

百度地图移动版 API for Android

常见问题参考文档 V1.0



2012.07

法律说明

版权所有百度在线网络技术有限公司。

本文档包含的所有内容除特别声明之外，版权均属于百度在线网络技术有限公司所有，百度可不在作任何声明的情况下对本文档内容进行修改。

本文档中所使用的商标所有权属于该商标的所有者。

文件说明

文件标识:	百度地图移动版 API for Android 常见问题参考文档
当前版本:	2012_07_V1.0
作 者:	Baidu
创建日期:	2012-07-20
最新更新:	2012-07-20

修改说明

修改日期	修订内容

百度在线网络技术有限公司

总机: (+86 10) 59928888

邮箱: mapapi@baidu.com

地址: 北京市海淀区上地信息路 9 号甲奎科科技大厦

邮编: 100085

百度地图网址: <http://map.baidu.com/>

百度地图 API 网址: <http://dev.baidu.com/wiki/static/index.htm>

百度地图移动版 API 网址: <http://dev.baidu.com/wiki/imap/>

欢迎使用百度地图 API !

目 录

1	定位、坐标相关.....	1
1.1	API 如何获取定位信息	1
1.2	如何得到定位的坐标和地址.....	1
1.3	如何让我的地图自动定位.....	2
1.4	如何计算两点之间距离.....	2
1.5	点击 MapView 上一点，获取该点坐标	3
1.6	百度地图地址解析服务 Geocoder 如何使用	4
1.7	百度地图与 google 地图坐标差异	5
2	搜索、覆盖物相关.....	5
2.1	如何获得驾车/步行/公交路线距离.....	5
2.2	POI 查询结果每页显示几个，如何翻页.....	7
2.3	如何设置 POI 检索每页显示容量	7
2.4	路线规划不支持模糊搜索	7
2.5	如何在一个地图上显示多条导航路线.....	8
2.6	搜索结果标注如何自定义点击处理.....	9
2.7	POI 是否支持按分类检索	10
2.8	如何清除覆盖物.....	10
2.9	如何修改标注显示的图片	11
2.10	标注点的气泡内容如何修改.....	11
2.11	添加覆盖物后地图上不显示.....	11
2.12	使用 routeOverlay 出现 crash 情况	11
3	离线地图相关.....	11
3.1	离线地图怎么使用.....	11
3.2	离线地图下载失败怎么办.....	14
4	定位相关.....	15
4.1	通过 Wi-Fi 和移动网络定位得到的数据精度偏差较大.....	15
4.2	如何获得定位的位置信息？（定位返回的信息有哪些？）	15
4.3	持续定位如何关闭.....	16
4.4	支持台湾、香港及其他大陆地区外的定位吗	16
5	地图相关.....	16
5.1	如何控制地图缩放级别.....	16
5.2	支持三维效果的地图吗.....	16
5.3	目前支持多少种地图类型	16
5.4	地图初始显示如何设置中心点	17
5.5	从一个 MapActivity 跳转到另一个 MapActivity 如何实现	17
5.6	MapView 嵌套在 LinearLayout 中，从其他页面切回地图页面时黑屏.....	17
6	其他.....	17
6.1	为何我的百度地图不显示底图.....	17
6.2	总是返回网络错误是怎么回事.....	17
6.3	Key 的申请和 AppName 有关系吗.....	17
6.4	BMapManager 对象初始化几次.....	18

6.5	为何我的程序在不混淆之前可以运行，混淆之后不能运行了	18
6.6	如何使用实时交通信息	18

1 定位、坐标相关

1.1 API 如何获取定位信息

在百度地图移动版 API 中，我们提供一个重要的特色功能：定位，通过这个功能，能获取到用户当前所在位置。在程序中，如果使用此功能，必须注册 GPS 和网络的使用权限。在获取用户位置时，优先使用 GPS 进行定位；如果 GPS 定位没有打开或者没有可用位置信息，则会通过判断是否为 Wi-Fi 连接，如果是，则通过请求百度网络服务，根据 Wi-Fi 热点位置定位；如果否，则通过百度网络服务根据基站信息进行定位。

目前系统自带的网络定位服务精度低，且服务不稳定、精度低，并且从未来的趋势看，基站定位是不可控的（移动公司随时可能更改基站编号以垄断定位服务），而 Wi-Fi 定位则不然，它是一种精度更高、不受管制的定位方法。国内其它使用 Wi-Fi 定位的地图软件，Wi-Fi 定位基本不可用，百度的定位服务量化指标优秀，网络接口返回速度快（服务端每次定位响应时间 50 毫秒以内），精度 280 米，覆盖率 96%，在国内处于一枝独秀的地位。

1.2 如何得到定位的坐标和地址

使用 `MKLocationManager` 类的 `requestLocationUpdates` 注册位置监听事件，在重写的回调函数 `onLocationChanged` 中，可获得定位的坐标（详见 `BMapApiDemoMain` 例程的 `MyLocation.java`），通过调用 `MKSearch` 类的 `reverseGeocode` 接口，得到地址。

```
1.      @Override
2.      public void onLocationChanged( Location location ){
3.          if( location != null ){
4.              pt.setLatitudeE6( ( int ) (location.getLatitude()*1e6));
5.              pt.setLongitudeE6( ( int ) (location.getLongitude()*1e6));
6.              mSearch.reverseGeocode(pt);
7.          }
8.      }
9.      public void onGetAddrResult( MKAddrInfo res, int error ){
10.         if( error != 0 ){
11.             String str = String.format("错误号: %d",error);
12.             return;
13.         }
14.         String strInfo = String.format("纬度: %f 经度: %f 地址: %s\r\n",
15.             res.geoPt.getLatitudeE6() / 1e6,
16.             res.geoPt.getLongitudeE6() / 1e6,
17.             res.addressComponents.city +
18.             res.addressComponents.district +
19.             res.addressComponents.street );
```

```
20.      }
```

1.3 如何让我的地图自动定位

首先在程序中打开相关的权限，然后将 `MyLocationOverlay` 实例添加到 `MapView` 中，并调用 `enableMyLocation` 方法，即可实现当前位置的显示，如果想让地图跟随当前位置移动，需要继承 `MyLocationOverlay` 并实现其 `onLocationChanged` 方法，调用 `MapView.getController().animateTo()` 方法移动地图位置。

1.4 如何计算两点之间距离

路线规划提供了获取路线距离的方法，见 `MKRoutePlan` 类的 `getDistance` 方法。如果是计算任意两点的距离，有两种方法：一种利用勾股定理计算，适用于两点距离很近的情况；一种按标准的球面大圆劣弧长度计算，适用于距离较远的情况。

```
1.      static double DEF_PI = 3.14159265359; // PI
2.      static double DEF_2PI= 6.28318530712; // 2*PI
3.      static double DEF_PI180= 0.01745329252; // PI/180.0
4.      static double DEF_R =6370693.5; // radius of earth
5.      public double GetShortDistance(double lon1, double lat1, double lon2,
        double lat2)
6.      {
7.          double ew1, ns1, ew2, ns2;
8.          double dx, dy, dew;
9.          double distance;
10.         // 角度转换为弧度
11.         ew1 = lon1 * DEF_PI180;
12.         ns1 = lat1 * DEF_PI180;
13.         ew2 = lon2 * DEF_PI180;
14.         ns2 = lat2 * DEF_PI180;
15.         // 经度差
16.         dew = ew1 - ew2;
17.         // 若跨东经和西经 180 度，进行调整
18.         if (dew > DEF_PI)
19.             dew = DEF_2PI - dew;
20.         else if (dew < -DEF_PI)
21.             dew = DEF_2PI + dew;
22.         dx = DEF_R * Math.cos(ns1) * dew; // 东西方向长度 (在纬度圈上的投影长度)
23.         dy = DEF_R * (ns1 - ns2); // 南北方向长度 (在经度圈上的投影长度)
24.         // 勾股定理求斜边长
```

```
25.         distance = Math.sqrt(dx * dx + dy * dy);
26.         return distance;
27.     }
28.     public double GetLongDistance(double lon1, double lat1, double lon2,
double lat2)
29.     {
30.         double ew1, ns1, ew2, ns2;
31.         double distance;
32.         // 角度转换为弧度
33.         ew1 = lon1 * DEF_PI180;
34.         ns1 = lat1 * DEF_PI180;
35.         ew2 = lon2 * DEF_PI180;
36.         ns2 = lat2 * DEF_PI180;
37.         // 求大圆劣弧与球心所夹的角(弧度)
38.         distance = Math.sin(ns1) * Math.sin(ns2) + Math.cos(ns1) * Ma
th.cos(ns2) * Math.cos(ew1 - ew2);
39.         // 调整到[-1..1]范围内, 避免溢出
40.         if (distance > 1.0)
41.             distance = 1.0;
42.         else if (distance < -1.0)
43.             distance = -1.0;
44.         // 求大圆劣弧长度
45.         distance = DEF_R * Math.acos(distance);
46.         return distance;
47.     }
48.     double mLat1 = 39.90923; // point1 纬度
49.     double mLon1 = 116.357428; // point1 经度
50.     double mLat2 = 39.90923; // point2 纬度
51.     double mLon2 = 116.397428; // point2 经度
52.     double distance = GetShortDistance(mLon1, mLat1, mLon2, mLat2);
```

1.5 点击 MapView 上一点, 获取该点坐标

通过继承 MapView 类重写 onTouchEvent 方法来获取点击屏幕的位置, 再通过接口 Projection 的 fromPixels 方法将点击位置转换为该点的地址坐标, 并且将地图空间 bMapView 的类型由 com.baidu.mapapi.MapView 改为子类类型, 本例中为 com.baidu.mapapi.demo.MapViewTest 具体代码如下:

```
1.     MapViewTest mapView=(MapViewTest) findViewById(R.id.bmapView);
2.     class MapViewTest extends MapView
```

```

3.      {
4.          public MapViewTest(Context context)
5.          {
6.              super(context);
7.          }
8.          public MapViewTest(Context context, AttributeSet attrs)
9.          {
10.             super( context, attrs);
11.          }
12.          public MapViewTest(Context context,AttributeSet attrs,int defStyle)
13.          {
14.              super( context, attrs,defStyle);
15.          }
16.          @Override
17.          public boolean onTouchEvent(MotionEvent event)
18.          {
19.              //获得屏幕点击的位置
20.              int x = (int)event.getX();
21.              int y = (int)event.getY();
22.              //将像素坐标转为地址坐标
23.              GeoPoint pt = this.getProjection.fromPixels(x,y);
24.              return super.onTouchEvent(event);
25.          }
26.      }
    
```

1.6 百度地图地址解析服务 Geocoder 如何使用

地址解析服务在 MKSearch 类中有两个接口：reverseGeocode、geocode。
初始化搜索类 MKSearch 并注册结构监听对象 MKSearchListener:

```

1.      //初始化搜索模块，注册事件监听
2.      mSearch = new MKSearch();
3.      mSearch.init(app.mBMapMan, new MySearchListener());
    
```

实现 MySearchListener 的 onGetAddrResult 获取得到的地址信息

```

1.      @Override
2.      public void onGetAddrResult( MKAddrInfo res, int error ){
3.          if( error != 0){
    
```



```
4.         String str = String.format("错误号: %d", error);
5.         return;
6.     }
7.     String strInfo = String.format("纬度: %f 经度: %f 地址: %s\r\n",
8.         res.geoPt.getLatitudeE6() / 1e6,
9.         res.geoPt.getLongitudeE6() / 1e6,
10.        res.addressComponents.city +
11.        res.addressComponents.district +
12.        res.addressComponents.street );
13. }
```

geocode: 根据地址名称获取地址信息:

```
1.     mSearch.geocode("天安门", "北京");
```

reverseGeocode: 根据地理坐标点获取地址:

```
1.     GeoPoint ptCenter = new GeoPoint(39904965, 116327764);
2.     mSearch.reverseGeocode(ptCenter);
```

1.7 百度地图与 google 地图坐标差异

国际经纬度坐标标准为 WGS-84，国内必须至少使用国家测绘局制定的 GCJ-02 对地理位置进行首次加密。百度地图在此基础上，进行了 BD-09 二次加密措施，因此百度地图对外接口的坐标系并不是 GPS 采集的真实经纬度，而是有较大偏移。

2 搜索、覆盖物相关

2.1 如何获得驾车/步行/公交路线距离

在搜索结果的回调函数中，可以返回驾车/步行/公交路线方案，路线距离可根据方案结构数据获得：

驾车路线：

```
1.     public void onGetDrivingRouteResult(MKDrivingRouteResult res,
2.         int error) {
3.         // 错误号可参考 MKEvent 中的定义
4.         if (error != 0 || res == null) {
```

```

5.                Toast.makeText(RoutePlan.this, "抱歉, 未找到结果", Toast.LENGTH_SHORT).show();
6.                return;
7.            }
8.            //获得路线距离
9.            int distance = res.getPlan(0).getRoute(0).getDistance();
10.           RouteOverlay routeOverlay = new RouteOverlay(RoutePlan.this, mMapView);
11.           // 此处仅展示一个方案作为示例
12.           routeOverlay.setData(res.getPlan(0).getRoute(0));
13.           mMapView.getOverlays().clear();
14.           mMapView.getOverlays().add(routeOverlay);
15.           mMapView.invalidate();
16.           mMapView.getController().animateTo(res.getStart().pt);
17.       }
    
```

公交线路:

```

1.       public void onGetTransitRouteResult(MKTransitRouteResult res,
2.                                           int error) {
3.           // 错误号可参考 MKEvent 中的定义
4.           if (error != 0 || res == null) {
5.               Toast.makeText(RoutePlan.this, "抱歉, 未找到结果", Toast.LENGTH_SHORT).show();
6.               return;
7.           }
8.           //获得路线距离
9.           int distance = res.getPlan(0).getDistance();
10.          TransitOverlay routeOverlay = new TransitOverlay(RoutePlan.this, mMapView);
11.          // 此处仅展示一个方案作为示例
12.          routeOverlay.setData(res.getPlan(0).getRoute(0));
13.          mMapView.getOverlays().clear();
14.          mMapView.getOverlays().add(routeOverlay);
15.          mMapView.invalidate();
16.          mMapView.getController().animateTo(res.getStart().pt);
17.      }
    
```

步行路线:

```
1.      public void onGetWalkingRouteResult (MKWalkingRouteResult res,
2.                                          int error) {
3.          // 错误号可参考 MKEvent 中的定义
4.          if (error != 0 || res == null) {
5.              Toast.makeText (RoutePlan.this, "抱歉，未找到结果", Toa
6. st.LENGTH_SHORT).show();
7.          return;
8.      }
9.      // 获得路线距离
10.     int distance = res.getPlan(0).getRoute(0).getDistance();
11.     RouteOverlay routeOverlay = new RouteOverlay (RoutePlan.this,
12. mMapView);
13.     // 此处仅展示一个方案作为示例
14.     routeOverlay.setData (res.getPlan(0).getRoute(0));
15.     mMapView.getOverlays().clear();
16.     mMapView.getOverlays().add (routeOverlay);
17.     mMapView.invalidate();
18.     mMapView.getController().animateTo (res.getStart().pt);
19. }
```

2.2 POI 查询结果每页显示几个，如何翻页

POI 检索结果每页容量默认情况下为 10，翻页功能通过 MKSearch 类的 goToPoiPage 实现，该方法是异步函数，搜索成功后会调用注册的事件处理函数 onGetPoiResult 返回查询页的结果，此接口只针对最后一次 POI 搜索进行翻页检索，即完成一次 POI 搜索后，调用此方法进行翻页。详见常见问题相关程序示例文件夹中的 PoiPageSearch 例程中 PageSearchButtonProcess 方法。

2.3 如何设置 POI 检索每页显示容量

POI 检索结果每页容量可以通过 MKSearch 类的 setPoiPageCapacity 接口进行设置，支持 1-50，默认情况下为 10。也可以通过 getPoiPageCapacity 获得当前 POI 搜索结果的每页容量。

2.4 路线规划不支持模糊搜索

目前不支持模糊搜索，需要输入准确的站名；当起点/终点不是公交站/地铁站时，需要在节点信息中输入地址坐标。

```
1.      //百度大厦坐标
2.      GeoPoint pt1 = new GeoPoint(40057030, 116307845);
3.      //天安门广场
4.      GeoPoint pt2 = new GeoPoint(39912725, 116404008);
5.      MKPlanNode stNode = new MKPlanNode();
6.      stNode.name = "百度大厦";
7.      stNode.pt = pt1;
8.      MKPlanNode enNode = new MKPlanNode();
9.      enNode.name = "天安门广场";
10.     enNode.pt = pt2;
11.     mSearch.transitSearch("北京", stNode, enNode);
```

获取坐标的方法可以通过关键字执行 poi 搜索，得到 poi 信息后取出地址坐标信息，再进行路线规划搜索。

```
1.      mSearch.poiSearchInCity("北京","百度大厦");
2.      public void onGetPoiResult(MKPoiResult res,int type,int error){
3.          if (error != 0 || res == null) {
4.              Toast.makeText(PoiSearch.this, "抱歉，未找到结果", Toast.LENGTH
5.                  _LONG).show();
6.              return;
7.          }
8.          //本例中取第一个 poi 点
9.          GeoPoint pt = res.getPoi(0).pt;
```

2.5 如何在一个地图上显示多条导航路线

以公交线路规划为例：

```
1.      public void onGetTransitRouteResult(MKTransitRouteResult res,
2.          int error) {
3.          if (error != 0 || res == null) {
4.              Toast.makeText(RoutePlan.this, "抱歉，未找到结果", Toa
5.                  st.LENGTH_SHORT).show();
6.              return;
7.          }
8.          int planNum = res.getNumPlan();//获得路线方案个数
```

```
8.          for(int i=0;i<planNum;i++)
9.          {
10.             TransitOverlay routeOverlay = new TransitOverlay (R
outePlan.this, mMapView);
11.             routeOverlay.setData(res.getPlan(i));
12.             mMapView.getOverlays().add(routeOverlay);
13.          }
14.
15.          mMapView.invalidate();
16.          mMapView.getController().animateTo(res.getStart().pt);
17.      }
```

2.6 搜索结果标注如何自定义点击处理

继承 PoiOverlay 类重写 OnTap 方法，具体参考如下代码：

定义 PoiOverlayTest 类继承 PoiOverlay 类，在类中重写 OnTap 方法响应自定义处理，再执行父类的 OnTap 事件

```
1.      class PoiOverItemT extends PoiOverlay {
2.          public PoiOverItemT(Activity activity, MapView mapView) {
3.              super(activity, mapView);
4.          }
5.          @Override
6.          protected boolean onTap(int i) {
7.              //自定义点击处理事件
8.              //
9.              //
10.             //
11.             return super.onTap(i); //响应父类的点击事件
12.          }
13.      }
```

在 Poi 搜索回调函数中定义 PoiOverlayTest 类对象，设置结果数据，并绘制在地图上，这点击 poi 点即可响应自定义的处理事件

```
0.      pubic void onGetPoiResult(MKPoiResult res,int type,int error){
1.          if (error != 0 || res == null) {
```

```
2.                Toast.makeText(PoiSearch.this, "抱歉, 未找到结果", Toast.LENGTH
_LONG).show();
3.                return;
4.            }
5.            if (res.getCurrentNumPois() > 0)
6.            {
7.                //将 poi 结果显示在地图上, PoiOverlayTest 类继承 PoiOverlay 类
8.                PoiOverlayTest poiOverlay = new PoiOverlayTest(PoiSearch.this, mMapView);
9.                poiOverlay.setData(res.getAllPoi());
10.               mMapView.getOverlays().clear();
11.               mMapView.getOverlays().add(poiOverlay);
12.               mMapView.invalidate();
13.            }
14.        }
```

2.7 POI 是否支持按分类检索

支持分类搜索, 例如 `mSearch.poiSearchInCity("北京", "娱乐")`; 得到的搜索结果中是和娱乐相关的地点, 例如保龄球馆、娱乐广场等。

2.8 如何清除覆盖物

通过 `getOverlays` 获得 `MapView` 的所有覆盖物, 并清除。

```
1.        public void onGetPoiResult(MKPoiResult res,int type,int error){
2.            if (error != 0 || res == null) {
3.                Toast.makeText(PoiSearch.this, "抱歉, 未找到结果", Toast.LENGTH
_LONG).show();
4.                return;
5.            }
6.            if (res.getCurrentNumPois() > 0)
7.            {
8.                //将 poi 结果显示在地图上
9.                PoiOverlay poiOverlay = new PoiOverlay(PoiSearch.this,
10.               mMapView);
11.                poiOverlay.setData(res.getAllPoi());
12.                //添加新的 Overlay 之前清除之前的所有 Overlay
13.                mMapView.getOverlays().clear();
14.            }
15.        }
```

```
14.         mMapView.getOverlays().add(poiOverlay);
15.         mMapView.invalidate();
16.     }
17. }
```

2.9 如何修改标注显示的图片

详情参见 BMapApiDemoMain 中的 ItemizedOverlayDemo.java, demo 中 Overlay 使用的标注图片是资源文件夹中的 iconmarka.png, 用户可随意修改使用的图片资源。

2.10 标注点的气泡内容如何修改

Overlay 点击的气泡是通过在应用上层添加一个自定义的 view 实现的, 具体参考 BMapApiDemoMain 例程中的 ItemizedOverlayDemo, 通过继承 ItemizedOverlay<OverlayItem> 实现重写标注点的点击响应事件, 弹出泡泡的内容是通过 mapView.addView 添加了一个 popview, 见 layout 下的 popview.xml, 例程中的 popview 添加的是一个 imageview, 用户可以任意改变 imageview 显示的图片, 或者放置 n 个 imageview 在 popview 中, 也可以修改为 EditText 等控件。

2.11 添加覆盖物后地图上不显示

使用 mMapView.getOverlays().add() 添加 overlay 后, 需要调用 mMapView.invalidate(); 来刷新显示 overlay。

2.12 使用 routeOverlay 出现 crash 情况

确保返回有搜索结果, 并且确保执行 RouteOverlay 类的 setData 方法, 之后再 addOverlay。

3 离线地图相关

3.1 离线地图怎么使用

到官网下载离线地图包, 网址: <http://shouji.baidu.com/map/map.html?from=3052>
首先选择了手机品牌和手机型号:

1 按手机机型下载



选择好手机后，系统会出来提示。请选择旧版本地图进行下载。
因为目前 SDK 只能使用栅格图，所以我们必须下载旧版本的地图。后期 SDK 将支持矢量图，到时候北京市的离线地图就从 114M 降低到 14M 了。

2 选择城市后点击下载

此处离线地图支持 Android V2.0 及以上版本（[点击下载](#)）；V2.0 以下版本 [请下载旧版离线地图包](#)。[如何查看版本号？](#)

比如，选择下载北京市的离线地图：



解压后，有两个文件。一个是 BaiduMap，一个是安装说明。



不要管安装说明，把 BaiduMap 打开，能看到 Mapdata 文件夹。拷贝这个文件夹。



将 Mapdata 文件夹拷贝到手机 SD 卡的 BaiduMapSdk 文件夹根目录下。



找到主 java 文件，并在 activity 里申明变量。

```
1.      MKOfflineMap mOffline = null;    //离线地图变量
```

然后在 onCreate 函数里写离线地图初始化的代码，具体请参考 BMapApiDemoMain 例程中的 OfflineDemo.java:

```
1.      /** 离线地图初始化 */          mOffline = new MKOfflineMap();
2.      mOffline.init(mBMapMan, new MKOfflineMapListener() {
3.          public void onGetOfflineMapState(int type, int state) {
4.              switch (type) {
5.                  case MKOfflineMap.TYPE_DOWNLOAD_UPDATE:
6.                      {
7.                          MKOLUpdateElement update = mOffline.getUpdateInfo(state);
8.                          //mText.setText(String.format("%s : %d%", update.cityName, update.ratio));
9.                      }

```

```
10.             break;
11.             case MKOfflineMap.TYPE_NEW_OFFLINE:
12.                 Log.d("OfflineDemo", String.format("add offlinemap num:%d", state));
13.                 break;
14.             case MKOfflineMap.TYPE_VER_UPDATE:
15.                 Log.d("OfflineDemo", String.format("new offlinemap ver"));
16.                 break;
17.             }
18.         }
19.     }
20. );
21. /** 离线地图导入离线包 */
22. int num = mOffline.scan();
```

导入成功后运行地图，可以看到不联网状态下的地图展示效果：



使用豌豆荚截图 wandoujia.com

3.2 离线地图下载失败怎么办

下载离线地图请确保设备中有 SD 卡，使用 MKOfflineMap 类，在初始化 init 时，会联网下

载相关配置文件，请确保配置文件全部下载完毕后开始离线地图的下载（配置文件位于 SD 卡根目录下的 BaiduMapSDK 文件夹内，共五个 dat 文件），否则会因为找不到配置文件而返回下载失败，之后使用 start 开始下载制定城市 ID 的离线地图。

4 定位相关

4.1 通过 Wi-Fi 和移动网络定位得到的数据精度偏差较大

通常 Wi-Fi 信号覆盖范围在几十米到几百米，通过 Wi-Fi 网络定位的设备距离 Wi-Fi 信号源不范围也就在几十米到几百米，同理，移动网络的覆盖范围在一公里甚至更远，因此可能产生的误差也越大。

4.2 如何获得定位的位置信息？（定位返回的信息有哪些？）

通过 MKLocationManager 注册或者移除定位监听器，监听返回的是 android 系统的 Location 类对象，坐标为百度坐标。具体如下：

```
1.      //获得定位服务
2.      mLocationManager = mBMapMan.getLocationManager();
3.      //设置监听间隔
4.      mLocationManager.setNotifyInterval(10,5);
5.      //定义定位监听事件
6.      LocationListener mLocationListener = new LocationListener()
7.      {
8.          @Override
9.          public void onLocationChanged(Location location)
10.         {
11.             if(location != null)
12.             {
13.                 String strLog = String.format("您当前的位置:\r\n" + "纬度:%f\r\n" + "经度:%f",location.getLongitude(), location.getLatitude());
14.             }
15.         }
16.     }
17.     //注册监听事件
18.     mLocationManager.requestLocationUpdates(mLocationListener);
19.     //不需要定位时移除监听事件
20.     mLocationManager.removeUpdates(mLocationListener);
```

4.3 持续定位如何关闭

使用 `MKLocationManager` 类的 `requestLocationUpdates` 来注册位置监听，用 `removeUpdates` 来取消位置监听。

```
1.     public void stopLoc()
2.     {
3.         BMapApiDemoApp app = (BMapApiDemoApp) this.getApplication();
4.         //移除 listener
5.         app.mBMapMan.getLocationManager().removeUpdates(mLocationListener);
6.     }
7.     public void startLoc()
8.     {
9.         BMapApiDemoApp app = (BMapApiDemoApp) this.getApplication();
10.        // 注册 listener
11.        app.mBMapMan.getLocationManager().requestLocationUpdates(mLocationLi
stener);
12.    }
```

4.4 支持台湾、香港及其他大陆地区外的定位吗

支持港澳台，暂不支持境外。

5 地图相关

5.1 如何控制地图缩放级别

使用 `MapController` 类的 `setZoom` 方法。

Eg: `mapView.getController().setZoom(13)`; 支持缩放级别范围为 3-18

5.2 支持三维效果的地图吗

目前不支持三维地图

5.3 目前支持多少种地图类型

目前的 SDK 底图为栅格图，类型包括普通地图、卫星图、交通流量图

5.4 地图初始显示如何设置中心点

使用 MapController 类的 setCenter 方法。

```
1.      GeoPpoint point = new GeoPoint((int) (31.764*1e6), (int) (119.953*1e6));  
2.      mMapView.getController().setCenter(point);
```

5.5 从一个 MapActivity 跳转到另一个 MapActivity 如何实现

具体请参见常见问题相关程序示例文件夹中 mapActivity 例程里的 PoiSearch.java，在 SearchButtonProcess 方法中实现了向另一个 MapActivity 跳转，只用一个 BMapManager 管理类对象，在进入第二个 MapActivity 时不要重新初始化 BMapManager 管理类对象，只在离开页面时 stop，进入页面时 start。

5.6 Mapview 嵌套在 Linearlayout 中，从其他页面切回地图页面时黑屏

如果在移出 Mapview 时调用了 LinearLayout 的 removeAllViews 方法，需要再切回地图页面时调用 MapActivity 类的 initMapActivity 方法重新加载地图控件，具体请参考常见问题相关程序示例文件夹中的 LinearLayoutMap 例程。

6 其他

6.1 为何我的百度地图不显示底图

检查在程序中是否使用了正确的 Key，有没有打开使用网络连接的权限，以及手机有没有连接互联网。

6.2 总是返回网络错误是怎么回事

首先请确认设备已联网且网络畅通，再次需确认 app 工程中 AndroidManifest.xml 设置是否加入了网络权限<uses-permission android:name="android.permission.INTERNET">

6.3 Key 的申请和 AppName 有关系吗

每个应用申请一个 key，可以用在不同移动平台上的相同应用中。

6.4 BMapManager 对象初始化几次

管理类对象只初始化一次，建议在 Application 里生成 BMapManager 对象并初始化，在程序退出时调用 destory，在需要使用 sdk 功能的 activity 的 onCreate 里调用 start, onDestroy 调用 stop，或者 onResume/onPause 分别调用 start 和 stop。

6.5 为何我的程序在不混淆之前可以运行，混淆之后不能运行了

如果您使用 proguard 进行混淆，请在您的混淆配置文件中添加如下语句：

```
-keep class com.baidu.mapapi.** {*;}
```

其他混淆工具也请进行与此类似的配置，指定对 mapapi 中的类和方法不进行混淆。

6.6 如何使用实时交通信息

目前支持实时交通信息的城市有 11 个：北京，上海，广州，深圳，南京，南昌，成都，重庆，武汉，大连，常州。在地图中查看实时交通信息是很简单的，调用 MapView.setTraffic(true) 即可。方法 isTraffic 返回当前是否显示实时交通信息。

其它问题可参考开发指南，类参考以及示例代码。