



《百度地图实战 实验手册 04》

Android 课程组

版本 1.0

文档提供：Java 课程组 丁盟

目录

第 4 章 百度定位	1
4.1 实验目的	1
4.2 准备工作	1
4.3 实验步骤	2
4.4 实验结论	15

第4章 百度定位

4.1 实验目的

- 目的一： 了解百度定位提供的定位模式。
- 目的二： 掌握基本的定位功能的使用流程。
- 目的三： 能够通过百度地图实时显示位置信息。
- 目的四： 能够通过定位信息获取 POI 信息点信息。

4.2 准备工作

准备一： 创建项目。

使用 Android Studio 创建一个新的空项目，并按照之前百度地图 SDK Android 开发环境配置对当前新项目进行配置。

准备二： 申请秘钥。

对当前项目申请百度地图开发者秘钥，并在项目中对秘钥信息进行设置。

当前面准备工作完成之后，在模拟器或真机中运行项目能够显示如下效果即表示准备工作已完成。



图 4.2.1

4.3 实验步骤

步骤一 在 MainActivity 中分别声明地图视图、地图控制器、定位客户端、定位监听器等属性成员。

```
/* 地图控件 */
private MapView mMapView = null;
/* 地图实例 */
private BaiduMap mBaiduMap;
/* 定位的客户端 */
private LocationClient mLocationClient;
/* 定位的监听器 */
public MyLocationListener mMyLocationListener;
/* 当前定位的模式 */
private MyLocationConfiguration.LocationMode mCurrentMode
    = MyLocationConfiguration.LocationMode.NORMAL;
/* 是否是第一次定位 */
private volatile boolean isFristLocation = true;

/* 最新一次的经纬度 */
private double mCurrentLantitude;
private double mCurrentLongitude;

/* 地图定位的模式 */
private String[] mStyles
    = new String[]{"地图模式【正常】",
        "地图模式【跟随】",
        "地图模式【罗盘】"};
/* 当前地图定位模式的 Index */
private int mCurrentStyle = 0;
```

步骤二 在 MainActivity 中创建 initBaiduMap() 方法，用来完成百度地图初始化工作，并在 onCreate() 中调用。

```
/**
 * 初始化百度地图
 */
private void initBaiduMap() {
    mMapView = (MapView)findViewById(R.id.bmapView);
    mBaiduMap = mMapView.getMap();
    MapStatusUpdate msu =
    MapStatusUpdateFactory.zoomTo(15.0f);
    mBaiduMap.setMapStatus(msu);
}
```

```
}
```

步骤三 实现百度定位客户端的定位回调接口类。

```
/**
 * 实现定位回调监听
 */
public class MyLocationListener implements
BDLocationListener {
    @Override
    public void onReceiveLocation(BDLocation location) {

        // mapView 销毁后不在处理新接收的位置
        if (location == null || mMapView == null)
            return;

        // 构造定位数据
        MyLocationData locData = new MyLocationData.Builder()
            // 此处设置开发者获取到的方向信息，顺时针0-360
            .latitude(location.getLatitude())
            .longitude(location.getLongitude()).build();

        // 设置BaiduMap的定位数据
        mBaiduMap.setMyLocationData(locData);
        // 记录位置信息
        mCurrentLatitude = location.getLatitude();
        mCurrentLongitude = location.getLongitude();

        // 第一次定位时，将地图位置移动到当前位置
        if (isFristLocation) {
            isFristLocation = false;
            center2myLoc();
        }

        // Log 记录位置信息
        StringBuffer sb = new StringBuffer(256);
        sb.append("\nlatitude : ");
        sb.append(location.getLatitude());
        sb.append("\nlontitude : ");
        sb.append(location.getLongitude());
        sb.append("\naddress : ");
        sb.append(location.getAddrStr());

        for(int i=0; i<location.getPoiList().size(); i++) {
            Poi p = location.getPoiList().get(i);
```

```

        sb.append("\nPoi NO.");
        sb.append(i);
        sb.append(" : ");
        sb.append(p.getId());
        sb.append("-");
        sb.append(p.getName());
        sb.append("-");
        sb.append(p.getRank());
    }

    Log.i("BaiduLocationInfo", sb.toString());
}
}

```

步骤四 在 MainActivity 中创建 initMyLocation () 方法，用来完成百度定位客户端的初始化工作，并在 onCreate() 中调用。

```

/**
 * 初始化定位相关代码
 */
private void initMyLocation() {
    // 定位 SDK 初始化
    mLocationClient = new
    LocationClient(getApplicationContext());

    // 设置定位的相关配置
    LocationClientOption option = new
    LocationClientOption();
    option.setOpenGps(true); // 打开 gps
    option.setCoorType("bd09ll"); // 设置坐标类型
    option.setScanSpan(1000); // 自动定位间隔
    option.setIsNeedAddress(true); // 是否需要地址
    option.setIsNeedLocationPoiList(true);

    // 定位模式

    option.setLocationMode(LocationClientOption.LocationMode.
    Hight_Accuracy);

    // 根据配置信息对定位客户端进行设置
    mLocationClient.setLocOption(option);

    // 注册定位监听

```

```

        mMyLocationListener = new MyLocationListener();

mLocationClient.registerLocationListener(mMyLocationListener);

        // 设置定位图标
        BitmapDescriptor mCurrentMarker =
        BitmapDescriptorFactory
            .fromResource(R.drawable.location);
        MyLocationConfiguration config = new
        MyLocationConfiguration(
            mCurrentMode, true, mCurrentMarker);
        mBaiduMap.setMyLocationConfiguration(config);
    }

```

步骤五 实现百度地图、百度定位客户端的生命周期相关接口。

```

@Override
protected void onStart()
{
    // 开启图层定位
    mBaiduMap.setMyLocationEnabled(true);
    if (!mLocationClient.isStarted())
    {
        mLocationClient.start();
    }

    super.onStart();
}

@Override
protected void onStop()
{
    // 关闭图层定位
    mBaiduMap.setMyLocationEnabled(false);
    mLocationClient.stop();

    super.onStop();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    // 在activity执行onDestroy时执行mMapView.onDestroy(),
    实现地图生命周期管理
}

```

```

        mMapView.onDestroy();
    }
    @Override
    protected void onResume() {
        super.onResume();
        // 在 activity 执行 onResume 时执行 mMapView. onResume (),
        实现地图生命周期管理
        mMapView.onResume();
    }
    @Override
    protected void onPause() {
        super.onPause();
        // 在 activity 执行 onPause 时执行 mMapView. onPause (), 实现
        地图生命周期管理
        mMapView.onPause();
    }
}

```

步骤六 定义 center2myLoc 函数来实现 BaiduMap 移动到定位位置。

```

/**
 * BaiduMap 移动到我的位置
 */
private void center2myLoc() {
    LatLng ll = new LatLng(mCurrentLantitude,
mCurrentLongitude);
    // 设置当前定位位置为 BaiduMap 的中心点, 并移动到定位位置
    MapStatusUpdate u =
MapStatusUpdateFactory.newLatLng(ll);
    mBaiduMap.animateMapStatus(u);
}

```

步骤七 在 res 目录下创建 menu 目录, 并添加 main.xml 文件。

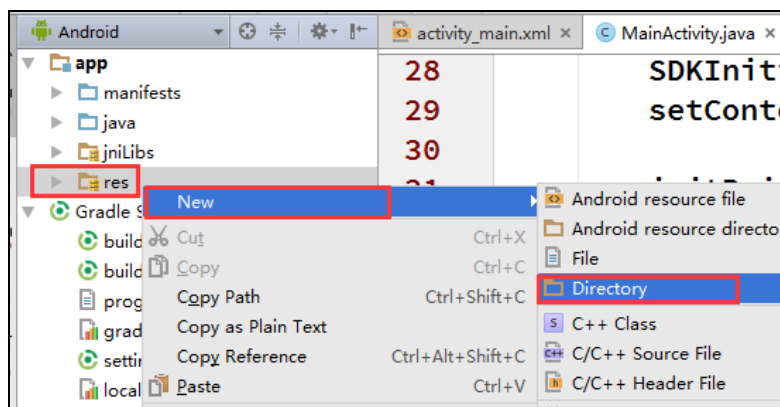


图 4.3.1

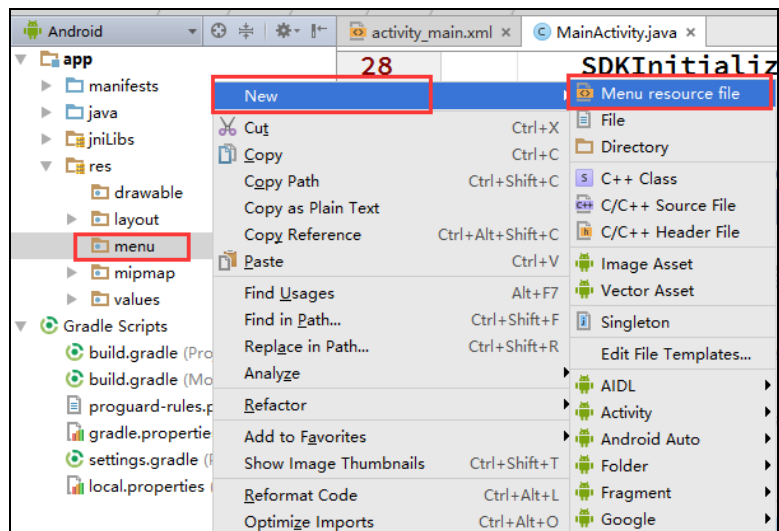


图 4.3.2

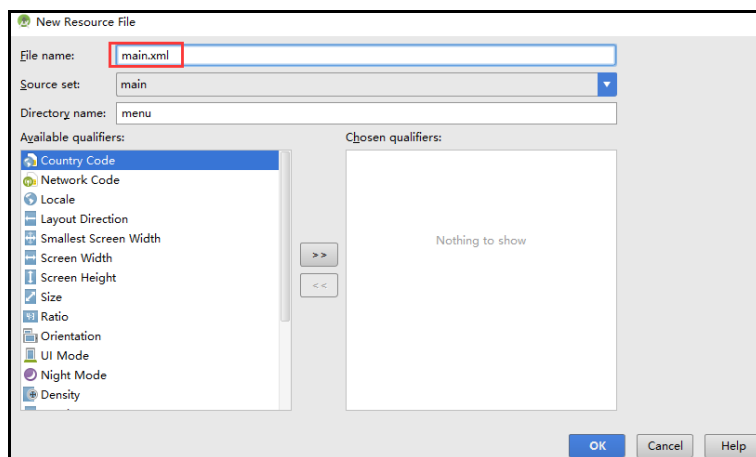


图 4.3.3

步骤八 当 main.xml 文件创建完成之后，在其中添加如下 item 项。(main.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<menu
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto">

  <item
    android:id="@+id/id_menu_map_myLoc"
    android:title="我的位置"
    app:showAsAction="never" />

  <item
    android:id="@+id/id_menu_map_style"
```

```
        android:title="地图模式【正常】"  
        app:showAsAction="never" />  
  
</menu>
```

步骤九 在 MainActivity 类中实现菜单创建接口。

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu, menu);  
    return true;  
}
```

步骤十 在 MainActivity 类中实现菜单点击相应接口。

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.id_menu_map_myLoc: // 标注覆盖物  
            center2myLoc();  
            break;  
  
        case R.id.id_menu_map_style: // 地图模式  
            mCurrentStyle = (++mCurrentStyle) %  
mStyles.length;  
            item.setTitle(mStyles[mCurrentStyle]);  
            // 设置自定义图标  
            switch (mCurrentStyle)  
            {  
                case 0:  
                    mCurrentMode =  
MyLocationConfiguration.LocationMode.NORMAL;  
                    break;  
                case 1:  
                    mCurrentMode =  
MyLocationConfiguration.LocationMode.FOLLOWING;  
                    break;  
                case 2:  
                    mCurrentMode =  
MyLocationConfiguration.LocationMode.COMPASS;  
                    break;  
            }  
  
            BitmapDescriptor mCurrentMarker =
```

```

BitmapDescriptorFactory
        .fromResource(R.drawable.location);
    MyLocationConfiguration config = new
MyLocationConfiguration(
        mCurrentMode, true, mCurrentMarker);
    mBaiduMap.setMyLocationConfiguration(config);
    break;
}

return super.onOptionsItemSelected(item);
}

```

步骤十一 最终 MainActivity 类完整代码如下。(MainActivity.java)

```

public class MainActivity extends AppCompatActivity {
    /* 地图控件 */
    private MapView mMapView = null;
    /* 地图实例 */
    private BaiduMap mBaiduMap;
    /* 定位的客户端 */
    private LocationClient mLocationClient;
    /* 定位的监听器 */
    public MyLocationListener mMyLocationListener;
    /* 当前定位的模式 */
    private MyLocationConfiguration.LocationMode
mCurrentMode
        = MyLocationConfiguration.LocationMode.NORMAL;
    /* 是否是第一次定位 */
    private volatile boolean isFristLocation = true;

    /* 最新一次的经纬度 */
    private double mCurrentLantitude;
    private double mCurrentLongitude;

    /* 地图定位的模式 */
    private String[] mStyles
        = new String[]{"地图模式【正常】",
            "地图模式【跟随】",
            "地图模式【罗盘】"};
    /* 当前地图定位模式的 Index */
    private int mCurrentStyle = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
        SDKInitializer.initialize(getApplicationContext());
        setContentView(R.layout.activity_main);

        // 初始化 BaiduMap 相关
        initBaiduMap();

        // 初始化百度定位客户端
        initMyLocation();
    }

    /**
     * 初始化百度地图
     */
    private void initBaiduMap() {
        mMapView = (MapView)findViewById(R.id.bmapView);
        mBaiduMap = mMapView.getMap();
        MapStatusUpdate msu =
        MapStatusUpdateFactory.zoomTo(15.0f);
        mBaiduMap.setMapStatus(msu);
    }

    /**
     * 初始化定位相关代码
     */
    private void initMyLocation() {
        // 定位 SDK 初始化
        mLocationClient = new
        LocationClient(getApplicationContext());

        // 设置定位的相关配置
        LocationClientOption option = new
        LocationClientOption();
        option.setOpenGps(true); // 打开 gps
        option.setCoorType("bd09ll"); // 设置坐标类型
        option.setScanSpan(1000); // 自动定位间隔
        option.setIsNeedAddress(true); // 是否需要地址
        option.setIsNeedLocationPoiList(true);

        // 定位模式
        option.setLocationMode(LocationClientOption.LocationMode.
        Hight_Accuracy);

        // 根据配置信息对定位客户端进行设置
    }

```

```

        mLocationClient.setLocOption(option);

        // 注册定位监听
        mMyLocationListener = new MyLocationListener();

mLocationClient.registerLocationListener(mMyLocationListener);

        // 设置定位图标
        BitmapDescriptor mCurrentMarker =
        BitmapDescriptorFactory
            .fromResource(R.drawable.location);
        MyLocationConfiguration config = new
        MyLocationConfiguration(
            mCurrentMode, true, mCurrentMarker);
        mBaiduMap.setMyLocationConfiguration(config);
    }

    /**
     * 实现定位回调监听
     */
    public class MyLocationListener implements
    BDLocationListener {
        @Override
        public void onReceiveLocation(BDLocation location) {

            // mapView 销毁后不在处理新接收的位置
            if (location == null || mMapView == null)
                return;

            // 构造定位数据
            MyLocationData locData = new
            MyLocationData.Builder()
                // 此处设置开发者获取到的方向信息，顺时针0-360
                .latitude(location.getLatitude())
                .longitude(location.getLongitude()).build
            ();

            // 设置BaiduMap的定位数据
            mBaiduMap.setMyLocationData(locData);
            // 记录位置信息
            mCurrentLatitude = location.getLatitude();
            mCurrentLongitude = location.getLongitude();

            // 第一次定位时，将地图位置移动到当前位置

```

```

        if (isFristLocation) {
            isFristLocation = false;
            center2myLoc();
        }

        // Log 记录位置信息
        StringBuffer sb = new StringBuffer(256);
        sb.append("\nlatitude : ");
        sb.append(location.getLatitude());
        sb.append("\nlontitude : ");
        sb.append(location.getLongitude());
        sb.append("\naddress : ");
        sb.append(location.getAddrStr());

        for(int i=0; i<location.getPoiList().size(); i++)
        {
            Poi p = location.getPoiList().get(i);
            sb.append("\nPoi NO.");
            sb.append(i);
            sb.append(" : ");
            sb.append(p.getId());
            sb.append("-");
            sb.append(p.getName());
            sb.append("-");
            sb.append(p.getRank());
        }

        Log.i("BaiduLocationInfo", sb.toString());
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.main, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.id_menu_map_myLoc: // 标注覆盖物
            center2myLoc();
            break;

        case R.id.id_menu_map_style: // 地图模式

```

```

        mCurrentStyle = (++mCurrentStyle) %
mStyles.length;
        item.setTitle(mStyles[mCurrentStyle]);

        // 设置自定义图标
        switch (mCurrentStyle)
        {
            case 0:
                mCurrentMode =
MyLocationConfiguration.LocationMode.NORMAL;
                break;
            case 1:
                mCurrentMode =
MyLocationConfiguration.LocationMode.FOLLOWING;
                break;
            case 2:
                mCurrentMode =
MyLocationConfiguration.LocationMode.COMPASS;
                break;
        }

        BitmapDescriptor mCurrentMarker =
BitmapDescriptorFactory
            .fromResource(R.drawable.location);
        MyLocationConfiguration config = new
MyLocationConfiguration(
            mCurrentMode, true, mCurrentMarker);

mBaiduMap.setMyLocationConfiguration(config);
        break;
    }

    return super.onOptionsItemSelected(item);
}

/**
 * BaiduMap 移动到我的位置
 */
private void center2myLoc() {
    LatLng ll = new LatLng(mCurrentLantitude,
mCurrentLongitude);
    // 设置当前定位位置为BaiduMap 的中心点，并移动到定位位置
    MapStatusUpdate u =
MapStatusUpdateFactory.newLatLng(ll);
    mBaiduMap.animateMapStatus(u);
}

```

```

@Override
protected void onStart()
{
    // 开启图层定位
    mBaiduMap.setMyLocationEnabled(true);
    if (!mLocationClient.isStarted())
    {
        mLocationClient.start();
    }

    super.onStart();
}

@Override
protected void onStop()
{
    // 关闭图层定位
    mBaiduMap.setMyLocationEnabled(false);
    mLocationClient.stop();

    super.onStop();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    // 在 activity 执行 onDestroy 时执行
    mMapView.onDestroy(), 实现地图生命周期管理
    mMapView.onDestroy();
}

@Override
protected void onResume() {
    super.onResume();
    // 在 activity 执行 onResume 时执行 mMapView.onResume(),
    实现地图生命周期管理
    mMapView.onResume();
}

@Override
protected void onPause() {
    super.onPause();
    // 在 activity 执行 onPause 时执行 mMapView.onPause(),
    实现地图生命周期管理
    mMapView.onPause();
}

```


}

4.4 实验结论

当编码工作完成后在模拟器或真机中运行项目，效果如下：



图 4.4.1

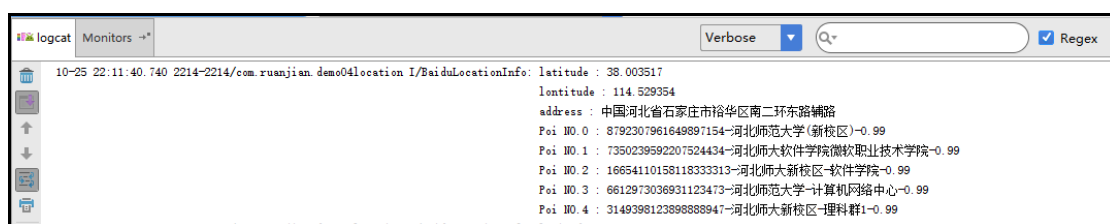


图 4.4.2