

第三方插件(定位, 分享)、RecyclerView使用

RecyclerView的使用

1 引入recyclerView

在module中的build.gradle文件里, 找到dependencies, 添加关于recyclerView的引用:

```
compile 'com.android.support:recyclerview-v7:24.2.0'
```

其实design包中含有Material Design相关的很多控件, 其中也包括了recyclerView的代码, 所以你也可以通过引用design包来使用recyclerView:

```
compile 'com.android.support:design:23.2.0'
```

2 在xml中写一个recyclerView

在MainActivity中添加一个RecyclerView进行展示, 代码如下:

```
<!--省略不提...-->
<android.support.v7.widget.RecyclerView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/recyclerView"/>
<!--省略不提...-->
```

3 在java中给recyclerView设置数据

在MainActivity中的代码如下:

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    initView();
    initData();
}

private void initData() {
    ArrayList<String> datas = new ArrayList<>();
    for(char i = 'A'; i <= 'Z'; i++){
        datas.add(i+"");
    }
    mRecyclerView.setAdapter(new MyAdapter(getApplicationContext(),datas));
}

private void initView() {
    mRecyclerView = (RecyclerView) findViewById(R.id.recyclerView);
    //必须要指定展示的效果，设置一个LayoutManager，不然还是会一片空白
    LinearLayoutManager layoutManager = new
LinearLayoutManager(getApplicationContext());
    mRecyclerView.setLayoutManager(layoutManager);
}

```

在MyAdapter中的代码如下：

```

public class MyAdapter extends RecyclerView.Adapter {

    private ArrayList<String> datas;
    private Context mContext;

    public MyAdapter(Context applicationContext, ArrayList<String> datas) {
        this.datas = datas;
        this.mContext = applicationContext;
    }

    //RecyclerView的Adapter已经集成了ViewHolder
    //getView中的两段逻辑 1 创建view对象（将xml转为view）    2 找到view对象里面的一些控件，对其进行一些展示的设置

    //onCreateViewHolder 中执行第一段逻辑，创建出view，设置到一个ViewHolder中，再将holder返回
    @Override
    public RecyclerView.ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {
        //第三个参数为false，如果为true的话，生成的这个view其实就是第二个参数指向的parent，会造成报错的
        View view = LayoutInflater.from(mContext).inflate(R.layout.item_recyclerview,
parent,false);
        MyViewHolder holder = new MyViewHolder(view);
        return holder;
    }

    //给holder里的控件设置数据
    @Override
    public void onBindViewHolder(RecyclerView.ViewHolder holder, int position) {
        ((MyViewHolder)holder) .mTextView.setText(datas.get(position));
    }

    //Count 控制控件有多少个item
    @Override
    public int getItemCount() {
        return datas.size();
    }
}

class MyViewHolder extends RecyclerView.ViewHolder{

    public TextView mTextView;

    public MyViewHolder(View itemView) {
        super(itemView);
        mTextView = (TextView) itemView.findViewById(R.id.tv_item);
    }
}

```

4 给recyclerView设置不同的展示效果

设置菜单来提供多个点击事件。首先给res/menu/menu_main.xml修改为以下代码：

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="com.a520it.recyclerview.MainActivity">
    <item
        android:id="@+id/action_listview"
        android:orderInCategory="100"
        android:title="listView"
        app:showAsAction="never"/>
    <item
        android:id="@+id/action_gridview"
        android:orderInCategory="100"
        android:title="gridView"
        app:showAsAction="never"/>
    <item
        android:id="@+id/action_hor_gridview"
        android:orderInCategory="100"
        android:title="横向的gridView"
        app:showAsAction="never"/>
    <item
        android:id="@+id/action_stagger"
        android:orderInCategory="100"
        android:title="交错的瀑布流"
        app:showAsAction="never"/>
</menu>
```

然后在MainActivity中，修改菜单代码：

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();

    if (id == R.id.action_listview) {
        Toast.makeText(MainActivity.this, "listview", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_gridview) {
        GridLayoutManager gridLayoutManager = new
GridLayoutManager(getApplicationContext(), 3);
        mRecyclerView.setLayoutManager(gridLayoutManager);
        Toast.makeText(MainActivity.this, "gridview", Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_hor_gridview) {
        GridLayoutManager gridLayoutManager = new
GridLayoutManager(getApplicationContext(),
3, GridLayoutManager.HORIZONTAL, true);
        mRecyclerView.setLayoutManager(gridLayoutManager);
        Toast.makeText(MainActivity.this, "横向的gridview",
Toast.LENGTH_SHORT).show();
        return true;
    } else if (id == R.id.action_stagger) {
        StaggeredGridLayoutManager staggeredGridLayoutManager =
new StaggeredGridLayoutManager(3,
StaggeredGridLayoutManager.HORIZONTAL);
        mRecyclerView.setLayoutManager(staggeredGridLayoutManager);
        Toast.makeText(MainActivity.this, "交错的瀑布流", Toast.LENGTH_SHORT).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

其实就是设置了不同的LayoutManager给RecyclerView而已。

5 给RecyclerView添加点击效果

RecyclerView没有提供OnItemClickListener相关的item点击方法，需要自己去实现。这里是通过在Adapter中去实现的。在Adapter中先自定义一个OnItemClickListener，然后在Adapter中的onBindViewHolder方法中，给holder.itemView设置OnClickListener，并在OnClickListener的点击方法中，调用自定义的OnItemClickListener。代码如下：

```

//给holder里的控件设置数据
@Override
public void onBindViewHolder(final RecyclerView.ViewHolder holder, final int position) {
    ((MyViewHolder)holder) .mTextView.setText(datas.get(position));
    //给item添加一个点击
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            // Toast.makeText(mContext, "点击了item"+position,
            Toast.LENGTH_SHORT).show();
            mOnItemClickListener.onItemClick(position,holder.itemView);
        }
    });
}

public void setOnItemClickListener(OnItemClickListener onItemClickListener){
    this.mOnItemClickListener = onItemClickListener;
}

private OnItemClickListener mOnItemClickListener = null;

public interface OnItemClickListener{
    void onItemClick(int position,View itemView);
}

```

然后在MainActivity中设置自定义的OnItemClickListener即可。代码如下：

```

private void initData() {
    ....
    myAdapter.setOnItemClickListener(new MyAdapter.OnItemClickListener() {
        @Override
        public void onItemClick(int position, View itemView) {
            Toast.makeText(MainActivity.this, "position: "+position,
            Toast.LENGTH_SHORT).show();
        }
    });
}

```

6 给RecyclerView设置动画效果

给菜单再添加两个按钮，点击时就插入、删除数据。修改menu_main.xml代码如下：

```

<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context="com.a520it.recyclerview.MainActivity">
    ...
    <item
        android:id="@+id/action_add"
        android:orderInCategory="100"
        android:title="add"
        android:icon="@drawable/ic_menu_add"
        app:showAsAction="always"/>
    <item
        android:id="@+id/action_delete"
        android:orderInCategory="100"
        android:title="delete"
        android:icon="@drawable/ic_menu_delete"
        app:showAsAction="always"/>
</menu>

```

在MainActivity中设置onOptionsItemSelected方法代码如下：

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    ...else if(id == R.id.action_add) {
        mMyAdapter.addItem(1, "B");
        Toast.makeText(MainActivity.this, "add", Toast.LENGTH_SHORT).show();
        return true;
    }else if(id == R.id.action_delete) {
        mMyAdapter.deleteItem(3);
        Toast.makeText(MainActivity.this, "delete", Toast.LENGTH_SHORT).show();
        return true;
    }
    return super.onOptionsItemSelected(item);
}

```

插入、删除数据的代码写在MyAdapter里：

```

    public void addItem(int position,String str){
        datas.add(position,str);
        //        notifyDataSetChanged();
        notifyItemInserted(position);
    }

    public void deleteItem(int position){
        datas.remove(position);
        //        notifyDataSetChanged();
        notifyItemRemoved(position);
    }

```

7 给RecyclerView设置分割线

RecyclerView不像ListView，它没有已设置好的分割线，需要自己处理。给RecyclerView设置分割线需要执行以下代码：

```

mRecyclerView.addItemDecoration(new RecyclerView.ItemDecoration() {

    String str = "我是分割线";
    @Override
    public void onDraw(Canvas c, RecyclerView parent, RecyclerView.State state) {
        ....
        c.drawText(str,left,top,paint);
    }
}

    @Override
    public void getItemOffsets(Rect outRect, View view, RecyclerView parent,
RecyclerView.State state) {
        //通过outRect设置各个item之间的位移间距
        outRect.set(0,0,0,getStringHeight(paint));
    }
});

```

需要给RecyclerView设置一个ItemDecoration，如果不想自己去实现里面的方法，可以参考一些三方库，如：[recyclerview-flexibledivider][<https://github.com/yqritc/RecyclerView-FlexibleDivider>]

百度地图的使用

1 找百度申请app key

申请步骤如下：

1. 进入[百度开发者平台][<http://lbsyun.baidu.com/index.php?title=androidsdk>]，按照文档提示开始操作。
2. [创建应用][<http://lbsyun.baidu.com/apiconsole/key/create>]，注意选择应用类型为**Android SDK**。

3. 获得发布版和开发版的SHA1，这个SHA1是关于你的android应用签名文件的数字特征签名。这个SHA1可以通过命令行或者集成高版本ADT的eclipse来获得，这里使用的是命令行来获得。

1. 首先进入到应用签名文件的所在目录，我们现在运行项目生成apk时默认都是采用的一个名为debug.keystore的签名文件，一般它位于C:/user/.android。来到该目录后，在地址栏输入cmd，打开命令行窗口。

2. 在命令行窗口中，输入以下命令：

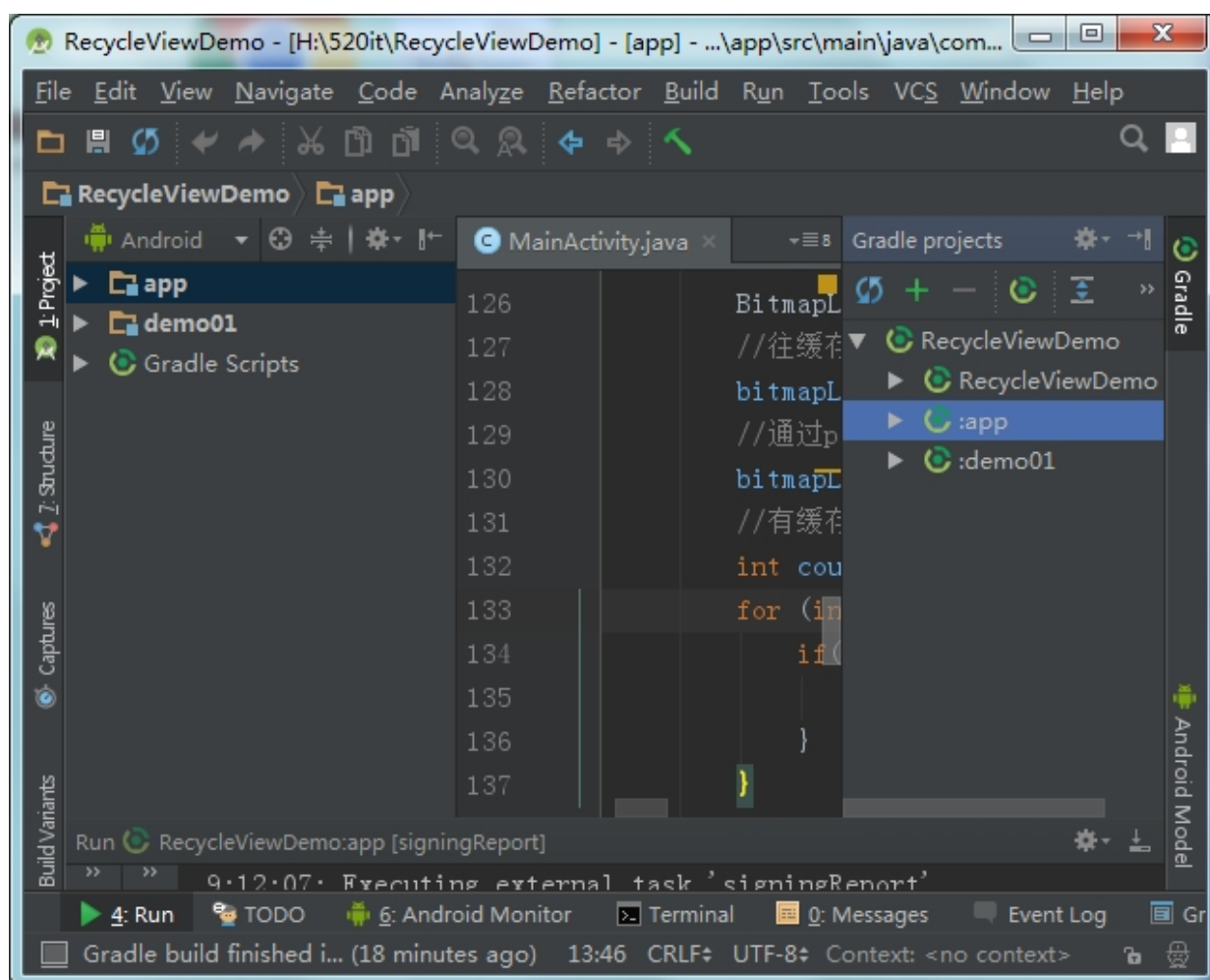
```
keytool -list -v -keystore debug.keystore
```

3. 它会提示你输入密钥，这个debug.keystore的密钥为：android

4. 然后将会展示出debug.keystore的一些信息，其中就包括我们所需要的SHA1。

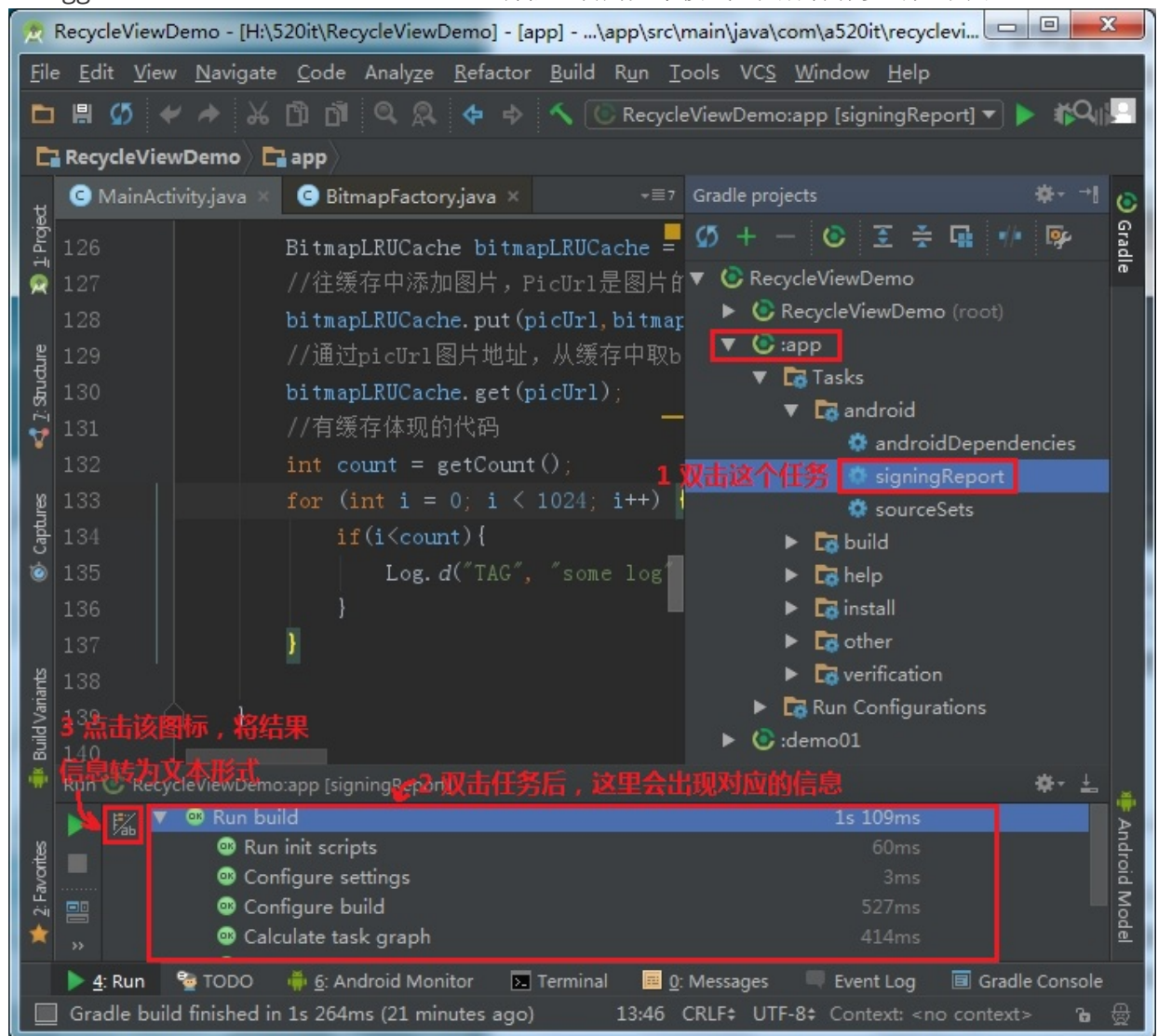
4. 将获得的SHA1进行输入，将应用的包名进行输入，然后点击“完成”，将会创建完毕，百度会创建一个appKey出来，这个appKey我们接下来就会用到。

5. 获得签名文件的SHA1数字特征还有其他的办法。如果使用的是Android Studio，在项目打开时，右边会有个收起来的视窗「Gradle」。

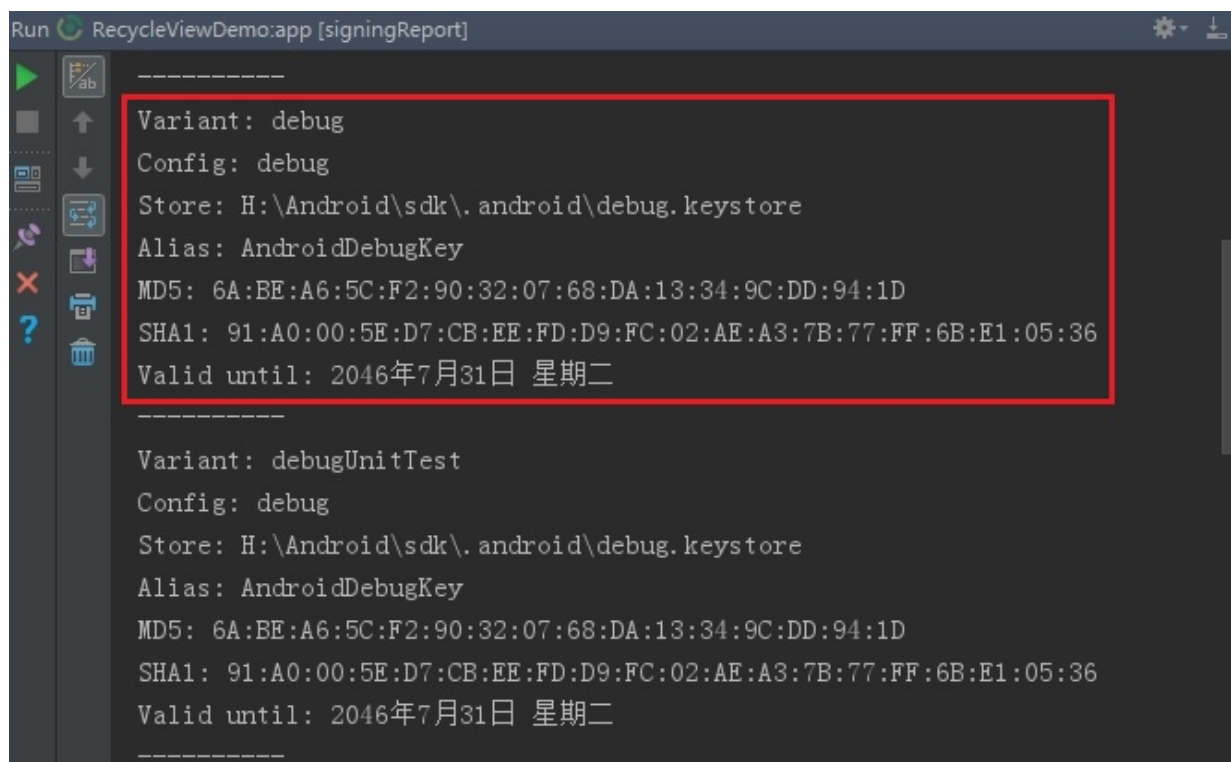


通过点击「Gradle」，然后来到你这个应用moudle下，找到如下目录文件，并对其双击（这里以名为app的moudle为例）。下面将会出现对应的结果信息，最后点击左侧的按钮

「Toggle tasks executions/text mode」，将信息转为文本模式。具体操作步骤如下图：

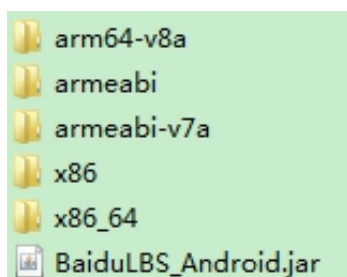


这时，下方的窗体就会呈现出签名文件的各种文字信息了：



2 在项目中引入百度地图SDK

SDK资源可以自行去[百度开发者官网][<http://lbsyun.baidu.com/index.php?title=androidsdk/sdkandev-download>]下载。拿到资源后，引入到Android Studio项目中来。SDK的文件目录如下：



里面有一些jni相关的so文件夹以及一个jar包。先将那个jar文件放到module中的libs目录，然后在module中的src目录中，创建一个jniLibs文件夹，

3 开始使用百度地图

先在AndroidManifest.xml中的application节点中配置信息：

```
<application
    .....>
    <meta-data
        android:name="com.baidu.lbsapi.API_KEY"
        android:value="开发者key" />
    .....
</application>
```

这个开发者key就是先前你申请的应用的appKey。这里我申请的key为：

```
6vIXDy1Rf7I6ULLEvMVgSZWQBosvZiy9    //应用的包名为: com.a520it.baidumap
```

使用SHA1会报错。错误log如下：

```
Authentication Error errorcode: 230 uid: -1 appid -1 msg: APP Scode码校验失败
```

如果报错，注意验证，可以通过百度提供的应用来检查。

在模拟器上使用MapView，会报错，而真机则不会。报错如下：

```
E/AndroidRuntime: FATAL EXCEPTION: GLThread 199
    java.lang.IllegalArgumentException: No config chosen
        at android.opengl.GLSurfaceView$BaseConfigChooser.chooseConfig(GLSurfaceView.java:874)
        at android.opengl.GLSurfaceView$EglHelper.start(GLSurfaceView.java:1024)
        at android.opengl.GLSurfaceView$GLThread.guardedRun(GLSurfaceView.java:1401)
        at android.opengl.GLSurfaceView$GLThread.run(GLSurfaceView.java:1240)
```

报错的原因：MapView需要使用一些openGL（图形程序接口）的功能，而模拟器支持openGL有些问题。想要不报错，将所有MapView控件都改为TextureMapView即可。

4 百度地图的定位与标注

定位当前所处的位置，并用一个小图标展示一下。

可以通过坐标拾取系统来获得坐标，比如天安门，可以得到坐标为：



当前坐标点如下：

116.403958,39.915049



获得的坐标在设置给java代码时，要交换，比如：

```
private void initOverlay() {  
    //定义Maker坐标点  
    //longitude 经度 latitude 纬度 这里传参数是（纬度，经度）  
    LatLng point = new LatLng(39.915049,116.403958);  
    //构建Marker图标  
    BitmapDescriptor bitmap = BitmapDescriptorFactory  
        .fromResource(R.drawable.icon_gcoding);  
    //构建MarkerOption，用于在地图上添加Marker  
    OverlayOptions option = new MarkerOptions()  
        .position(point)  
        .icon(bitmap);  
    //在地图上添加Marker，并显示  
    mBaiduMap.addOverlay(option);  
}
```

定位需要在清单文件中的Application节点中配置：

```
<service android:name="com.baidu.location.f" android:enabled="true"  
    android:process=":remote"></service>
```

别忘了清单文件中还要声明对应的定位权限，请注意检查。

定位的详细代码可以参考官方文档。以下代码为参考：

```

public LocationClient mLocationClient = null;
public BDLocationListener myListener = new MyLocationListener();

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    mLocationClient = new LocationClient(getApplicationContext());    //声明
    LocationClient类
    mLocationClient.registerLocationListener(myListener);    //注册监听函数
    initLocation();
    mLocationClient.start();
}

private void initLocation() {
    LocationClientOption option = new LocationClientOption();
    option.setLocationMode(LocationClientOption.LocationMode.Hight_Accuracy
);//可选，默认高精度，设置定位模式，高精度，低功耗，仅设备
    option.setCoorType("bd0911");//可选，默认gcj02，设置返回的定位结果坐标系
    int span = 1000;
    option.setScanSpan(span);//可选，默认0，即仅定位一次，设置发起定位请求的间隔需要大于等于1000ms才是有效的
    option.setIsNeedAddress(true);//可选，设置是否需要地址信息，默认不需要
    option.setOpenGps(true);//可选，默认false，设置是否使用gps
    option.setLocationNotify(true);//可选，默认false，设置是否当gps有效时按照1S1次频率输出GPS结果
    option.setIsNeedLocationDescribe(true);//可选，默认false，设置是否需要位置语义化结果，可以在BDLocation.getLocationDescribe里得到，结果类似于“在北京天安门附近”
    option.setIsNeedLocationPoiList(true);//可选，默认false，设置是否需要POI结果，可以在BDLocation.getPoiList里得到
    option.setIgnoreKillProcess(false);//可选，默认true，定位SDK内部是一个SERVICE，并放到了独立进程，设置是否在stop的时候杀死这个进程，默认不杀死
    option.SetIgnoreCacheException(false);//可选，默认false，设置是否收集CRASH信息，默认收集
    option.setEnableSimulateGps(false);//可选，默认false，设置是否需要过滤gps仿真结果，默认需要
    mLocationClient.setLocOption(option);
    mBaiduMap.setMyLocationEnabled(true);
}

boolean mIsFirstLoc = true;

public class MyLocationListener implements BDLocationListener {

    @Override
    public void onReceiveLocation(BDLocation location) {
        Log.e("BaiduLocationApiDem", "接收到信号");
        if (location == null || mMapView == null) {
            return;
        }
        MyLocationData locData = new MyLocationData.Builder()

```



```

        .accuracy(location.getRadius())
        // 此处设置开发者获取到的方向信息，顺时针0-360
        .direction(100).latitude(location.getLatitude())
        .longitude(location.getLongitude()).build();
mBaiduMap.setMyLocationData(locData);
if (mIsFirstLoc) {
    mIsFirstLoc = false;
    LatLng ll = new LatLng(location.getLatitude(),
        location.getLongitude());
    MapStatus.Builder builder = new MapStatus.Builder();
    builder.target(ll).zoom(18.0f);

mBaiduMap.animateMapStatus(MapStatusUpdateFactory.newMapStatus(builder.build()));
    }
}

@Override
protected void onDestroy() {
    super.onDestroy();
    // 当不需要定位图层时关闭定位图层
    mBaiduMap.setMyLocationEnabled(false);
}

```

5 手机定位方式介绍

手机定位主要分为这三种：GPS定位、基站定位、WIFI定位、AGPS定位。

5.1 GPS定位

GPS定位的特点是：

- 定位准，无需网络
- 耗电
- 用户默认关闭GPS
- 室内几乎无效

GPS定位的优势是精确，只要能接收到四颗卫星的定位信号，就可以进行误差在5-10米以内的定位。而GPS定位由于接收机任何时刻都至少被4颗卫星覆盖，所以信号得到了很好的保证，并且由于卫星居高临下，排除卫星钟及大气干扰等因素，精度也能保证在几米至几十米。缺点是GPS受天气和位置的影响较大。当遇到天气不佳的时候、或者处于高架桥/树荫的下面，或者在高楼的旁边角落、地下车库或露天的下层车库（或者简单地说当见不到天空的时候），GPS的定位就会受到相当大的影响，甚至无法进行定位服务。

5.2 基站定位

基站定位的特点如下：

- 定位精度低
- 使用范围窄

- 需要网络
- 室内有效

只要计算三个基站的信号差异,就可以判断出手机所在的位置。因此,只要用户手机处于移动通信网络的有效范围之内,就可以随时进行位置定位,而不受天气、高楼、位置等等的影响。

LBS虽然不会受到天气、高架桥或高楼的影响,但如果超出手机的服务范围,或者手机所处的基站数量不足,则无法进行定位。所以基站定位不太适合野外使用,由于基于现有手机通信基站,受环境影响较大,由于基站稀少,在郊区和农村移动台定位在1000~2000米范围内;在城区定位范围为100~200米,而在无法接收到手机信号的地方,就谈不上定位了。

5.3 wifi定位

wifi定位的特点:

- 定位精度低
- 使用范围窄
- 需要wifi
- wifi位置服务商

定位原理与基站定位类似。每一个无线热点AP (Access Point) 都有一个全球唯一的MAC地址,并且一般来说无线热点在一段时间内是不会移动的。

设备在开启Wi-Fi的情况下,即可扫描并收集周围的AP信号,无论是否加密,是否已连接,甚至信号强度不足以显示在无线信号列表中,都可以获取到AP广播出来的MAC地址。

设备将这些能够标示AP的数据发送到位置服务器,服务器检索出每一个AP的地理位置,并结合每个信号的强弱程度,计算出设备的地理位置并返回到用户设备。

位置服务商要不断更新、补充自己的数据库,以保证数据的准确性,毕竟无线AP不像基站塔那样基本100%不会移动。位置服务商现在来说只有Skyhook和Google两家,收集数据的方式:

- 主动采集: Google的街景拍摄车还有一个重要的功能就是采集沿途的无线信号,并打上通过GPS定位出的坐标回传;
- 用户提交: Android手机用户在开启“使用无线网络定位”时会提示是否允许Google的定位服务手机匿名地点数据。

5.4 AGPS定位

AGPS (AssistedGPS, A-GPS, 网络辅助GPS) 定位技术结合了 **GPS定位** 和 **基站定位** 的优势,借助蜂窝网络的数据传输功能,可以达到很高的定位精度和很快的定位速度,在移动设备尤其是手机中被越来越广泛的使用。

传统GPS的冷启动TTFF(第一次定位时间)大约在1分钟以上。通过网络提供卫星数据(卫星在哪里,频移等),可以降低到几秒。

在传统GPS定位中需要全频段搜索以找到可用卫星因而耗时较长,而AGPS通过网络直接下载当前地区的可用卫星信息,从而提高了搜星速度。同时,也减小了设备的电量消耗。

第三方分享

有很多应用需要分享数据到其他平台上，比如将你公司的链接分享到微信、QQ、微博上。像微信、微博等大公司都有提供分享SDK出来供开发者去集成和调用。但是如果一家接一家的去集成，会很累很麻烦，市面上有些开发者平台将这些SDK都合并到了一块，做成了统一的分享SDK。这些开发者平台有：友盟、[Mob][mob.com]等。这里以mob分享为例，建议大家查看mob的[快速集成官方提示][http://bbs.mob.com/thread-22130-1-1.html]。

1 获取Appkey

如果你尚未在mob注册开发者账号，需要先注册，注册之后登录你的账号，点击添加新应用，填写完应用基本信息后，将进入"下载SDK并添加代码"页面，此页面即可得到Appkey。这里我获得的AppKey为：

```
sharesdk的appkey:      1758971896a38
```

2 下载并引入SDK

注册后，开始下载ShareSDK，进入[下载页面](#)，选择Android后，进行SDK下载。

下载后根据需要生成资源代码，然后导入到项目中。

3 去各大平台注册应用信息

需要去各大平台（如：微信、微博等）去注册你的应用信息，不然没法分享。注册可以参考[这些步骤][http://bbs.mob.com/forum.php?mod=viewthread&tid=275&page=1&extra=#pid860]。

以下是我注册各大平台后得到的信息：

新浪微博

App Key: 854093674 App Secret: 7db36179eb248cf73f4184172e708e89

QQ（除了改asset中文件，那个AndroidManifest中QQ分享相关Activity也要记得改下id）

APP ID: 1105711356 APP KEY: SbP8vAzHCfhTrLgL

微信

待审核

4 分享

[http://bbs.mob.com/thread-22130-1-1.html][http://bbs.mob.com/thread-22130-1-1.html]: