

[android] 百度地图开发 (二).定位城市位置和城市POI搜索

原创 Eastmount 最后发布于2014-12-24 02:41:34 阅读数 16380 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



Eastmount

¥9.90

去订阅

一. 百度地图城市定位和POI搜索知识

上一篇文章“[百度地图开发\(一\)](#)”中讲述了如何申请百度APIKey及解决显示空白网格的问题.该篇文章主要讲述如何定位城市位置、定位自己的位置和进行城市兴趣点POI(Point of Interest)搜索.那么如何在百度地图上定位某一个位置呢?

通过类GeoPoint可以定义经纬度,它存放着纬度值和经度值,通过getLastKnownLocation()方法可以获取Location对象,再定位经纬度设置其为地图中心即可显示当前位置.

其中Geopoint(纬度值,经度值)以微度为单位,需要乘以10的6次方.核心代码如下:

```
// 通过网络获取当前位置
String provider = LocationManager.NETWORK_PROVIDER;
Location location = locationManager.getLastKnownLocation(provider);
Geopoint point = new Geopoint((int)(location.getLatitude*1E6),
    (int)(location.getLongitude*1E6));
mMapController.setCenter(point);
```

同样的道理,如果知道了城市的经纬度就可以设置其为当前地图中心,这样就实现了定位城市位置的功能.那么怎样获取城市的经纬度呢?

百度Map API提供MKSearch.geocode(String address, String city)方法进行GEO地理编码检索,它的意思就是搜索某个城市具体地址的位置,而如果只搜索城市使用geocode(city, city)即可.同时逆地址解析函数MKSearch.reverseGeocode(new GeoPoint(latitude, longitude))可以实现通过输入经纬度查询具体地址.

其中核心代码如下(代码放置位置不同,详见后面实例):

```
// 初始化MKSearch
mMKSearch = new MKSearch();
mMKSearch.init(mBMapManager, new MySearchListener());
// 搜索城市
mMKSearch.geocode(city, city);
// 内部类实现MKSearchListener接口,实现异步搜索服务
public class MySearchListener implements MKSearchListener {

    @Override
    public void onGetAddrResult(MKAddrInfo result, int iError) {
        // 经纬度与地址搜索
        ...
        mMapController.setCenter(result.geoPt);
    }
}
```

其中百度地图API搜索主要通过初始化MKSearch类,同时其结果监听对象MKSearchListener类来实现一部搜索服务.在该类中有很多方法实现不同功能,其中onGetAddrResult()方法可以根据经纬度搜索地址信息,而我们需要实现的POI兴趣点搜索是通过onGetPoiResult()实现的,同样公交线路等搜索都可以通过它实现.

具体核心代码如下:

```
// 内部类实现MKSearchListener接口,实现异步搜索服务
public class MySearchListener implements MKSearchListener {
    // 经纬度与地址搜索结果
    public void onGetAddrResult(MKAddrInfo result, int iError) {
```

```

    }
    //POI搜索结果（范围检索、城市POI检索、周边检索）
    public void onGetPoiResult(MKPoiResult result, int type, int iError) {
    }
    // 驾车路线搜索结果
    public void onGetDrivingRouteResult(MKDrivingRouteResult result, int iError) {
    }
    // 公交换乘路线搜索结果
    public void onGetTransitRouteResult(MKTransitRouteResult result, int iError) {
    }
    // 步行路线搜索结果
    public void onGetWalkingRouteResult(MKWalkingRouteResult result, int iError) {
    }
    // 获取详细信息
    public void onGetBusDetailResult(MKBusLineResult arg0, int arg1) {
    }
    public void onGetPoiDetailSearchResult(int arg0, int arg1) {
    }
    public void onGetShareUrlResult(MKShareUrlResult arg0, int arg1,int arg2) {
    }
    public void onGetSuggestionResult(MKSuggestionResult arg0, int arg1) {
    }
}

```

在android使用百度地图中可以添加地图覆盖物,那么什么是覆盖物呢?

所有叠加或覆盖到地图的内容,我们统称为地图覆盖物.如标注、矢量图形元素(包括:折线和多边形和圆)、定位图标等.覆盖物拥有自己的地理坐标,当您拖动或缩放地图时,它们会相应的移动.

地图API提供了如下几种覆盖物:

- 1.Overlay 覆盖物的抽象基类,所有的覆盖物均继承此类的方法,实现用户自定义图层显示.
- 2.MyLocationOverlay 一个负责显示用户当前位置的Overlay.
- 3.ItemizedOverlay<Item extends OverlayItem> 它是Overlay的一个基类,包含了一个OverlayItem列表,相当于一组分条的Overlay,通过继承此类将一组兴趣点显示在地图上.
- 4.PoiOverlay 本地搜索图层,提供某一特定地区的位置搜索服务,比如在北京市搜索“公园”,通过此图层将公园显示在地图上.
- 5.RouteOverlay 步行驾车导航线路图层,将步行驾车出行方案的路线及关键点显示在地图上.
- 6.TransitOverlay 公交换乘线路图层,将某一特定地区的公交出行方案的路线及换乘位置显示在地图上.

我们这里可以使用MyLocationOverlay定位自己当前位置添加覆盖物,也可以在POI搜索过程中通过PoiOverlay添加搜索的兴趣点覆盖物.下面讲述代码及实现.

二. 源码实现

下载地址:<http://download.csdn.net/detail/eastmount/8292969>

首先,设置其activity_main.xml布局

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#000000"
    tools:context="com.example.baidumapshow.MainActivity"
    tools:ignore="MergeRootFrame" >
    <!-- 顶部路径 -->
    <RelativeLayout
        android:id="@+id/MyLayout_top"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="40dp"
        android:layout_alignParentTop="true"

```

```

        android:gravity="center">

            <LinearLayout
                android:orientation="horizontal"
                android:layout_width="fill_parent"
                android:layout_height="wrap_content"
                android:background="@null"
                android:padding="0dip" >
                <EditText android:id="@+id/city_edittext"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="center_vertical"
                    android:layout_marginLeft="5dp"
                    android:background="#ffffff"
                    android:textSize="22dp"
                    android:hint="输入城市"
                    android:layout_weight="15" />
                <EditText android:id="@+id/keyword_edittext"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="center_vertical"
                    android:layout_marginLeft="5dp"
                    android:background="#ffffff"
                    android:textSize="22dp"
                    android:hint="输入关键词"
                    android:layout_weight="25" />
                <Button android:id="@+id/query_button"
                    android:layout_width="wrap_content"
                    android:layout_height="wrap_content"
                    android:layout_gravity="center_vertical"
                    android:textColor="#ffffff"
                    android:textSize="20dp"
                    android:text="搜索" />
            </LinearLayout>
        </RelativeLayout>
    <!-- 底部添加按钮 -->
    <RelativeLayout
        android:id="@+id/MyLayout_bottom"
        android:orientation="horizontal"
        android:layout_width="fill_parent"
        android:layout_height="50dp"
        android:layout_alignParentBottom="true"
        android:gravity="center">
        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:orientation="horizontal"
            android:layout_alignParentBottom="true" >
            <Button
                android:id="@+id/button1"
                android:layout_width="wrap_content"
                android:layout_height="match_parent"
                android:layout_weight="1"
                android:textColor="#ffffff"
                android:text="定位" />
        </LinearLayout>
    </RelativeLayout>
    <!-- 中部显示图片 -->
    <RelativeLayout
        android:id="@+id/Content_Layout"
        android:orientation="horizontal"
        android:layout_width="fill_parent"

```

```

        android:layout_height="fill_parent"                android:layout_above="@id/MyLayout_bottom"
        android:layout_below="@id/MyLayout_top"
        android:gravity="center">
        <com.baidu.mapapi.map.MapView
            android:id="@+id/map_view"
            android:layout_width="fill_parent"
            android:layout_height="fill_parent"
            android:clickable="true" />
    </RelativeLayout>
</RelativeLayout>

```

然后是MainActivity.java源码

```

public class MainActivity extends Activity {

    //BMapManager 对象管理地图、定位、搜索功能
    private BMapManager mBMapManager;
    private MapView mapView = null;           // 地图主控件
    private MapController mMapController = null; // 地图控制
    MKMapViewListener mMapListener = null;    // 处理地图事件回调
    private MKSearch mMKSearch;              // 定义搜索服务类
    // 搜索
    private EditText keyWordEditText;
    private EditText cityEditText;
    private Button queryButton;
    private static StringBuilder sb;
    private MyLocationOverlay myLocationOverlay;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        /**
         * 创建对象BMapManager并初始化操作
         * V2.3.1中init(APIKey,null) V2.4.1在AndroidManifest中赋值AK
         * 注意 初始化操作在setContentView()前
         */
        mBMapManager = new BMapManager(getApplicationContext());
        mBMapManager.init(null);
        setContentView(R.layout.activity_main);
        mapView = (MapView) findViewById(R.id.map_view);
        cityEditText = (EditText) findViewById(R.id.city_edittext);
        keyWordEditText = (EditText) findViewById(R.id.keyword_edittext);
        queryButton = (Button) findViewById(R.id.query_button);

        mMapController = mapView.getController(); // 获取地图控制器
        mMapController.enableClick(true);         // 设置地图是否响应点击事
        mMapController.setZoom(16);               // 地图缩放级别(3-19级) 级别越高信息越详细
        mapView.setBuiltInZoomControls(true);     // 显示内置缩放控件

        /**
         * 获取学校经纬度 设置地图中心点
         */
        GeoPoint point = new GeoPoint((int)(39.96703 * 1E6), (int)(116.323772 * 1E6));
        mMapController.setCenter(point);
        mapView.regMapViewListener(mBMapManager, new MKMapViewListener() {

            /**
             * 地图移动完成时会回调此接口方法

```

```

        */
        @Override
        public void onMapMoveFinish() {
            //Toast.makeText(MainActivity.this, "地图移动", Toast.LENGTH_SHORT).show();
        }

        /**
         * 地图加载完毕回调此接口方法
         */
        @Override
        public void onMapLoadFinish() {
            Toast.makeText(MainActivity.this, "地图载入", Toast.LENGTH_SHORT).show();
        }

        /**
         * 地图完成带动画的操作（如: animationTo()）后, 此回调被触发
         */
        @Override
        public void onMapAnimationFinish() {

        }

        /**
         * 当调用过 mMapView.getCurrentMap()后, 此回调会被触发
         * 可在此保存截图至存储设备
         */
        @Override
        public void onGetCurrentMap(Bitmap arg0) {

        }

        /**
         * 点击地图上被标记的点回调此方法
         */
        @Override
        public void onClickMapPoi(MapPoi arg0) {
            if (arg0 != null){
                Toast.makeText(MainActivity.this, arg0.strText, Toast.LENGTH_SHORT).show();
            }
        }
    });

    /**
     * 初始化MKSearch 调用城市和POI搜索
     */
    mMKSearch = new MKSearch();
    mMKSearch.init(mBMapManager, new MySearchListener());
    queryButton.setOnClickListener(new OnClickListener() {
        @Override
        public void onClick(View v) {
            mMapController = mapView.getController();
            mMapController.setZoom(10);
            sb = new StringBuilder(); //内容清空
            //输入正确城市关键字
            String city = cityEditText.getText().toString().trim();
            String keyWord = keyWordEditText.getText().toString().trim();
            if(city.isEmpty()) { //默认城市设置为贵阳
                city="贵阳";
            }
            //如果关键字为空只搜索城市 GEO搜索 geocode(adress,city) 具体地址和城市
            if(keyWord.isEmpty()) {

```

```

        mMKSearch.geocode(city, city);
    }

    else {
        // 搜索城市+关键字
        mMKSearch.setPoiPageCapacity(10); // 每页返回POI数
        mMKSearch.poiSearchInCity(city, keyWord);
    }
}

});
}

@Override
protected void onResume() {
    mapView.onResume();
    if (mBMapManager != null) {
        mBMapManager.start();
    }
    super.onResume();
}

@Override
protected void onDestroy() {
    mapView.destroy();
    if (mBMapManager != null) {
        mBMapManager.destroy();
        mBMapManager = null;
    }
    super.onDestroy();
}

@Override
protected void onPause() {
    mapView.onPause();
    if (mBMapManager != null) {
        mBMapManager.stop();
    }
    super.onPause();
}

/**
 * 内部类实现MKSearchListener接口,用于实现异步搜索服务
 */
public class MySearchListener implements MKSearchListener {

    /**
     * 根据经纬度搜索地址信息结果
     * 同时mMKSearch.geocode(city, city)搜索城市返回至该函数
     *
     * @param result 搜索结果
     * @param iError 错误号 (0表示正确返回)
     */
    @Override
    public void onGetAddrResult(MKAddrInfo result, int iError) {
        if (result == null) {
            return;
        }
        StringBuffer sbcity = new StringBuffer();
        sbcity.append(result.strAddr).append("\n"); // 经纬度所对应的位置
        mapView.getOverlays().clear(); // 清除地图上已有的所有覆盖物
        mMapController.setCenter(result.geoPt); // 置为地图中心
        // 添加原点并刷新
        LocationData locationData = new LocationData();

```

```

locationData.latitude = result.geoPt.getLatitudeE6();
locationData.longitude = result.geoPt.getLongitudeE6();
myLocationOverlay = new MyLocationOverlay(mapView);
myLocationOverlay.setData(locationData);
    mapView.getOverlays().add(myLocationOverlay);
    mapView.refresh();
    // 通过AlertDialog显示地址信息
new AlertDialog.Builder(MainActivity.this)
    .setTitle("显示当前城市地图")
    .setMessage(sbcity.toString())
    .setPositiveButton("关闭", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int whichButton) {
            dialog.dismiss();
        }
    }).create().show();
}

/**
 * POI搜索结果（范围检索、城市POI检索、周边检索）
 *
 * @param result 搜索结果
 * @param type 返回结果类型（11,12,21:poi列表 7:城市列表）
 * @param iError 错误号（0表示正确返回）
 */
@Override
public void onGetPoiResult(MKPoiResult result, int type, int iError) {
    if (result == null) {
        return;
    }
    // 获取POI并显示
    mapView.getOverlays().clear();
    PoiOverlay poioverlay = new PoiOverlay(MainActivity.this, mapView);
    poioverlay.setData(result.getAllPoi()); // 设置搜索到的POI数据
    mapView.getOverlays().add(poioverlay); // 兴趣点标注在地图上
    mapView.refresh();
    // 设置其中一个搜索结果所在地理坐标为地图的中心
    if(result.getNumPois() > 0) {
        MKPoiInfo poiInfo = result.getPoi(0);
        mMapController.setCenter(poiInfo.pt);
    }
    // 添加StringBuffer 遍历当前页返回的POI（默认只返回10个）
    sb.append("共搜索到").append(result.getNumPois()).append("个POI\n");
    for (MKPoiInfo poiInfo : result.getAllPoi()) {
        sb.append("名称: ").append(poiInfo.name).append("\n");
    }
    // 通过AlertDialog显示当前页搜索到的POI
    new AlertDialog.Builder(MainActivity.this)
        .setTitle("搜索到的POI信息")
        .setMessage(sb.toString())
        .setPositiveButton("关闭", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int whichButton) {
                dialog.dismiss();
            }
        }).create().show();
}

/**
 * 驾车路线搜索结果
 *
 * @param result 搜索结果
 * @param iError 错误号（0表示正确返回）
 */

```

```

@Override
    public void onGetDrivingRouteResult(MKDrivingRouteResult result, int iError) {
    }

    /**
     * 公交换乘路线搜索结果
     *
     * @param result 搜索结果
     * @param iError 错误号 (0表示正确返回)
     */
    @Override
    public void onGetTransitRouteResult(MKTransitRouteResult result, int iError) {
    }

    /**
     * 步行路线搜索结果
     *
     * @param result 搜索结果
     * @param iError 错误号 (0表示正确返回)
     */
    @Override
    public void onGetWalkingRouteResult(MKWalkingRouteResult result, int iError) {
    }

    @Override
    public void onGetBusDetailResult(MKBusLineResult arg0, int arg1) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onGetPoiDetailSearchResult(int arg0, int arg1) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onGetShareUrlResult(MKShareUrlResult arg0, int arg1, int arg2) {
        // TODO Auto-generated method stub
    }

    @Override
    public void onGetSuggestionResult(MKSuggestionResult arg0, int arg1) {
        // TODO Auto-generated method stub
    }
}
}

```

最后设置AndroidManifest.xml文件,主要是申请网络权限和设置APIKey

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.baidumapshow"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="19"
        android:targetSdkVersion="19" />

    <!-- 获取网络状态 -->
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

```



```

<!-- 访问网络 -->
        <uses-permission android:name="android.permission.INTERNET" />
<!-- 获取WiFi状态 -->
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<!-- 允许程序写入外部存储,如SD卡上写文件 -->
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_SETTINGS" />
<!-- 读取电话状态 -->
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<!-- 获取精确位置 GPS芯片接收卫星的定位信息,定位精度达10米以内 -->
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<!-- 通过WiFi或移动基站的方式获取用户粗略的经纬度信息 -->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<!-- 获取模拟定位信息 -->
<uses-permission android:name="android.permission.ACCESS_mock_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_GPS" />

<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >
    <meta-data
        android:name="com.baidu.lbsapi.API_KEY"
        android:value="QwaNhFQ0ty2QmdYh3Nrr0gQx">
    </meta-data>
    <activity
        android:name="com.example.baidumapshow.MainActivity"
        android:label="@string/app_name" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

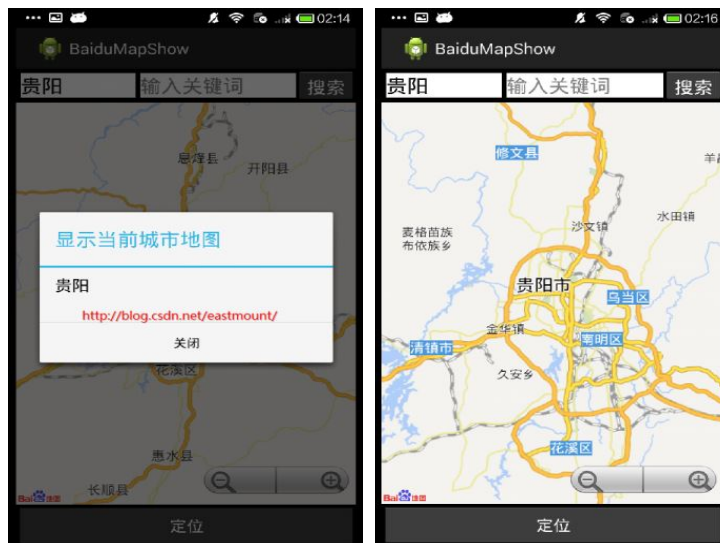
            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>

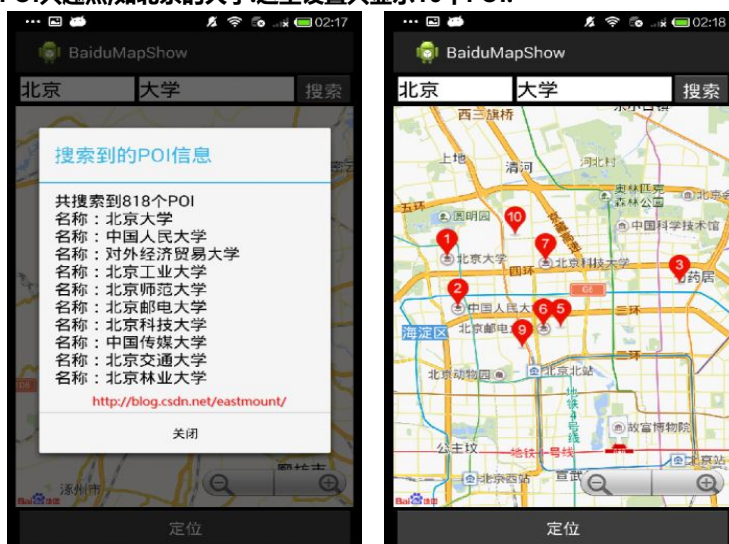
```

程序运行结果如下图所示:

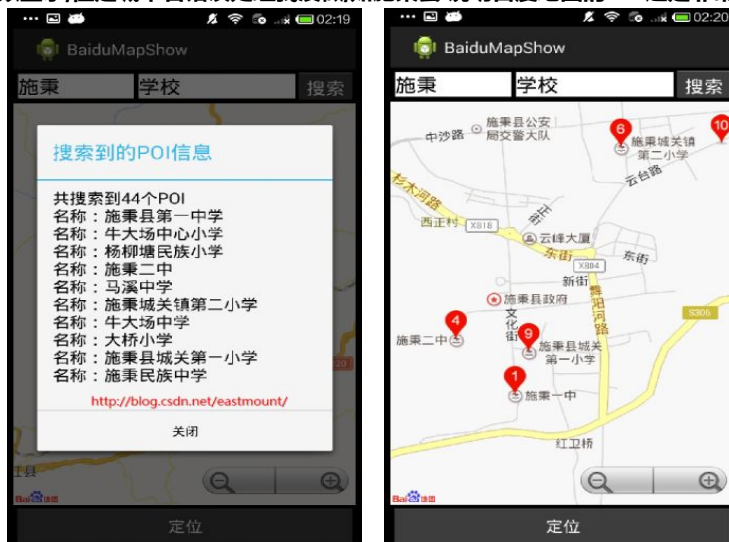
当只输入城市名的时候显示的是城市对应的地图.如下图贵阳.



当输入城市+关键字时显示POI兴趣点,如北京的大学.这里设置只显示10个POI.



最后我测试了下县份同样可以显示,但是城市名错误处理我没做,如施秉县.说明百度地图的API还是非常强大的,希望后面接着学习吧!



最后希望文章对大家有所帮助,刚刚接触android开发百度地图,而且还是使用V2.4.1版本,同时搜索城市时没有显示覆盖物不知道其原因.如果有错误或不足之处,还请海涵!建议大家看看官方文档和百度提供的Demo.文章主要参考百度官方文档、柳峰大神博客和《Android第一行代码》.

下载地址: <http://download.csdn.net/detail/eastmount/8292969>

(By:Eastmount 2014-12-24 半夜3点 <http://blog.csdn.net/eastmount/>)

参考资料及在线笔记:

百度官方文档 <http://developer.baidu.com/map/sdkandev-4.htm>

百度官方文档 [Android SDK开发指南之覆盖物](#)

百度地图相关Demo下载 [Demo v3.2.0](#)

柳峰博客 [\[010\]百度地图API之根据经纬度查询地址信息\(Android](#)

柳峰博客 [\[013\] 百度地图API之城市POI搜索-获取所有结果\(Android\)](#)

👍 点赞 6 ☆ 收藏 ➦ 分享 ...



Eastmount 博客专家

发布了445 篇原创文章 · 获赞 5981 · 访问量 487万+

他的留言板

关注