高德地图

# 准备

## 简介

高德是中国领先的数字地图内容、导航和位置服务解决方案提供商。公司2010年登陆美国纳斯达克全球精选市场（NasdaqAMAP）。高德拥有导航电子地图甲级测绘资质、测绘航空摄影甲级资质和互联网地图服务甲级测绘资质“三甲”资质，其优质的电子地图数据库成为公司的核心竞争力。



## 注册

<http://lbs.amap.com/>



## 下载sdk



<http://lbs.amap.com/api/android-sdk/download/>

## 把demo导入AS

一般需要修改build.gradle文件修改buildToolsVersion

## 添加应用









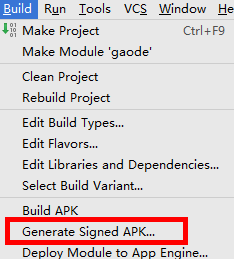
## SHA1安全号码的获取

[http://lbs.amap.com/dev/ticket#/faq/249](http://lbs.amap.com/dev/ticket" \l "/faq/249)

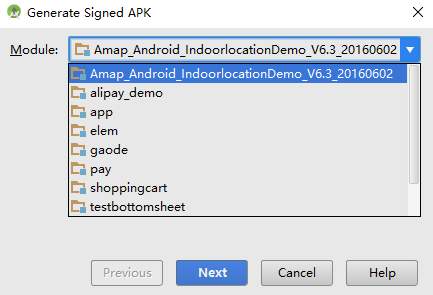
官方的文档在测试的时候存在一些问题。

**使用的是Android Studio开发工具**

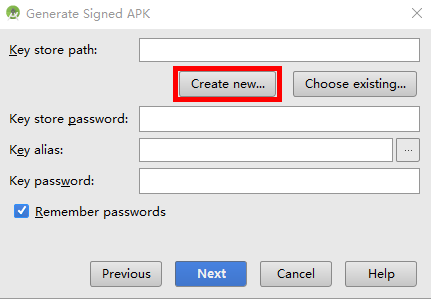
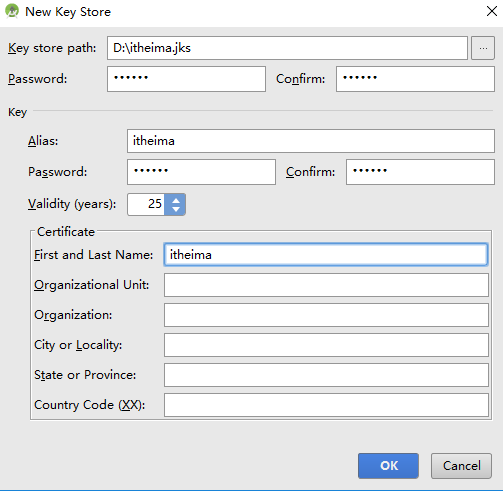
### 产生debug.keystore

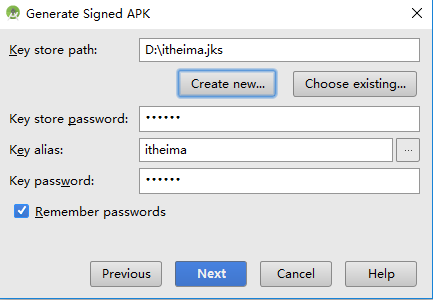


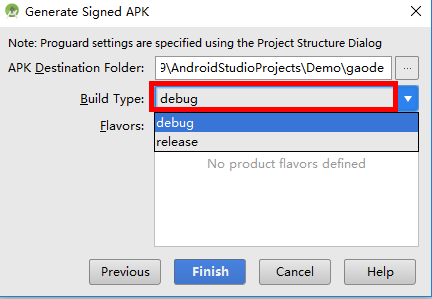
选择要签名的Module



新建keyStore





点击Finish生成debug的签名

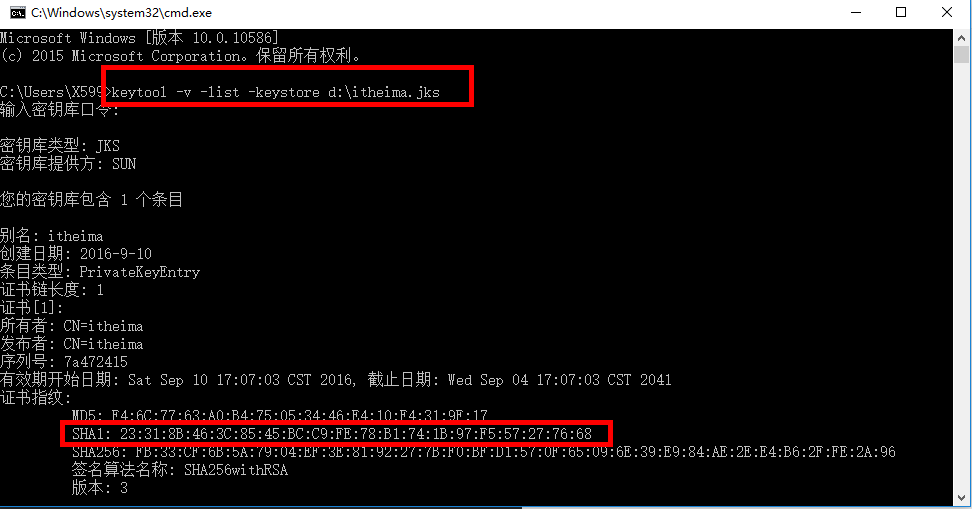
### 生成SHA1安全码

第一步：打开cmd窗口debug F2:D6:77:17:90:69:97:43:7C:BE:72:69:D4:A4:33:37:D8:EF:E1:E6

E8:D9:AA:9F:99:2D:8D:BC:89:7A:AD:2E:32:7D:87:79:57:CA:94:B0

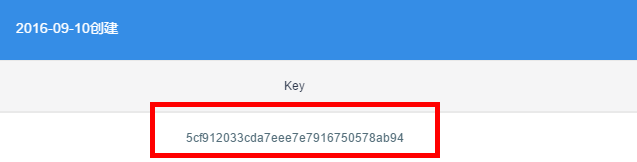
第二步：输入命令：keytool -v -list -keystore keystore文件路径 回车

第三步：输入Keystore密码



## 添加应用信息

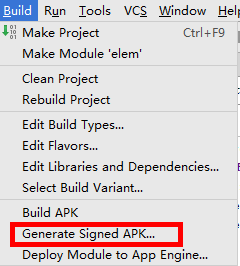


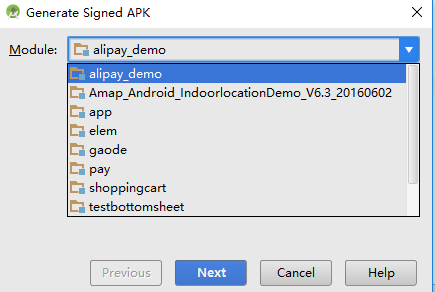


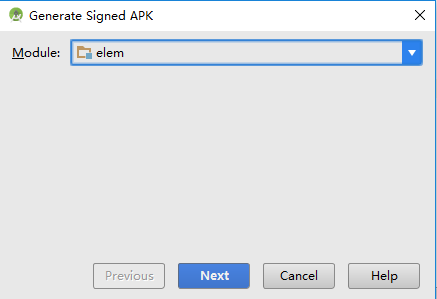
## 修改demo的清单文件

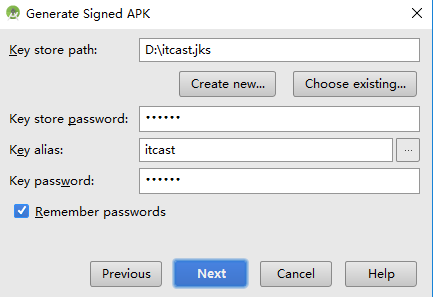


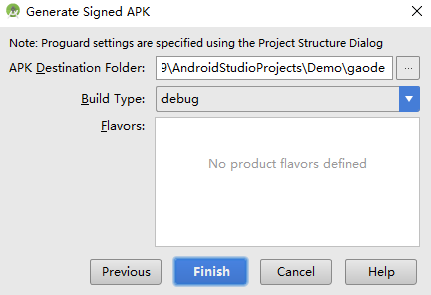
## 生成Debug签名的apk

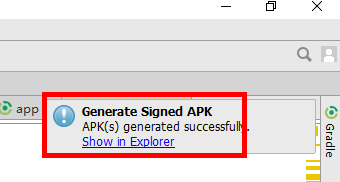


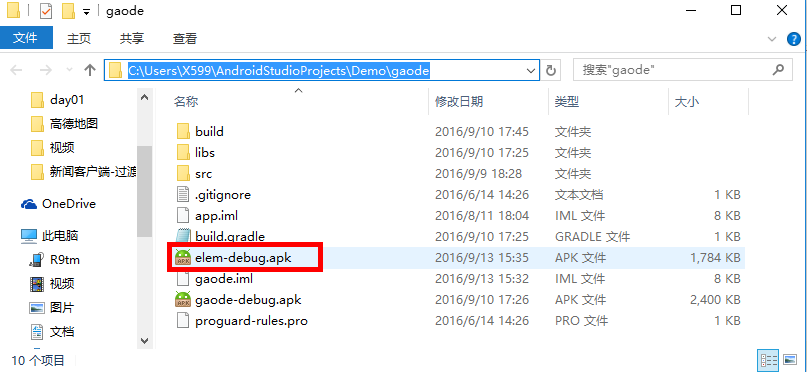






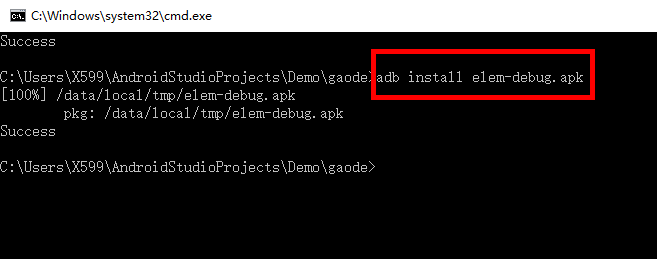






## 运行app 安装apk安装到真机上

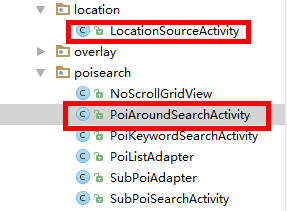
注意：模拟器是不支持定位功能的。



# 通过需求锁定需要查看的模块

定位模块

搜索附近



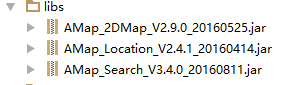
大致看懂流程，复制代码

# 手动编写测试案例

* 创建工程

包名和demo的报名保持一致

* 复制jar包

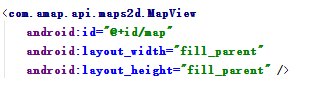


* 修改清单文件

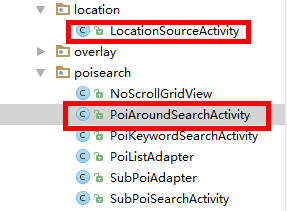


* 复制核心代码

布局



定位|搜索



注意：定位onLocationChanged()方法会不断的调用，只要定位成功一次就可以停止定位。

# 整合项目

## 新建收货地址，选择地址



修改EditText为TextView 修改maxLines为2

RequestCode == 200 ResultCode == 100

回传数据

Location 地址(Title)

Snippet

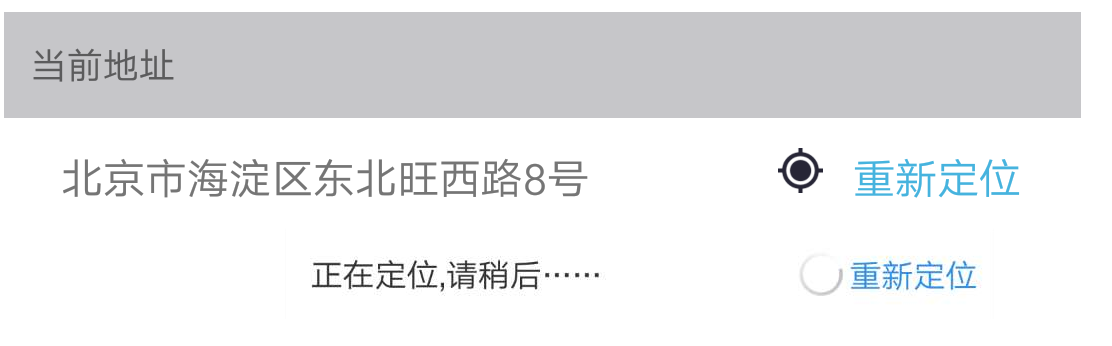
## 回送数据



点击是为了获取新的地址信息。StartActivityForResult().

Activity里面的onActivityResulut()方法 调用Fragment里面 onActivityResult()方法

## 定位



文字和图片状态的改变

添加id **tv\_location\_address**

<http://lbs.amap.com/api/android-location-sdk/locationsummary/>

通过代码直接定位

@Override  
**public void** onLocationChanged(AMapLocation aMapLocation) {  
 **this**.**aMapLocation** = aMapLocation;  
 *//停止定位* **mlocationClient**.onDestroy();  
 System.***out***.println(aMapLocation.getAddress() + **","** + aMapLocation.getLocationDetail());  
 **lp** = **new** LatLonPoint(aMapLocation.getLatitude(),aMapLocation.getLongitude());  
 **city** = aMapLocation.getCity();  
 System.***out***.println(**"city:"**+**city**);  
 **tvLocationAddress**.setText(aMapLocation.getAddress());  
 **ivLocation**.setVisibility(View.***VISIBLE***);  
 **pbLocation**.setVisibility(View.***GONE***);  
 doSearchQuery();  
}

## 定位条目的添加



## 更多地址



Intent intent = **new** Intent(getApplicationContext(),SelectReceiptAddressActivity.**class**);  
intent.putExtra(**"lp"**,**lp**);  
startActivityForResult(intent,200);

绘制Mark

**lp** = getIntent().getParcelableExtra(**"lp"**);  
**if**(**lp** == **null**){  
 **aMap**.setLocationSource(**this**);*// 设置定位监听*}**else**{  
 LatLng latlngLocation = **new** LatLng(**lp**.getLatitude(),**lp**.getLongitude());  
 Marker markerBuyer = **aMap**.addMarker(**new** MarkerOptions()  
 .anchor(0.5f, 1.0f)  
 .position(latlngLocation)  
 .draggable(**true**));  
  
 ImageView markerBuyerIcon=**new** ImageView(**this**);  
 markerBuyerIcon.setImageResource(R.drawable.***location\_marker***);  
 markerBuyer.setIcon(BitmapDescriptorFactory.*fromView*(markerBuyerIcon));  
  
 *// 设置地图中心点* **aMap**.moveCamera(CameraUpdateFactory.*changeLatLng*(latlngLocation));

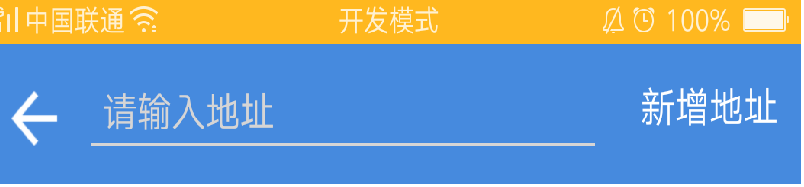
## 点击条目返回数据



view.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Intent intent = **new** Intent();  
 intent.putExtra(**"location"**, receiptAddressBean.getReceiptAddress());  
 setResult(100, intent);  
 finish();  
 }  
});

@Override  
**protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
 *//获取更多的地址* **if**(requestCode == 200 && resultCode == 100 && data != **null**){  
 setResult(100,data);  
 finish();  
 }  
}

## 新增地址



登录进入新增地址界面

未登录进入登录界面

**case** R.id.***tv\_add\_address***:  
 **if** (**application**.**USERID** != -1) {  
 Intent intent = **new** Intent(**this**, AddEditReceiptAddressActivity.**class**);  
 intent.putExtra(**"lp"**,**lp**);  
 startActivityForResult(intent,200);  
 } **else** {  
 startActivity(**new** Intent(**this**, LoginActivity.**class**));  
 }  
 **break**;

**case** R.id.***et\_receipt\_address***:  
 Intent intent = **new** Intent(**this**,SelectReceiptAddressActivity.**class**);  
 intent.putExtra(**"lp"**,**lp**);  
 startActivityForResult(intent,200);  
 **break**;

## 关键字搜索

**protected void** doSearchQuery(String keyWord) {  
 **currentPage** = 0;  
 **query** = **new** PoiSearch.Query(keyWord, **""**, **city**);

**query**.setPageSize(20);*// 设置每页最多返回多少条poiitem* **query**.setPageNum(**currentPage**);*// 设置查第一页* **if** (**lp** != **null**) {  
 **poiSearch** = **new** PoiSearch(**this**, **query**);  
 **poiSearch**.setOnPoiSearchListener(**this**);  
 **if**(TextUtils.*isEmpty*(keyWord)){  
 **poiSearch**.setBound(**new** PoiSearch.SearchBound(**lp**, 500, **true**));*//  
 // 设置搜索区域为以lp点为圆心，其周围5000米范围* }**else**{  
 **query**.setCityLimit(**true**);  
 }  
 **poiSearch**.searchPOIAsyn();*// 异步搜索* }  
}

@Override  
**public void** onPoiSearched(PoiResult result, **int** i) {  
 **if** (i == 1000) {  
 **if** (result != **null** && result.getQuery() != **null**) {*// 搜索poi的结果* **if** (result.getQuery().equals(**query**)) {*// 是否是同一条* **poiResult** = result;  
 **poiItems** = **poiResult**.getPois();*// 取得第一页的poiitem数据，页数从数字0开始* System.***out***.println(**"成功的获取到周围的数据"**+**poiItems**.size());  
 **if**(**poiItems**.size() != 0){  
 **if**(TextUtils.*isEmpty*(**etEnterAddress**.getText().toString())){  
 createNearByInfo(**poiItems**);  
 }**else**{  
 *//显示搜索信息* SelectReceiptAddressAdapter adapter = **new** SelectReceiptAddressAdapter(**this**,**poiItems**,**true**);  
 **rvSearchAddress**.setAdapter(adapter);  
 }  
  
 }  
 }  
 }   
 }  
}