. 一个很好的图标下载网站 <http://www.iconfont.cn/>

是阿里巴巴提供的

1. Android常用网络框架
2. volley <https://github.com/mcxiaoke/android-volley>
3. Android-async-http <https://github.com/loopj/android-async-http>
4. Afinal <https://github.com/yangfuhai/afinal>
5. xutils <https://github.com/wyouflf/xUtils3>
6. Okhttp <https://github.com/square/okhttp>
7. retrofit <https://github.com/square/retrofit>

最后两个非常流行，而xutils是国产的，缺点是功能太复杂了。

测试了一下，无论是在gradle还是直接导入源码，xutils都要报错，太恶心了。

1. retrofit的导入方法

compile 'com.squareup.retrofit2:retrofit:*(insert latest version)*'

compile 'com.squareup.retrofit2:retrofit:2.0.2'

发现用gradel始终都无法导入，各种错误，百度也没有找到解决方法。

发现retrofit官网提供jar包下载，<https://repo1.maven.org/maven2/com/squareup/retrofit2/retrofit/2.4.0/retrofit-2.4.0.jar>

于是通过导入jar包的方式导入retrofit

2018年3月27日22:00:12

1. okhttp

使用Request的post方法来提交请求体RequestBody   
**POST提交键值对**   
OkHttp也可以通过POST方式把键值对数据传送到服务器

1. 使用okhttp提交form表单

*//使用OKHttp的post方法提交表单***private void** doPost5()  
{  
 FormBody formBody = **new** FormBody.Builder()  
 .add(**"name"**,**"guozhang"**)  
 .add(**"sex"**,**"nan"**)  
 .add(**"age"**,**"26"**)  
 .build();  
  
 Request request = **new** Request.Builder()  
 .url(**urlAddress**+**"take"**)  
 .post(formBody)  
 .build();  
  
 Call call = **okHttpClient**.newCall(request);  
  
 call.enqueue(**new** Callback() {  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 Log.*e*(**"ok"**, **"onFailure: "**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
  
 **if**(response.code()==200)  
 {  
 String result= response.body().string();  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,result);  
 msg.setData(bundle);  
 msg.sendToTarget();  
 }  
 }  
 });  
}

服务器端解析表单：

**public** Result take()

{

DynamicForm requestData= Form.*form*().bindFromRequest();

String name= requestData.get("name");

String sex= requestData.get("sex");

**return** *ok*("hello:"+name+",sex:"+sex);

}

1. 使用GSON将对象转换为string

//使用Gson 添加 依赖 compile 'com.google.code.gson:gson:2.8.1'

Gson gson = new Gson();

//使用Gson将对象转换为json字符串

String json = gson.toJson(book);

将json字符串转换为string

**public class** GsonService {  
  
 **public static** <T> T parseJson(String jsonString, Class<T> clazz) {  
 T t = **null**;  
 **try** {  
 Gson gson = **new** Gson();  
 t = gson.fromJson(jsonString, clazz);  
 } **catch** (Exception e) {  
 *//* ***TODO: handle exception*** e.printStackTrace();  
 System.***out***.println(**"解析json失败"**);  
 }  
 **return** t;  
  
 }  
}

2018年4月2日16:41:13

1. 使用okhttp的multipart/form-data方式上传一个文件，不带参数

之前已经写过通过浏览器上传文件到play服务器的实现过程，后台接收数据的过程可以参考我写的palyframework笔记。

后台主要用到了这两句代码：

MultipartFormData body = *request*().body().asMultipartFormData();

FilePart picture = body.getFile("myfile");

这里主要讲解用okhttp上传文件。客户端和后台都使用了表单的方式上传文件，都是基于multipart/form-data的，其实还有别的，但是没有实现成功。

Android端需要先浏览本地文件，然后通过文件路径生成一个File对象。然后将这个对象添加到MultipartBody的builder中，使用了builder设计模式。

接下来就和okhttp的普通请求类似了，创建一个request，里面携带了请求体，然后调用okhttp的newcall方法传递这个request，最后就是调用异步或者通过执行请求并处理返回结果了。

*/\*\*  
 \* 使用multipart/form-data上传一个文件,不带参数  
 \*/***public** <T> **void** upLoadFile2(String filePath) {  
 *//补全请求地址* String requestUrl = **urlAddress**+**"fileadd"**;  
 *//创建File* File file = **new** File(filePath);  
  
 MultipartBody.Builder builder = **new** MultipartBody.Builder()  
 .setType(MultipartBody.***FORM***)  
 .addFormDataPart(**"myfile"**,file.getName(),RequestBody.*create*(**MEDIA\_OBJECT\_STREAM**, file));  
 RequestBody body = builder.build();**final** Request request = **new** Request.Builder().url(requestUrl).post(body).build();  
 **final** Call call = **okHttpClient**.newBuilder().writeTimeout(50, TimeUnit.***SECONDS***).build().newCall(request);  
 call.enqueue(**new** Callback() {  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 Log.*e*(**"file"**, e.toString()+**"上传失败"**);  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 **if** (response.isSuccessful()) {  
 String string = response.body().string();  
 Log.*e*(**"file"**, **"response ----->"** + string);  
  
 Message msg= **myhandler**.obtainMessage();  
 Bundle bundle = **new** Bundle();  
 bundle.putString(**"responseMsg"**,string);  
 msg.setData(bundle);  
 msg.sendToTarget();  
  
 } **else** {  
 Log.*e*(**"file"**, **"上传失败2"**);  
 }  
 }  
 });  
}

1. 实现带参数的okhttp文件上传，带参数就是额外提交一个form表单

这里和前面非常相似，只需要在MultipartBody.Builder里面使用addFormDataPart方法添加键值对就可以了。

addFormDataPart方法有两个重载的方法：

1. MultipartBody.Builder addFormDataPart(String name, String value)

Add a form data part to the body.

1. MultipartBody.Builder addFormDataPart(String name, String filename, RequestBody body)

Add a form data part to the body.

前者直接添加一个键值对，后者就是天机文件的方式，MultiparBody类也是继承于RequestBody的。

修改后如下：

MultipartBody.Builder builder = **new** MultipartBody.Builder()  
 .setType(MultipartBody.***FORM***)  
 .addFormDataPart(**"name"**,**"guozhang"**) *//添加的参数* .addFormDataPart(**"myfile"**,file.getName(),RequestBody.*create*(**MEDIA\_OBJECT\_STREAM**, file));  
  
RequestBody body = builder.build();

2018年4月3日16:41:03

1. 使用okhttp还可以提交字符串string

okhttp的源码里面有很多参考代码，

C:\Users\Administrator\Desktop\okhttp-master\samples\guide\src\main\java\okhttp3\recipes

比如：PostString.java

public static final MediaType MEDIA\_TYPE\_MARKDOWN

= MediaType.parse("text/x-markdown; charset=utf-8");

String postBody = ""

+ "Releases\n"

+ "--------\n"

+ "\n"

+ " \* \_1.0\_ May 6, 2013\n"

+ " \* \_1.1\_ June 15, 2013\n"

+ " \* \_1.2\_ August 11, 2013\n";

Request request = new Request.Builder()

.url("https://api.github.com/markdown/raw")

.post(RequestBody.create(MEDIA\_TYPE\_MARKDOWN, postBody))

.build();

后台接收使用asText() 就可以了。

*request*().body().asText()

1. 网上找了个带进度的okhttp例子，它封装得很好，各种重载，而且进度的显示在子线程和ui之间处理得很好，用的是handler。然后还有下载的功能。

但是我用了它的例子，却一直无法上传文件，手机端可以显示上传完成，且能明显看到网速的变化。说明确实上传了数据，可是后却收不到数据，控制台显示空指针异常，具体也看不出来是哪里的问题。

今天仔细研究他的代码，最终发现了问题。在设置请求体的时候MultipartBody.Builder少设置了一个参数。

正常情况是下面这样的：

MultipartBody.Builder builder = **new** MultipartBody.Builder();  
  
  builder .setType(MultipartBody.FORM);

它却没有builder .setType(MultipartBody.FORM);

添加上后一切正常。

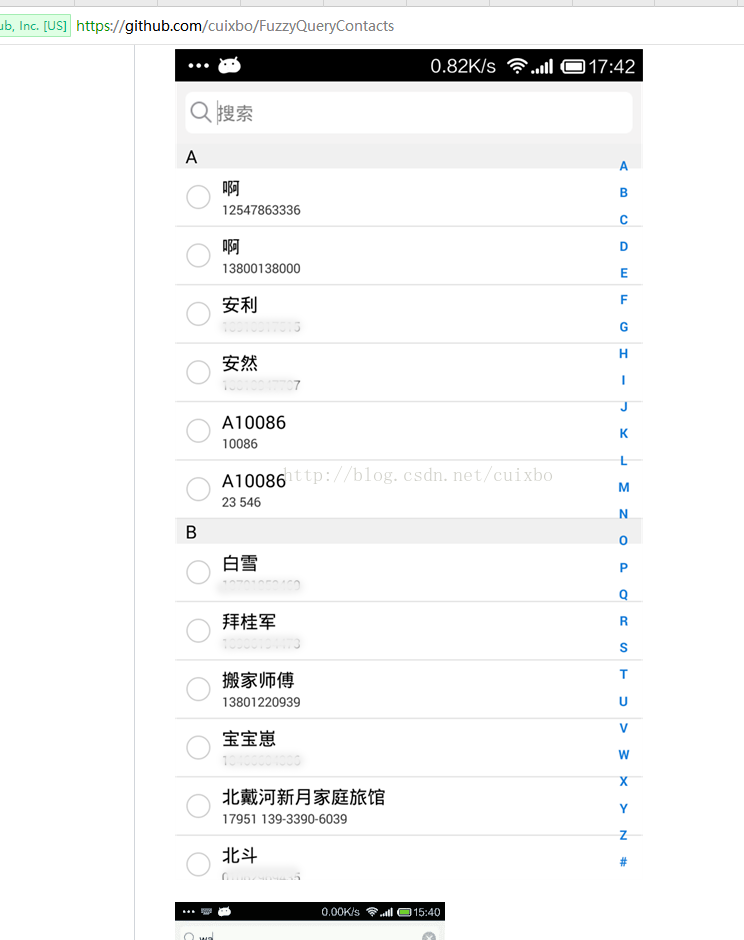
**private static** RequestBody getRequestBody(List<String> fileNames) {  
 *//创建MultipartBody.Builder，用于添加请求的数据* MultipartBody.Builder builder = **new** MultipartBody.Builder();  
  
 builder .setType(MultipartBody.***FORM***);  
  
 **for** (**int** i = 0; i < fileNames.size(); i++) { *//对文件进行遍历* File file = **new** File(fileNames.get(i)); *//生成文件  
 //根据文件的后缀名，获得文件类型* String fileType = *getMimeType*(file.getName());  
 builder.addFormDataPart( *//给Builder添加上传的文件* **"image"**, *//请求的名字* file.getName(), *//文件的文字，服务器端用来解析的* RequestBody.*create*(MediaType.*parse*(fileType), file) *//创建RequestBody，把上传的文件放入* );  
 }  
 **return** builder.build(); *//根据Builder创建请求*}

2018年10月17日20:36:27

1. 通讯录 contact 开发

今天要帮同学的同学做一个通讯录的安卓软件。突然想起以前做过，不过也是网上找的代码。可是怎么也找不到源码了，只好网上到处搜寻，找了好多例子都不满意。很多项目都根本无法运行。

最后还是在github上找到了以前用过的例子了，虽然他是用的eclipse开发的，但是移植还是比较容易。直接用android studio导入会有很多错误。我直接新建了一个项目，考入相关代码和资源，很快就能运行了。



<https://github.com/cuixbo/FuzzyQueryContacts>

2018年10月22日22:10:12

1. fastjson的使用

Json的工具有很多，谷歌的gson。但貌似在gradle里面很难下载jar包。

Springboot默认使用了jackson。

fastjson是阿里巴巴提供的工具。

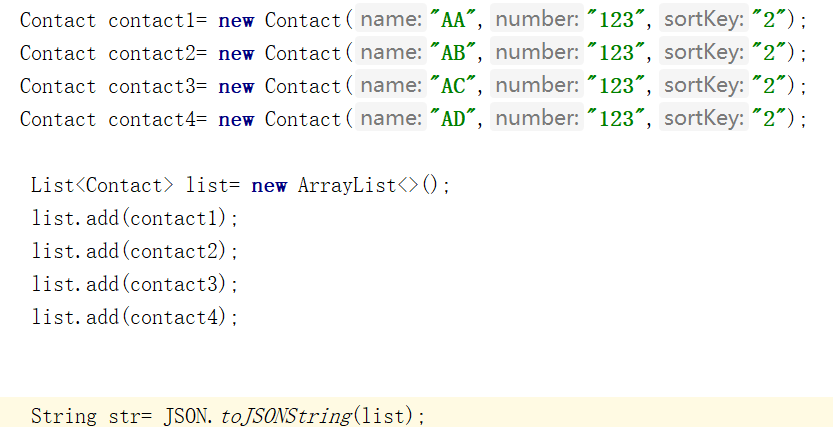
github的地址是：https://github.com/alibaba/fastjson

首先在gradle里面引用的 方法是：

compile **'com.alibaba:fastjson:1.2.49'**

1. 将对象转换成JSON字符串的方法：

String str= JSON.toJSONString(list);



1. 将string转换成json对象

JSONArray jsonArray =JSON.parseArray(str);

