# Day03 支付与地图

**支付前页面跳转**

1.点餐达到指定价格后显示结算按钮

if (countPrice > Float.parseFloat(mSeller.getSendPrice())) {  
 //显示下单按钮  
 mTvSubmit.setVisibility(VISIBLE);  
 mTvSendPrice.setVisibility(GONE);  
} else {  
 mTvSubmit.setVisibility(GONE);  
 mTvSendPrice.setVisibility(VISIBLE);  
}

2.确认订单页面

private void processIntent() {  
 if(getIntent()!=null){  
 mSeller = (Seller) getIntent().getSerializableExtra("seller");  
 mTvSellerName.setText(mSeller.getName());  
 mTvDeliveryFee.setText(PriceFormater.format(Float.parseFloat(mSeller.getDeliveryFee())));

3.在线支付线面

private void fillGoods() {  
 //动态添加所有商品条目view  
 if (mCartList != null && mCartList.size() > 0) {  
 for (GoodsInfo goodsInfo : mCartList) {  
 TextView textView = new TextView(this);  
 textView.setText(goodsInfo.getName() + " ￥" + goodsInfo.getNewPrice() + " \* " + goodsInfo.getCount());  
 mLlGoods.addView(textView);  
 }  
 }  
}

@OnClick({R.id.ib\_back, R.id.iv\_triangle, R.id.bt\_confirm\_pay})  
public void onClick(View view) {  
 switch (view.getId()) {  
 case R.id.ib\_back:  
 finish();  
 break;  
 case R.id.iv\_triangle:  
 //点击三角形展开订单详情，ll\_order\_detail  
 if (isOpened) {  
 //关闭它  
 mLlOrderDetail.setVisibility(View.GONE);  
 isOpened = false;  
 } else {  
 //展开它  
 mLlOrderDetail.setVisibility(View.VISIBLE);  
 isOpened = true;  
 }

}

**支付宝支付**

1. 支付宝使用须知
   1. 卖家必须要上传法人代表身份证和营业执照
   2. 商务美眉柜台或者网上签约
   3. 签约成功后拿到id、卖家账号、公钥、私钥
2. 支付宝Demo代码分析
   1. 构建订单信息
   2. 加密并签名
   3. 提交支付并回调

3.移植支付宝到外卖

//切换到总部商户PID  
public static final String PARTNER = "2088011085074233";  
//商户收款账号  
public static final String SELLER = "917356107@qq.com";  
//商户私钥，pkcs8格式  
public static final String PRIVATE\_KEY = "";  
//支付宝公钥  
public static final String RSA\_PUBLIC = "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC+dY2sUIcVvT51qVvp8+sie+rafXhhV5nxp7sTCvj5sf6mr3bHsFziwXyNjkgkd/U88cDGLwplK567dH1Umu5m5WzaLNhZzDPYo58fG7AAIZvouwSobmhzLpySh+vS/g++d8x9z9WCIdJcuQ5wK91j6h8nad/OKgb2/DUvU2stBwIDAQAB";

**高德地图配置**

1.添加地址时地图定位按钮

public void onClick(View view) {  
 switch (view.getId()) {  
 case R.id.btn\_map\_location:  
 //TODO:地图选择地址  
 Intent intent = new Intent(this, MapLocationActivity.class);  
 startActivityForResult(intent, REQUEST\_SELECT\_ADDRESS);  
 break;

2.导入高德地图demo（key不正确）

### 3.地图控制台添加应用









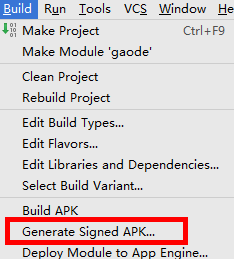
### 4.SHA1安全号码的获取

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

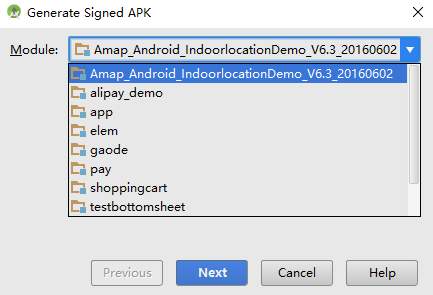
官方的文档在测试的时候存在一些问题。

**使用的是Android Studio开发工具**

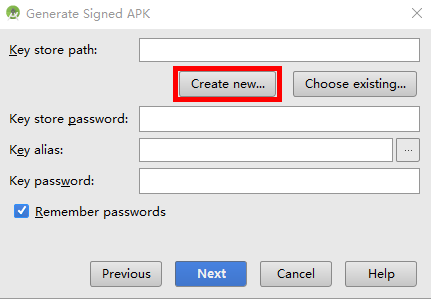
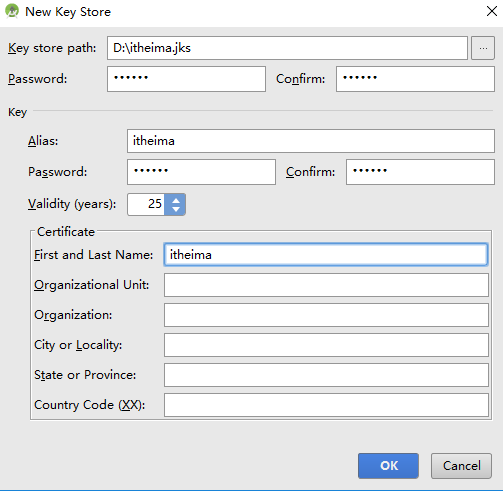
### 5.产生debug.keystore

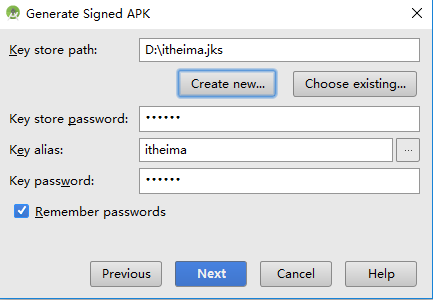


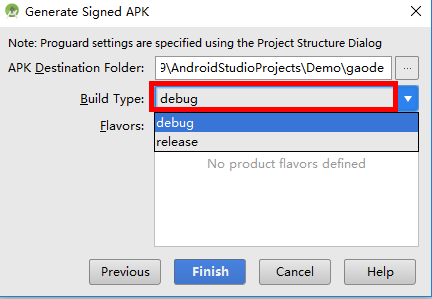
选择要签名的Module



新建keyStore





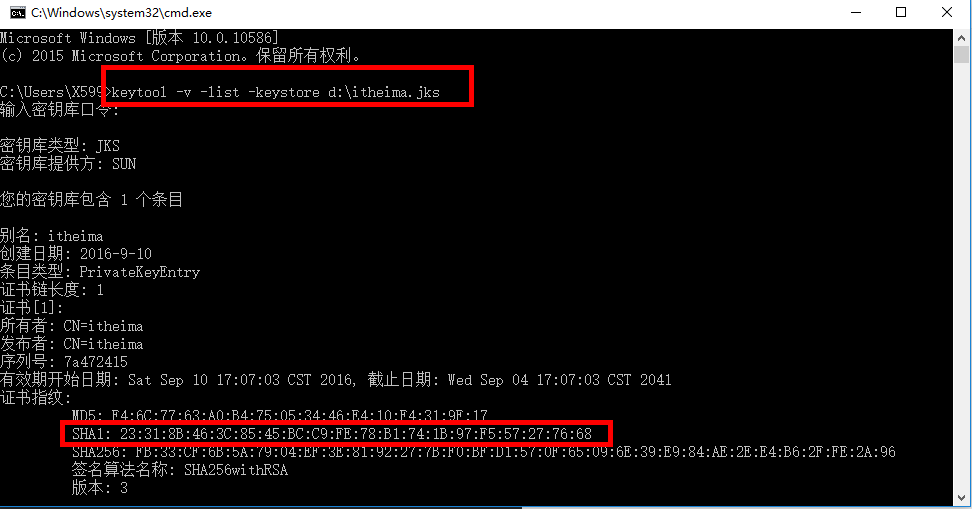
点击Finish生成debug的签名

### 6.生成SHA1安全码

第一步：打开cmd窗口debug 4D:9B:60:44:66:97:02:CB:F3:EA:37:60:87:B9:46:0C:9D:4E:03:80

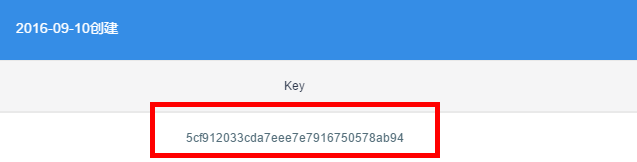
52:AF:D0:30:B1:82:E8:89:90:32:51:DB:94:55:28:52:A4:F9:F4:CA  
第二步：输入命令：keytool -v -list -keystore keystore文件路径 回车

第三步：输入Keystore密码



### 7.添加应用信息

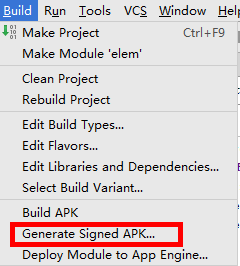


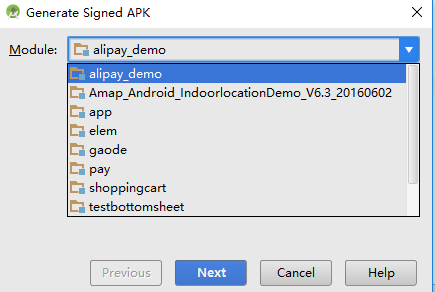


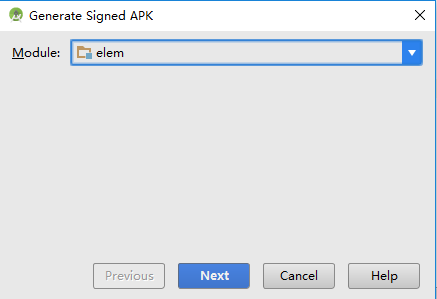
### 8.修改demo的清单文件

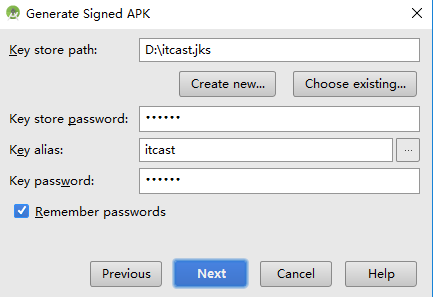


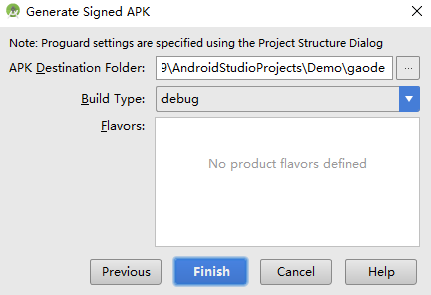
### 9.生成Release签名的apk

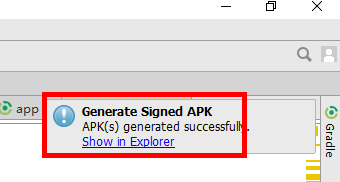


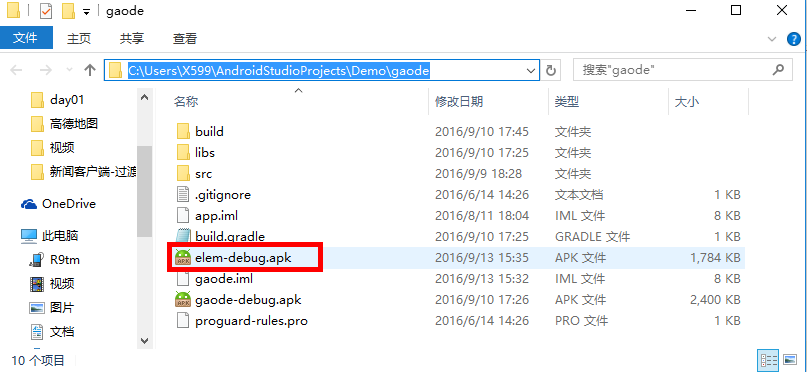






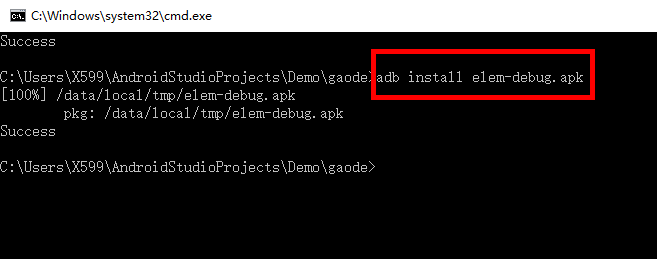






### 10.运行app 安装apk安装到真机上

注意：模拟器是不支持定位功能的。



**地图API**

1.地图view与地图控制器

private MapView mapView;  
private AMap aMap;  
  
@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.basicmap\_activity);  
 mapView = (MapView) findViewById(R.id.map);  
 mapView.onCreate(savedInstanceState);// 此方法必须重写  
   
 init();  
}  
  
/\*\*  
 \* 初始化AMap对象  
 \*/  
private void init() {  
 if (aMap == null) {  
 aMap = mapView.getMap();  
   
 }  
}

2.地图定位（LocationClient与OnLocationCahngeListener）

@Override  
public void activate(OnLocationChangedListener listener) {  
 mListener = listener;  
 if (mlocationClient == null) {  
 mlocationClient = new AMapLocationClient(this);  
 mLocationOption = new AMapLocationClientOption();  
 //设置定位监听  
 mlocationClient.setLocationListener(this);  
 //设置为高精度定位模式  
 mLocationOption.setLocationMode(AMapLocationMode.Hight\_Accuracy);  
 //设置定位参数  
 mlocationClient.setLocationOption(mLocationOption);  
 // 此方法为每隔固定时间会发起一次定位请求，为了减少电量消耗或网络流量消耗，  
 // 注意设置合适的定位时间的间隔（最小间隔支持为2000ms），并且在合适时间调用stopLocation()方法来取消定位请求  
 // 在定位结束后，在合适的生命周期调用onDestroy()方法  
 // 在单次定位情况下，定位无论成功与否，都无需调用stopLocation()方法移除请求，定位sdk内部会移除  
 mlocationClient.startLocation();  
 }  
}

@Override  
public void onLocationChanged(AMapLocation amapLocation) {  
 if (mListener != null && amapLocation != null) {  
 if (amapLocation != null  
 && amapLocation.getErrorCode() == 0) {  
 mListener.onLocationChanged(amapLocation);// 显示系统小蓝点  
 } else {  
 String errText = "定位失败," + amapLocation.getErrorCode()+ ": " + amapLocation.getErrorInfo();  
 Log.e("AmapErr",errText);  
 }  
 }  
}

3.切换地图的当前位置与放大级别

mAMap.moveCamera(CameraUpdateFactory.newLatLngZoom(new LatLng(lp.getLatitude(), lp.getLongitude()), 14));

4.周边搜索

protected void doSearchQuery() {  
 currentPage = 0;  
 query = new PoiSearch.Query(keyWord, "", city);// 第一个参数表示搜索字符串，第二个参数表示poi搜索类型，第三个参数表示poi搜索区域（空字符串代表全国）  
 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);  
 poiSearch.setBound(new SearchBound(lp, 5000, true));//  
 // 设置搜索区域为以lp点为圆心，其周围5000米范围  
 poiSearch.searchPOIAsyn();// 异步搜索  
 }  
}

@Override  
public void onPoiSearched(PoiResult result, int rcode) {  
 if (rcode == 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  
 List<SuggestionCity> suggestionCities = poiResult  
 .getSearchSuggestionCitys();// 当搜索不到poiitem数据时，会返回含有搜索关键字的城市信息  
 if (poiItems != null && poiItems.size() > 0) {  
 //清除POI信息显示  
 whetherToShowDetailInfo(false);  
 //并还原点击marker样式  
 if (mlastMarker != null) {  
 resetlastmarker();  
 }   
 //清理之前搜索结果的marker  
 if (poiOverlay !=null) {  
 poiOverlay.removeFromMap();  
 }  
 mAMap.clear();  
 poiOverlay = new myPoiOverlay(mAMap, poiItems);  
 poiOverlay.addToMap();  
 poiOverlay.zoomToSpan();  
   
 mAMap.addMarker(new MarkerOptions()  
 .anchor(0.5f, 0.5f)  
 .icon(BitmapDescriptorFactory  
 .fromBitmap(BitmapFactory.decodeResource(  
 getResources(), R.drawable.point4)))  
 .position(new LatLng(lp.getLatitude(), lp.getLongitude())));  
   
 mAMap.addCircle(new CircleOptions()  
 .center(new LatLng(lp.getLatitude(),  
 lp.getLongitude())).radius(5000)  
 .strokeColor(Color.BLUE)  
 .fillColor(Color.argb(50, 1, 1, 1))  
 .strokeWidth(2));  
 } else if (suggestionCities != null  
 && suggestionCities.size() > 0) {  
 showSuggestCity(suggestionCities);  
 } else {  
 ToastUtil.show(PoiAroundSearchActivity.this,  
 R.string.no\_result);  
 }  
 }  
 } else {  
 ToastUtil  
 .show(PoiAroundSearchActivity.this, R.string.no\_result);  
 }  
 }  
}

## 每日作业

1.掌握地图key的配置

2.会使用定位

3.了解周边搜索

4.了解支付宝支付