MapDemo(地图打点及轨迹回放工具)代码文 档



🅰 **简介**:此文档为MapDemo(地图打点及轨迹回放工具)的代码文档,旨在介绍**项目结构**; 各个不同模块的代码实现;可调整的变量;注意事项等,以方便工具后续维护和新需求的开 发。文档仅涵盖重要的类和文件,部分系统生成的文件或资源文件将被省略。

**工具的使用详情请参考MapDemo使用手册



文件夹 ::

Java类(Class):

AndroidManifest:

布局文件(xml):

其他文件 ::

重要文件位置指引

- libs : 包含高德地图SDK jar包,SO库; mbClientLib-1.0.0.aar。
- app/build.gradle:添加高德地图,腾讯地图,MBClient等依赖。
- AndroidManifest: 配置应用权限;配置高德和腾讯地图的API Key。
- java/com/example/mapdemo/entity: 目录下包含所有自定义的数据类。
 - 。 请确保回放所用Json文件的数据类型和命名与 GpsData 和 DrData 中的相同。
- java/com/example/mapdemo/util/MapTools.java: 工具类。
- java/com/example/mapdemo/MainActivity.java: 地图打点代码实现。
- res/layout/activity_main.xml : 地图打点界面布局。
- java/com/example/mapdemo/TencentReplayActivity.java: 腾讯地图轨 迹回放代码实现。
- res/layout/activity_tencent_replay.xml: 腾讯地图轨迹回放界面布局。

- java/com/example/mapdemo/AmapReplayActivity.java: 高德地图轨迹回 放代码实现。
- res/layout/activity_amap_replay.xml: 高德地图轨迹回放界面布局。

MapDemo

app

build.gradle

• 腾讯地图依赖:

dependencies

- implementation 'com.tencent.map:tencent-map-vectorsdk:6.3.0.250311.940c5007.146962199'
- implementation 'com.tencent.openmap:foundation:0.5.7.fc32fe3'
- 3 implementation 'com.tencent.map:sdk-utilities:1.0.10'

• 高德地图依赖:

代码块

- implementation
 files('libs\\AMap3DMap_10.1.201_AMapSearch_9.7.4_AMapLocation_6.4.9_20250317
 .jar')
- MBClient依赖:

盟博惯导SDK接入文档

版本说明

版本	说明	日期
1.0.0	初始版本	2024.12.30

引用SDK

```
1 implementation files('libs/mb_vdr.aar')
```

接口说明

1. 初始化

```
public static void init(Context context, VdrInitListener listener)
```

VdrInitListener

```
1 public interface VdrInitListener {
2 void onResult(int code, String msg); // 初始化结果,code为错误码,msg为错误信息
3 }
```

2. 添加监听

需要在注册成功后添加

📙 盟博惯导SDK接入文档.pdf

代码块mplementation files('libs/mbClientLib-1.0.0.aar')

• Gson依赖:

代码块

- implementation 'com.google.code.gson:gson:2.8.9'
- mlhfileselector(文件选择器)依赖:

代码块

implementation 'io.github.molihuan:pathselector:1.1.16'

libs

• 以下为高德SDK的jar包和SO库,不能缺失否则地图初始化会白屏。x86和x86_64的SO文件在网上已经找不到资源,官方下载地址解包后只有arm64-v8a和armeabi-v7a。

arm64-v8a

armeabi

armeabi-v7a

x86

x86_64

AMap3DMap_10.1.201_AMapSearch_9.7.4_AMapLocation_6.4.9_20250317.jar

mbClientLib-1.0.0.aar

src/main

AndroidManifest

```
AndroidManifest.xml
```

- 1 <!--允许访问网络,必选权限-->
- 2 <uses-permission android:name="android.permission.INTERNET" />

3

```
<!--允许获取粗略位置,若用GPS实现定位小蓝点功能则必选-->
    <uses-permission android:name="android.permission.ACCESS COARSE LOCATION" />
 5
 6
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
 7
 8
    <!--允许获取网络状态,用于网络定位,若无gps但仍需实现定位小蓝点功能则此权限必选-->
9
    <uses-permission android:name="android.permission.ACCESS NETWORK STATE" />
10
11
    <!--允许获取wifi网络信息,用于网络定位,若无gps但仍需实现定位小蓝点功能则此权限必选-->
12
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
13
14
    <!--允许获取wifi状态改变,用于网络定位,若无gps但仍需实现定位小蓝点功能则此权限必选-->
15
    <uses-permission android:name="android.permission.CHANGE WIFI STATE" />
16
17
    <!--允许写入扩展存储,用于数据缓存,若无此权限则写到私有目录-->
18
19
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
20
    <!--允许读设备等信息,用于问题排查-->
21
22
    <uses-permission android:name="android.permission.READ EXTERNAL STORAGE" />
23
    <!-- 允许访问文件 -->
24
    <uses-permission android:name="android.permission.MANAGE EXTERNAL STORAGE" />
25
26
27
    <uses-permission android:name="android.permission.ACCESS BACKGROUND LOCATION"</pre>
    />
```

• ★配置腾讯高德API Key:

```
AndroidMenifest.xml
```

- 1 <meta-data
- 2 android:name="TencentMapSDK"
- 3 android:value="VCNBZ-7CK65-I5YIG-IVENL-Z43ZK-WKBZQ"/> // 替换api key

• ★配置高德地图API Key:

AndroidMenifest.xml

- 1 <meta-data
- 2 android:name="com.amap.api.v2.apikey"
- 3 android:value="e838669f66499aec9c5c8fa4550b6b4c"/> // 替换api key

java

MainActivity

- 此类为地图打点工具的具体代码实现,仅介绍部分关键代码
- 可配置变量

```
MainActivity.java

1 // 以下两个变量控制打点抽稀的阈值,可根据需要调整

2 private static final long UPDATE_INTERVAL = 3000; // 打点最小时间间隔(毫秒)

3 private static final double MIN_DISTANCE_DELTA = 2.0; // 最小移动距离(米
```

- 主要方法
 - 。 服务初始化相关:

```
MainActivity.java
    @Override
 2
    protected void onCreate(Bundle savedInstanceState) {
 3
 4
        // 进行高德地图的隐私政策合规检查,必要
 5
        MapsInitializer.updatePrivacyShow(this, true, true);
 6
        MapsInitializer.updatePrivacyAgree(this, true);
 7
 8
        // 设置需要请求的权限,必要
 9
        if (Build.VERSION. SDK_INT > 28 &&
10
     getApplicationContext().getApplicationInfo().targetSdkVersion > 28) {
            needPermissions = new String[]{
11
12
                     Manifest.permission.ACCESS_COARSE_LOCATION,
                     Manifest.permission.ACCESS_FINE_LOCATION,
13
                     BACK_LOCATION_PERMISSION
14
15
            };
16
            needCheckBackLocation = true;
        }
17
18
        // 初始化盟博VDR
19
        MBServiceInstance.init()
20
21
22
        // 初始化传感器
        sensorManager = (SensorManager) getSystemService(SENSOR_SERVICE);
23
24
        Sensor accelerometer =
    sensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
         Sensor gyroscope =
25
     sensorManager.getDefaultSensor(Sensor.TYPE_GYROSCOPE);
```

```
26
       Sensor temperature =
    sensorManager.getDefaultSensor(Sensor.TYPE AMBIENT TEMPERATURE);
        sensorManager.registerListener(sensorEventListener, accelerometer,
27
    SENSOR UPDATE INTERVAL);
       sensorManager.registerListener(sensorEventListener, gyroscope,
28
    SENSOR UPDATE INTERVAL);
29
       sensorManager.registerListener(sensorEventListener, temperature,
    SENSOR_UPDATE_INTERVAL);
30
31
       . . . . . .
32
    }
33
    /**
34
    * 初始化原生NMEA信号监听
35
36
37
    private void initNmeaListener()
38
39
    /**
40
    * 初始化原生GPS监听
41
42
    private void initGpsListener()
43
    /**
44
    * 初始化高德地图和定位蓝点
45
46
    private void amapInit(Bundle savedInstanceState)
47
48
49
    private void amapMyLocatoinInit() {
50
       . . . . . .
51
       // 设置定位蓝点的定位模式
52
       // 只定位一次: MyLocationStyle.LOCATION_TYPE_SHOW
53
       // 定位一次,且将视角移动到地图中心点: MyLocationStyle.LOCATION_TYPE_LOCATE
54
       // 连续定位、且将视角移动到地图中心点,定位蓝点跟随设备移动:
55
    MyLocationStyle.LOCATION_TYPE_FOLLOW
56
       -// 连续定位、且将视角移动到地图中心点,地图依照设备方向旋转,定位点会跟随设备移
    动: MyLocationStyle.LOCATION_TYPE_MAP_ROTATE
       // 连续定位、且将视角移动到地图中心点,定位点依照设备方向旋转,并且会跟随设备移
57
    动: MyLocationStyle.LOCATION TYPE LOCATION ROTATE
       // 连续定位、蓝点不会移动到地图中心点,定位点依照设备方向旋转,并且蓝点会跟随设备
58
    移动: MyLocationStyle.LOCATION TYPE LOCATION ROTATE NO CENTE
       // 连续定位、蓝点不会移动到地图中心点,并且蓝点会跟随设备移动:
59
    MyLocationStyle.LOCATION TYPE FOLLOW NO CENTER
       // 连续定位、蓝点不会移动到地图中心点,地图依照设备方向旋转,并且蓝点会跟随设备移
60
    动: MyLocationStyle.LOCATION_TYPE_MAP_ROTATE_NO_CENTER
61
    myLocationStyle.myLocationType(MyLocationStyle.LOCATION_TYPE_LOCATION_ROTATE
```

```
_NO_CENTER)
62
       //设置连续定位模式下的定位间隔,只在连续定位模式下生效,单次定位模式下不会生效。单
63
    位为毫秒。
       myLocationStyle.interval(UPDATE_INTERVAL);
64
65
       // 注册定位蓝点移动监听,在信号透出时打点
66
       // 注: 高德地图透出的经纬度坐标系为gcj02
67
68
       aMap.addOnMyLocationChangeListener();
69
70
       . . . . . .
    }
71
72
73
    /**
  * 注册惯导监听,在信号透出时打点
74
    * 注: 盟博惯导透出的经纬度坐标系为wgs84,需使用
75
    * convertVdrLatLonStrToDecimal(String coordinate, String latOrLon) 和
76
77
    * MapTools.wgs84ToGcj02(double lat, double lon) 进行坐标系转换才能正常打点
78
    */
79
    private void addOnVdrListener()
80
    /**
81
    * 传感器监听,在初始化传感器时注册,用于接收并保存传感器信号数据
82
83
84
    SensorEventListener sensorEventListener = new SensorEventListener() {
85
       @Override
       public void onSensorChanged(SensorEvent event) {
86
87
       }
88
    }
89
```

。 数据保存相关

```
MainActivity.java
1 /**
2
   * 保存GPS数据
3
4
   private void saveGpsData() {
5
        . . . . . .
6
7
        // 定义GPS数据文件保存路径(文件名称和保存路径可以在此修改)
        File dir = getApplicationContext().getFilesDir();
8
        File file = new File(dir, "gps_data.json");
9
10
```

```
11 .....
12
   }
13
14 /**
15 * 保存惯导数据
16 */
17 private void saveDrData() {
18
      . . . . .
19
       // 定义DR数据文件保存路径 (文件名称和保存路径可以在此修改)
20
       File dir = getApplicationContext().getFilesDir();
21
       File file = new File(dir, "dr_data.json");
22
23
24
      . . . . . .
25 }
```

• 参考文档

- 。 高德地图开发文档
- 。 盟博惯导SDK接入文档

盟博惯导SDK接入文档

版本说明

版本	说明	日期
1.0.0	初始版本	2024.12.30

引用SDK

```
1 implementation files('libs/mb_vdr.aar')
```

接口说明

1. 初始化

```
1 public static void init(Context context, VdrInitListener listener)
```

VdrInitListener

```
1 public interface VdrInitListener {
2  void onResult(int code, String msg); // 初始化结果, code为错误码, msg为错误信息
3 }
```

2. 添加监听

需要在注册成功后添加

Gson

TencentReplayActivity

- 此类为腾讯地图轨迹回放的具体代码实现,仅介绍部分关键代码
- 可配置变量

```
TencentReplayActivity.java
1 /**
   * 倍速乘数,默认倍速为1.0,即原速播放
    * 界面中提供倍速选择,修改此参数以更改界面初始化时的默认倍速
   */
4
5
   private float speedMultiplier = 1.0f;
6
7 /**
   * 是否启用惯导回放,默认关闭
8
   * 界面中提供惯导回放开关,修改此参数以更改界面初始化时是否开启惯导回放
9
10
   */
   private boolean isVdrEnable = false;
11
```

• 主要方法

。 初始化相关

```
TencentReplayActivity.java
    @Override
1
2
    protected void onCreate(Bundle savedInstanceState) {
        // 进行腾讯地图的隐私政策合规检查,必要
3
       TencentMapInitializer.setAgreePrivacy(this, true);
4
5
       TencentMapInitializer.start(this);
6
       // 显示缩放控件
7
8
       tencentMap.getUiSettings().setZoomControlsEnabled(true);
9
        // 加载保存的轨迹数据,可在此更改初始化界面时读取的文件
10
       loadGpsData("gps_data.json");
11
12
    }
13
    /**
14
   * 初始化文件选择器
15
16
17
    private void initPathSelector(int selectorCode) {
        // 关闭自动申请权限弹窗,否则即使已开启权限仍会弹出
18
        PathSelectorConfig.setAutoGetPermission(false);
19
20
```

```
21
        // 获取PathSelectFragment实例然后在onBackPressed中处理返回按钮点击事件
        // 各种初始化配置的使用方法请移步下方参考文档中的 mlhfileselector
22
        PathSelectFragment selector = PathSelector.build(this,
23
    MConstants. BUILD DIALOG)
                //.setBuildType(MConstants.BUILD DIALOG)//已经在build中已经设置了
24
                //.setContext(this)//已经在build中已经设置了
25
                .setRootPath(getApplicationContext().getFilesDir().toString())//
26
    初始路径
27
                .setShowSelectStorageBtn(false)//是否显示内部存储选择按钮
                .setShowTitlebarFragment(true)//是否显示标题栏
28
                .setShowTabbarFragment(true)//是否显示面包屑
29
                .setShowFileTypes("json")//只显示(后缀为json)的文件
30
                .setSelectFileTypes("ison")//只能选择(后缀为ison)的文件
31
                .setMaxCount(\frac{1}{2})//最多可以选择1个文件,默认是-1不限制
32
                .setRadio()//单选(如果需要单选文件夹请使用setMaxCount(0)来替换)
33
                .setSortType(MConstants.SortRules.SORT_NAME_ASC)//按名称排序
34
                .setTitlebarMainTitle(new FontBean("请选择JSON轨迹数据文件"))//设置
35
    标题栏主标题,还可以设置字体大小,颜色等
                .setTitlebarBG(Color.CYAN)//设置标题栏颜色
36
                .setPathSelectDialogHeight(mapView.getHeight())
37
                .setPathSelectDialogWidth(mapView.getWidth())
38
                .setFileItemListener(//设置文件item点击回调(点击是文件才会回调,如果点
39
    击是文件夹则不会)
40
                       new FileItemListener() {
                           @Override
41
                           public boolean onClick(View v, FileBean file,
42
    String currentPath, BasePathSelectFragment pathSelectFragment) {
                               if (selectorCode == GPS FILE SELECTOR CODE) {
43
                                  loadGpsData(file.getName());
44
                               } else if (selectorCode ==
45
    DR_FILE_SELECTOR_CODE) {
                                  loadDrData(file.getName());
46
                                  Log.d("PathSelector", "Dr File loaded: " +
47
    file.getName());
48
                               }
49
                               pathSelectFragment.close();
                               return false;
50
51
                           }
                       }
52
53
                )
54
                .show();
55
56
        . . . . . .
57
    }
58
59
     * 初始化倍速下拉框
60
```

```
61
      */
     private void initSpeedSpinner() {
62
63
         . . . . . .
64
         // 注册倍速下拉框的点击监听
65
         // 下拉框的选项可以在res/values/strings.xml中找到@array/speed array进行编辑
66
         // 配置各选项的倍速时请注意speedMultiplier是个乘数,因此倍速越高,其数值应该越小
67
         // 例: 2倍速 -> speedMultiplier = 0.5f; 10倍速 -> speedMultiplier = 0.1f;
68
     以此类推
69
         speedSpinner.setOnItemSelectedListener(new
     AdapterView.OnItemSelectedListener() {
            @Override
70
71
            public void onItemSelected(AdapterView<?> adapterView, View view,
     int i, long l) {
                switch (i) {
72
73
                    case 0:
74
                        speedMultiplier = 1.0f;
75
                        Toast.makeText(view.getContext(), "已重置回放并切换为1倍
     速,点击'开始'按钮进行回放", Toast. LENGTH_SHORT).show();
                        break;
76
77
                    case 1:
                        speedMultiplier = 0.5f;
78
                        Toast.makeText(view.getContext(), "已重置回放并切换为2倍
79
     速,点击'开始'按钮进行回放", Toast. LENGTH_SHORT).show();
80
                        break;
81
                    case 2:
                        speedMultiplier = 0.25f;
82
                        Toast.makeText(view.getContext(), "已重置回放并切换为4倍
83
     速,点击'开始'按钮进行回放", Toast.LENGTH_SHORT).show();
                        break;
84
85
                    case 3:
                        speedMultiplier = 0.1f;
86
                        Toast.makeText(view.getContext(), "已重置回放并切换为10倍
87
     速,点击'开始'按钮进行回放", Toast. LENGTH_SHORT).show();
88
                        break;
89
                }
90
91
                resetReplay();
        }
92
93
94
         . . . . . .
95
     }
96
     /**
97
      * 初始化启用惯导回放开关
98
99
100
     private void initVdrEnableButton() {
```

```
101
         . . . . . .
102
         // 注册开关监听
103
         vdrEnableSwitch.setOnCheckedChangeListener(new
104
     CompoundButton.OnCheckedChangeListener() {
105
            @Override
106
            public void onCheckedChanged(CompoundButton buttonView, boolean
     isChecked) {
107
                if (isChecked){
108
                    if (!drDataList.isEmpty()) { // Case 1: 惯导文件已读取,直接启
     用回放
                       Toast.makeText(mapView.getContext(),"已启用惯导回放",
109
     Toast. LENGTH_SHORT).show();
                       isVdrEnable = true;
110
                    } else if (loadDrData("dr_data.json")) { // Case 2: 读取默认惯
111
     导文件并启用回放,可以在此修改文件名
                       Toast.makeText(mapView.getContext(),"已读取默认惯导文件并启
112
     用惯导回放", Toast. LENGTH_SHORT).show();
113
                       isVdrEnable = true;
                    } else { // Case 3: 读取失败,不启用回放
114
115
                       Toast.makeText(mapView.getContext(),"无法读取默认惯导文件,
     请手动选择后再尝试启用惯导回放", Toast. LENGTH SHORT).show();
                       vdrEnableSwitch.setChecked(false);
116
117
                    }
118
                }else {
                    Toast.makeText(mapView.getContext(),"已关闭惯导回放",
119
     Toast.LENGTH_SHORT).show();
                    isVdrEnable = false;
120
                }
121
122
123
                resetReplay();
            }
124
        });
125
126
     }
127
128
    /**
     * 初始化Gps回放
129
      * 画线和小车平滑移动的初始化代码详解请移步下方参考文档中的腾讯地图开发文档
130
131
     private void initGpsReplay()
132
133
134
     /**
     * 初始化惯导回放
135
     * 画线和小车平滑移动的初始化代码详情请移步下方参考文档中的腾讯地图开发文档
136
137
      */
138
     private void initDrReplay()
139
```

• 回放控制相关

• 数据读写相关

```
TencentReplayActivity.java
1 /**
2 * 加载GPS回放数据
4 private boolean loadGpsData(String fileName) {
5
     . . . . . .
6
       // 设置轨迹总时长,秒数为dataList的轨迹点总数,乘以1000ms/s因为腾讯地图平滑移动
7
    入参以ms为单位
       gpsTotalDuration = dataList.size() * 1000;
8
9
10
    . . . . .
11 }
12
13 /**
14 * 加载惯导回放数据
15
   */
    private boolean loadDrData(String fileName) {
16
```

```
17
        . . . . . .
18
        // 设置轨迹总时长,秒数为dataList的轨迹点总数,乘以1000ms/s因为腾讯地图平滑以后
19
    入参以ms为单位
        drTotalDuration = dataList.size() * 1000;
20
21
22
        . . . . . .
23
    }
24
25
    /**
    * 保存裁剪后的GPS轨迹数据
26
27
    private void saveTrimmedGpsData() {
28
29
        . . . . . .
30
        // 定义文件保存路径,可在此修改裁剪后GPS轨迹数据的路径和文件名
31
        File dir = getApplicationContext().getFilesDir();
32
33
        File file = new File(dir, "gps_data_trimmed.json");
34
35
        . . . . . .
36
    }
37
   /**
38
39
    * 保存裁剪后的惯导轨迹数据
40
    private void saveTrimmedDrData() {
41
42
        . . . . . .
43
        // 定义文件保存路径,可在此修改裁剪后惯导轨迹数据的路径和文件名
44
        File dir = getApplicationContext().getFilesDir();
45
        File file = new File(dir, "dr_data_trimmed.json");
46
47
48
        . . . . . .
49
    }
```

参考文档

- 。 腾讯地图开发文档
- 。 mlhfileselector (文件选择器)
- 。 双滑块拖动条
- Gson

AmapReplayActivity

此类为高德地图轨迹回放的具体代码实现。由于 AmapReplayActivity 和
TencentReplayActivity 在代码结构上高度相似,仅有部分涉及高德地图SDK画线和平滑移动的位置有些许不同,因此不做过多赘述。请直接参考高德地图开发文档和以上对
TencentReplayAcitivity 的代码介绍。

entity

DrData

• 用于存储**Dr(惯导**)轨迹数据的数据类型,如需导入外部Json文件进行轨迹回放,请确保包含此数据类型中回放所需的所有 attributes (标注为"必要"),并且命名也需保持一致。具体如下:

```
DrData.java
    public double latitude = -1; // gcj02, 必要
    public double longitude = -1; // gcj02, 必要
 2
    public double alt;
 3
    public float yaw; // 必要
 4
    public float speed;
 5
   public long timestamp;
 6
    public String utcDate;
 7
    public String utcTime;
 8
    public int accValidAxes;
 9
    public int accInterval;
10
    public double accYawRate;
11
    public double accPitchRate;
12
    public double accRollRate;
13
    public int gyroValidAxes;
14
    public int gyroInterval;
15
16
    public double gyroYawRate;
    public double gyroPitchRate;
17
18
    public double gyroRollRate;
    public double temperature;
19
```

①:如果经纬度命名不一致(如lat和lon)且文档过大无法通过AI进行效率转换,可以<u>Refactor</u>数据类中相关变量的命名,但数据类型必须一致。

• Json文件示例:

```
dr_data.json

1 [
2 {
3 "accInterval": 100,
```

```
4
         "accPitchRate": -9.698094367980957,
 5
         "accRollRate": 0.3184020221233368,
         "accValidAxes": 7,
 6
         "accYawRate": -0.2465820014476776,
 7
         "alt": 19.642266305098637,
 8
 9
         "gyroInterval": 100,
         "gyroPitchRate": -0.011038999073207378,
10
         "gyroRollRate": 0.009575999341905117,
11
12
         "gyroValidAxes": 7,
         "gyroYawRate": 3.989999822806567E-4,
13
14
         "latitude": 24.46049648884436,
         "longitude": 118.17800947183106,
15
         "speed": 17.16275,
16
         "temperature": 47.5,
17
18
         "timestamp": 538962,
         "utcDate": "180625",
19
         "utcTime": "134841.258",
20
21
         "yaw": 9.724103
22
       },
23
       {
24
         "accInterval": 100,
25
         "accPitchRate": -10.468962669372559,
         "accRollRate": -0.3184020221233368,
26
         "accValidAxes": 7,
27
         "accYawRate": 0.05027399957180023,
28
         "alt": 19.291250838995193,
29
         "gyroInterval": 100,
30
         "gyroPitchRate": 0.002526999916881323,
31
         "gyroRollRate": -0.035909999161958694,
32
         "gyroValidAxes": 7,
33
34
         "gyroYawRate": 0.047214996069669724,
         "latitude": 24.460651106544034,
35
         "longitude": 118.1780385593718,
36
         "speed": 17.166649,
37
38
         "temperature": 47.5,
39
         "timestamp": 539975,
40
         "utcDate": "180625",
         "utcTime": "134842.268",
41
         "yaw": 9.368404
42
       }
43
44
    1
```

GgaEntity

• GGA信号数据类,用于存储信号中解析出的数据信息,包含的 attributes 如下:

```
GgaEntity.java

1  String time;
2  double lat; // WGS84
3  String latDir;
4  double lon; // WGS84
5  String lonDir;
6  String quality;
7  String satellites;
8  String hdop;
9  double alt; // WGS84
10  String geoid;
```

GpsData

• 用于存储GPS轨迹数据的数据类型,如需导入外部Json文件进行轨迹回放,请确保包含此数据类型中回放所需的所有 attributes (标注为"必要"),并且命名也需保持一致。具体如下:

```
GpsData.java

1 public double latitude = -1; // GCJ02, 必要

2 public double longitude = -1; // GCJ02, 必要

3 public double alt;

4 public double yaw; // 必要

5 public double speed;

6 public String utcDateAndTime;

7 public long timestamp;
```

①:如果经纬度命名不一致(如lat和lon)且文档过大无法通过AI进行效率转换,可以<u>Refactor</u>数据类中相关变量的命名,但数据类型必须一致。

• Json文件示例:

```
代码块
1
    2
    {
        "alt": 29.9,
3
4
        "course": -1.0,
        "latitude": 24.47270549715084,
 5
        "longitude": 118.18261041732323,
 6
 7
        "speed": -1.852,
        "timestamp": 116572,
8
       "utcDateAndTime": "2025-06-17 03:36:16"
9
10
      },
```

```
11
         "alt": 20.3,
12
         "course": 22.47,
13
         "latitude": 24.472834179755406,
14
         "longitude": 118.18267925374737,
15
         "speed": 0.0,
16
         "timestamp": 117606,
17
         "utcDateAndTime": "2025-06-17 03:36:17"
18
19
       },
20
       {
         "alt": 18.1,
21
22
         "course": 24.68,
         "latitude": 24.472993345828915,
23
         "longitude": 118.18275424359405,
24
         "speed": 71.279776,
25
         "timestamp": 118627,
26
         "utcDateAndTime": "2025-06-17 03:36:18"
27
28
      }
29
    1
```

RmcEntity

• RMC信号数据类,用于存储信号中解析出的数据信息,包含的 attributes 如下:

```
RmcEntity.java

1  public String time;
2  public String status;
3  public String latStr;
4  public String latDir;
5  public String lonStr;
6  public String lonDir;
7  public double speed;
8  public double course;
9  public String utcDateAndTime;
```

util

MapTools

地图工具类,包含数学计算;经纬度转换;NMEA信号解析等方法,各方法详情如下:

```
MapTools.java

1 /**
```

```
* WGS84转GCJ02
3
     * @param lat WGS84纬度
 5
    * @param lon WGS84经度
     * @return GCJ02坐标
 6
    */
7
    public static double[] wgs84ToGcj02(double lat, double lon)
8
9
   /**
10
    * 判断是否在国内
11
12
  * @param lat 纬度
13
     * @param lon 经度
14
     * @return 是否在国内
15
     */
16
17
    private static boolean outOfChina(double lat, double lon)
18
19
    /**
20
  * 计算纬度偏移量
21
22
    * @param x x坐标
    * @param y y坐标
23
    * @return 纬度偏移量
24
25
     */
    private static double transformLat(double x, double y)
26
27
28
   /**
    * 计算经度偏移量
29
30
     * @param x x坐标
31
     * @param y y坐标
32
    * @return 经度偏移量
33
    */
34
    private static double transformLon(double x, double y)
35
36
37
   /**
    * 将VDR格式的经纬度字符串转换为十进制小数
38
     * @param coordinate 坐标字符串
39
     * @param latOrLon 经纬度识别字符串
40
     * @return
41
42
     */
    public static double convertLatLonStrToDecimal(String coordinate, String
43
    lat0rLon)
44
   /**
45
  * 计算两点间的距离
46
    * @param lat1
47
```

```
48
     * @param lon1
     * @param lat2
49
     * @param lon2
50
     * @return
51
     */
52
    public static double calculateDistance(double lat1, double lon1, double lat2,
53
    double lon2)
54
    /**
55
    * 计算UTC时间
56
     * @param timeStr
57
    * @param dateStr
58
     * @return
59
    */
60
    public static String calculateUTCTime(String timeStr, String dateStr)
61
```

view

DoubleSeekBar

- 双滑块拖动条。
- Credit: 此类非原创,代码原型和出处详见此文章。

res/layout

activity_amap_replay.xml

• 高德地图轨迹回放布局文件。

activity_main.xml

• 地图打点工具布局文件。

activity_tencent_replay.xml

• 腾讯地图轨迹回放布局文件。