
第9章 位置服务与地图应用

本章学习目标：

- 了解位置服务的概念
 - 了解地图密钥的申请方法
 - 掌握获取位置信息的方法
 - 掌握MapView和MapController的使用方法
 - 掌握Google地图覆盖层的使用方法
-

9.1 位置服务

■ 位置服务

- 通过电信移动运营商的无线电通讯网络或外部定位方式，获取移动终端用户的位置信息，在GIS平台的支持下，可以为用户提供很多便捷服务(LBS)
- 定位方式
 - GPS、基站、Wi-Fi、签到
- API自动处理对地图服务的获取，数据下载，地图显示，还有地图上的触摸手势等。还可以利用API的函数增加一些标记、多边形和覆盖，还可以改变用户对于一个特定地图的视角。

9.1 位置服务

■ 位置服务

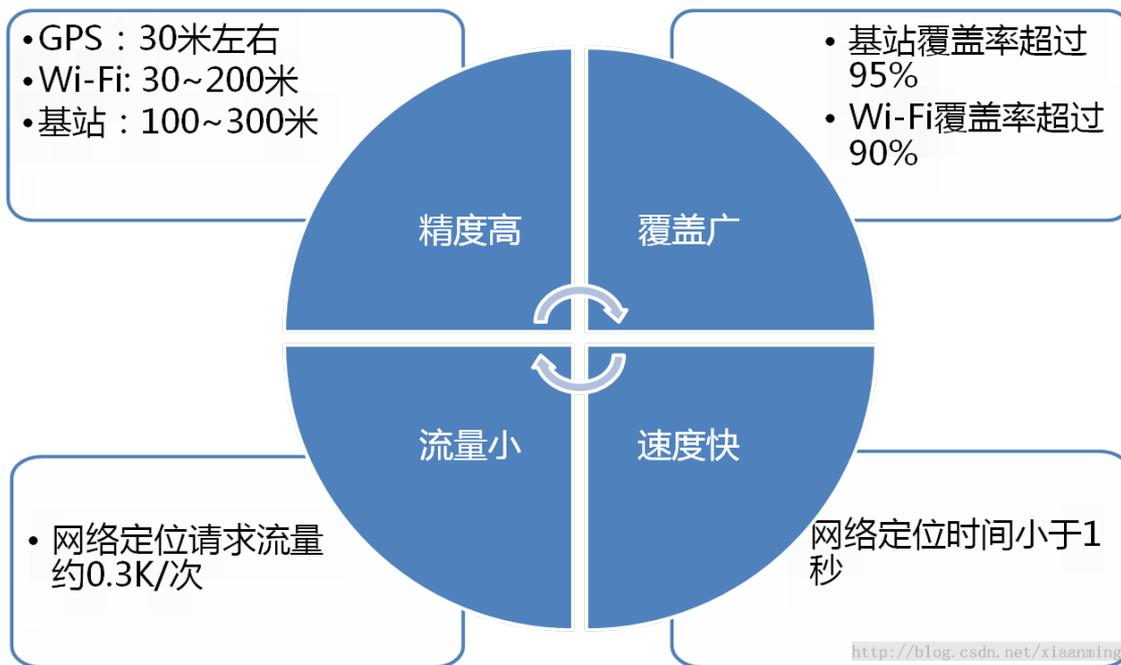
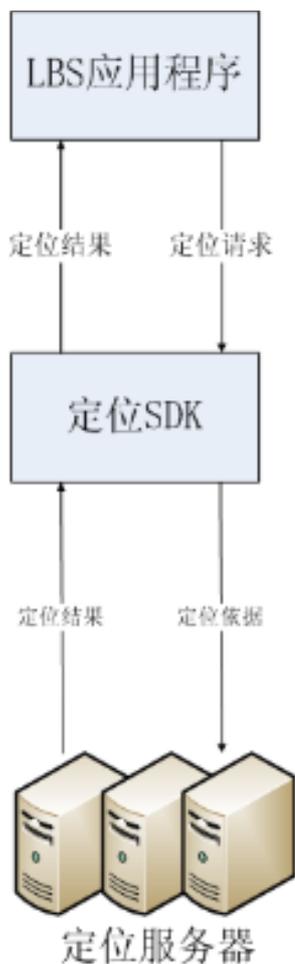
□ 定位原理

- 使用百度Android定位SDK必须注册GPS和网络使用权限。当应用程序向定位SDK发起定位请求时，定位SDK会根据应用的定位因素（GPS、基站、Wi-Fi信号）的实际情况（如是否开启GPS、是否连接网络、是否有信号等）来生成相应定位依据进行定位。
- 用户可以设置满足自身需求的定位依据：
若用户设置GPS优先，则优先使用GPS进行定位，如果GPS定位未打开或者没有可用位置信息，且网络连接正常，定位SDK则会返回网络定位（即Wi-Fi与基站）的最优结果。为了使获得的网络定位结果更加精确，请打开手机的Wi-Fi开关。

9.1 位置服务

■ 位置服务

□ 定位原理



9.1 位置服务

■ 位置服务

- **Android**平台支持提供位置服务的**API**，在开发过程中主要用到**LocationManager**和**LocationProviders**对象
- **LocationManager**可以用来获取当前的位置，追踪设备的移动路线，或设定敏感区域，在进入或离开敏感区域时设备会发出特定警报
- **LocationProviders**则是提供定位功能的组件集合，集合中的每种组件以不同的技术提供设备的当前位置，区别在于定位的精度、速度和成本等方面

9.1 位置服务

■ 位置服务

□ 为了使开发的程序能够提供位置服务，首先的问题是如何获取LocationManager

□ 获取LocationManager可以通过调用android.app.Activity.getSystemService()函数获取，代码如下

```
1 String serviceString = Context.LOCATION_SERVICE;
```

```
2 LocationManager locationManager = (LocationManager)getSystemService(serviceString);
```

■ 代码第1行的Context.LOCATION_SERVICE指明获取的是位置服务

■ 代码第2行的getSystemService()函数，可以根据服务名称获取Android提供的系统级服务

9.1 位置服务

■ 位置服务

□ Android支持的系统级服务表

Context类的静态常量	值	返回对象	说明
LOCATION_SERVICE	location	LocationManager	控制位置等设备的更新
WINDOW_SERVICE	window	WindowManager	最顶层的窗口管理器
LAYOUT_INFLATER_SERVICE	layout_inflater	LayoutInflater	将 XML 资源实例化为 View
NOTIFICATION_SERVICE	notification	NotificationManager	后台事件通知
POWER_SERVICE	power	PowerManager	电源管理
ALARM_SERVICE	alarm	AlarmManager	在指定时间接受Intent
KEYGUARD_SERVICE	keyguard	KeyguardManager	锁定或解锁键盘
SEARCH_SERVICE	search	SearchManager	访问系统的搜索服务
VIBRATOR_SERVICE	vibrator	Vibrator	访问支持振动的硬件
CONNECTIVITY_SERVICE	connection	ConnectivityManager	网络连接管理
WIFI_SERVICE	wifi	WifiManager	Wi-Fi连接管理
INPUT_METHOD_SERVICE	input_method	InputMethodManager	输入法管理

9.1 位置服务

■ 位置服务

- 在指定LocationManager的定位方法后，则可以调用getLastKnownLocation()方法获取当前的位置信息。以使用GPS定位为例，获取位置信息的代码如下

```
1 String provider = LocationManager.GPS_PROVIDER;  
2 Location location = locationManager.getLastKnownLocation(provider);
```

- 代码第2行返回的Location对象中，包含了可以确定位置的信息，如经度、纬度和速度等

- 通过调用Location中的getLatitude()和getLongitude()方法可以分别获取位置信息中的纬度和经度，示例代码如下

```
1 double lat = location.getLatitude();  
2 double lng = location.getLongitude();
```

9.1 位置服务

■ 位置服务

□ Location类的常用方法

No.	方法	描述
1	getAccuracy()	精确度
2	getBearing()	方位
3	getExtras()	所有附加的信息
4	getProvider()	数据提供者
5	getSpeed()	速度
6	getTime()	时间

9.1 位置服务

■ 位置服务

- 在很多提供定位服务的应用程序中，不仅需要获取当前的位置信息，还需要监视位置的变化，在位置改变时调用特定的处理方法
- LocationManager提供了一种便捷、高效的位置监视方法 `requestLocationUpdates()`，可以根据位置的距离变化和时间间隔设定，产生位置改变事件的条件，这样可以避免因微小的距离变化而产生大量的位置改变事件
- LocationManager中设定监听位置变化的代码如下

```
locationManager.requestLocationUpdates(provider, 2000, 10, locationListener);
```

- 第1个参数是定位的方法，GPS定位或网络定位
- 第2个参数是产生位置改变事件的时间间隔，单位为毫秒
- 第3个参数是距离条件，单位是米
- 第4个参数是回调函数，用于处理位置改变事件

9.1 位置服务

■ 位置服务

- 为了使**GPS**定位功能生效，还需要在**AndroidManifest.xml**文件中加入用户许可

```
<!-- 访问精确位置的权限 -->
```

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

```
<!-- 访问网络状态的权限 -->
```

```
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

```
<!-- 访问WIFI网络状态的权限 -->
```

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

```
<!-- 改变WIFI网络状态的权限 -->
```

```
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
```

```
<!-- 读写存储卡的权限 -->
```

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<!-- 读取电话状态的权限 -->
```

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```

9.1 位置服务

■ 位置服务

□ 获得最佳的LocationProvider

- 不同的LocationProvider特性不同
- 每个应用数据提供商有各自的操作准则

□ 利用android.location.Criteria类进行条件过滤

No.	方法	类型	描述
1	ACCURACY_COARSE/FINE/HIGH/MEDIUM/LOW	常量	精度
2	POWER_HIGH/MEDIUM/LOW	常量	电量
3	Criteria()	构造	创建一个Criteria对象
4	getAccuracy()	普通	取得当前设置的精度条件
5	isAltitudeRequired()	普通	是否提供高度信息
6	setAccuracy	普通	

9.2 百度地图应用

■ 9.2.1 申请地图密钥

- 为了在手机中更直观地显示地理信息，程序开发人员可以直接使用Google提供的地图服务，实现地理信息的可视化开发
- 只要使用MapView（`com.google.android.maps.MapView`）就可以将Google地图嵌入到Android应用程序中
- 但在使用MapView进行开发前，必须向Google申请经过验证的“地图密钥”（Map API Key），这样才能正常使用Google的地图服务
- “地图密钥”是访问Google地图数据的密钥，无论是模拟器还是在真实设备中需要使用这个密钥

9.2 百度地图应用

■ 9.2.1 申请地图密钥

- 在国内比较常用的是百度地图或者高德地图
 - <http://developer.baidu.com/map/android-mobile-apply-key.htm>
 - <http://id.amap.com/?ref=http%3A%2F%2Flbs.amap.com%2Fconsole%2F>
- 在申请**Key**的时需要填写应用名称，后期不可更改，因此**key**和应用名称是一一对应的关系。如果更换了应用包名则需要重新配置**key**

9.2 百度地图应用

■ 9.2.1 申请地图密钥

- 使用keytool工具来获取签名证书的sha1值，在DOS输入

keytool -list -keystore C:\Users\bds\.android\debug.keystore

- 红色部分为debug.keystore的路径，接下来会要输入输入密钥库口令，默认输入“android”，这样子我们就能获取证书指纹 (SHA1)

9.2 百度地图应用

■ 9.2.1 申请地图密钥

- 红色框里面的东西就是的Android签名证书的证书指纹(SHA1)值，然后我们复制出来，输入到API key的安全码输入框里面，再用分号隔开加上你的应用程序包名，02:5C:80:25:B2:8F:6F:60:54:B9:F4:B2:EF:94:FF:EE:CC:3C:5A:29;com.example.baidumapdemo 这样子就配置好了API key

```
C:\Users\bds>keytool -list -keystore "C:\Users\bds\.android\debug.keystore"  
输入密钥库口令:  
  
密钥库类型: JKS  
密钥库提供方: SUN  
  
您的密钥库包含 1 个条目  
  
androiddebugkey, 2013-3-5, PrivateKeyEntry,  
证书指纹 (SHA1): 02:5C:80:25:B2:8F:6F:60:54:B9:F4:B2:EF:94:FF:EE:CC:3C:5A:29
```

9.2 百度地图应用

■ 9.2.2 使用百度地图

□ 下载百度地图API库

- 要在Android应用中使用百度地图API，就需要在工程中引用百度地图API开发包，下载地址

[http://developer.baidu.com/map/sdkandev-](http://developer.baidu.com/map/sdkandev-download.htm)

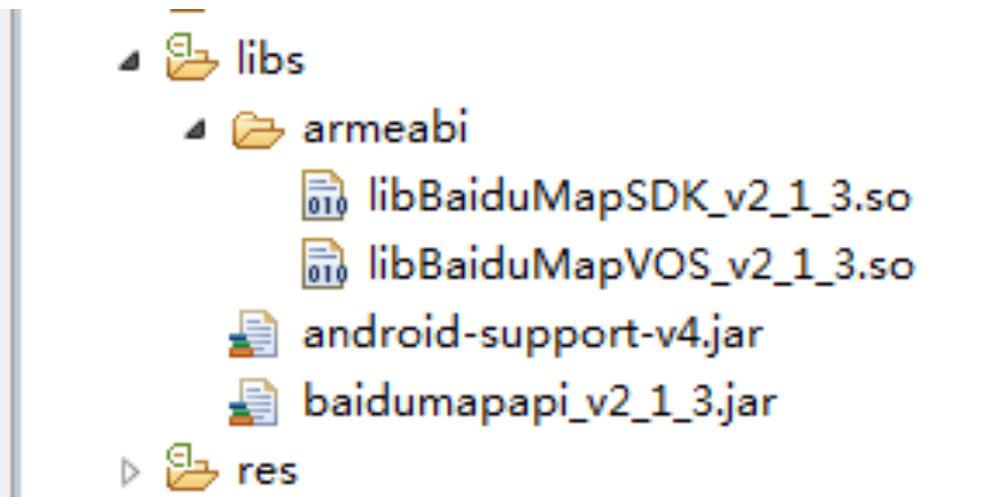
[download.htm](http://developer.baidu.com/map/sdkandev-download.htm)， 下载Android SDK v3.5.0 lib库就行了

9.2 百度地图应用

■ 9.2.2 使用百度地图

□ 下载百度地图API库

- 在Android项目中引用百度地图
- 将开发包中的jar包和so文件添加到libs文件下



9.2 百度地图应用

■ 9.2.2 使用百度地图

- 创建工程后，修改/res/layout/main.xml文件，布局文件中加入一个MapView控件，并设置刚获取的“地图密钥”。main.xml文件的完整代码如下

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:orientation="vertical"
4     android:layout_width="fill_parent"
5     android:layout_height="fill_parent">
6     <TextView android:layout_width="fill_parent"
7     android:layout_height="wrap_content"
8     android:text="@string/hello"/>
9
10     <com.google.android.maps.MapView
11     android:id="@+id/mapview"
12     android:layout_width="fill_parent"
13     android:layout_height="fill_parent"
14     android:enabled="true"
15     android:clickable="true"
16     android:apiKey="0mVK8GeO6WUzmtzGfNoqXI5KRW4oeh5ljVMoiCg"/>
17 </LinearLayout>
```

9.2 百度地图应用

■ 9.2.2 使用百度地图

- 仅在布局中添加MapView控件，还不能够直接在程序中调用这个控件，还需要将程序本身设置成MapActivity（com.baidu.android.maps.MapActivity）
- MapActivity类负责处理显示百度地图所需的生命周期和后台服务管理

```
package com.bdmap.view;  
import com.baidu.mapapi.SDKInitializer;  
import com.baidu.mapapi.map.BaiduMap;  
import com.baidu.mapapi.map.MapView;  
  
public class MainActivity extends Activity {  
    private MapView mMapView = null; // 百度地图控件  
    private BaiduMap bdMap;         // 百度地图对象  
    private MapController mMapController = null; //用MapController完成地图控制  
    MKMapViewListener mMapListener = null; //MKMapViewListener 用于处理地图事件回调
```

9.2 百度地图应用

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    mBMapManager = new BMapManager(this); //使用地图sdk前需先初始化BMapManager  
    , 这个必须在setContentView()先初始化  
    //第一个参数是API key, 第二个参数是常用事件监听, 用来处理通常的网络错误, 授权验证错误等, 你也可以不添加这个回调接口  
    mBMapManager.init(“你的API Key”, new MKGeneralListener() {.....})  
  
    mMapController = mMapView.getController(); //获取地图控制器  
    mMapController.enableClick(true); // 设置地图是否响应点击事件  
    mMapController.setZoom(12); // 设置地图缩放级别  
    mMapView.setBuiltInZoomControls(true); // 显示内置缩放控件  
    GeoPoint p = new GeoPoint((int)(22.547923 * 1E6), (int)(114.067368 * 1E6)); //保存精度和纬度的类  
    mMapController.setCenter(p); //设置p地方为中心点  
    mMapView.regMapViewListener(mBMapManager, new MKMapViewListener() {.....})  
}
```

9.2 百度地图应用

■ 9.2.2 使用百度地图

□ 普通地图和卫星地图切换

- 百度地图将地图的类型分为两种：普通矢量地图和卫星图

- `mMapView = (MapView) findViewById(R.id.bMapView);`
`mBaiduMap = mMapView.getMap();`

//普通地图

`mBaiduMap.setMapType(BaiduMap.MAP_TYPE_NORMAL);`

//卫星地图

`mBaiduMap.setMapType(BaiduMap.MAP_TYPE_SATELLITE);`

9.2 百度地图应用

■ 9.2.2 使用百度地图

□ 显示实时交通图（路况图）

- //开启交通图

- `mBaiduMap.setTrafficEnabled(true);`

9.2 百度地图应用

■ 9.2.2 使用百度地图

□ 显示热力图

- 热力图就是以特殊高亮的形式显示访客热衷的页面区域和访客所在的地理区域的图示。通俗来说就是显示地图上某一块区域的人的密集程度
- //开启热力图
- `mBaiduMap.setBaiduHeatMapEnabled(true);`