

关于flutter-geolocator库使用问题，解决android使用LocationManager定位问题

原创

Super-Bin

已于 2022-07-07 11:41:39 修改

4119

收藏 5

版权

分类专栏: Flutter Android 文章标签: webview flutter android

Flutter Android

同时被 2 个专栏收录

0 订阅 17 篇文章 订阅专栏

文章目录

- 最新改动
- 相关知识
- 问题
- 现象
- 分析问题
- 解决问题
 - 修改源码
- 使用方式
- 2022.5.5更新
 - 解决方案1:本地引入
 - 解决方案2:远程依赖
- 2022.7.6更新

有需要直接定位到最下面。

- android中几种定位方式详解

最新改动

2022年7月7日更新，如果getCurrentPosition获取不到数据，可以尝试先调用getLastKnownPosition。
2022年1月28日更新，geolocator官方库在v8+的版本中适配了android12，并且执行方法有变动，[基于源码v8.0.3修改](#)，分支geolocator_v8.0.3_20220121，引入和修改源码位置可以参考以下内容。

相关知识

- geolocator flutter一个比较好用的定位库
- android原生提供的LocationManager，常用的定位库
- 谷歌定位服务Google Location Service，需要引入google服务已经具备定位权限，并用户同意授权。

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

问题

- <https://github.com/Baseflow/flutter-geolocator/issues/117>

根据官方例子，调用方式

```
1 | import 'package:geolocator/geolocator.dart';
2 |
3 | Position position = await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.highest);
```

在使用flutter-geolocator遇到的问题。有的手机获取定位很快，有的手机半天没有定位结果，定位很慢。

现象

查看flutter-geolocator源码，发现内部使用了两种定位方式。一种是android原生提供的LocationManager，另一种是谷歌定位服务Google Location Service，需要引入google服务。
1.在使用Google Location Service的方式获取定位，没有翻墙，没有回调。
2.在使用原生LocationManager的方式获取定位，有两种情况，一种是网络定位、一种是GPS定位。使用网络定位，但没有连接网络，没回调。使用PGS定位，在室内测试没回调。

分析问题

在Flutter调用Geolocator.getCurrentPosition方法

```
1 | Position position = await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.HIGH);
```

会进入到原生，具体使用哪个方式来获取定位，从这里可以知道

1. 设置属性forceAndroidLocationManager为true，则会使用android原生的LocationManager获取定位信息。
forceAndroidLocationManager为false，则会判断google服务是否可用
2. google服务可用，则会使用Google Location Service的方式，这里就是FusedLocationClient封装了方法。
3. google服务不可用，也会使用android原生的LocationManager获取定位信息。

```
1 | public LocationClient createLocationClient(  
2 |     Context context,  
3 |     boolean forceAndroidLocationManager,  
4 |     @Nullable LocationOptions locationOptions) {  
5 |     if (forceAndroidLocationManager) {  
6 |         return new LocationManagerClient(context, locationOptions);  
7 |     }  
8 |  
9 |     return isGooglePlayServicesAvailable(context)  
10 |        ? new FusedLocationClient(context, locationOptions)  
11 |        : new LocationManagerClient(context, locationOptions);  
12 | }
```

解决问题

为了在国内外都能正常使用，就不能依赖google服务，所以这里我们打将forceAndroidLocationManager属性设置为true。

```
1 | Geolocator.getCurrentPosition(forceAndroidLocationManager: true)
```

这样就会使用Android原生的LocationManager定位，但又有新的问题，上面说了，LocationManager有两种情况，一种是网络定位、一种是GPS定位。使用GPS的话，在室内会非常慢，这肯定是不太想要的，所以我们只要使用网络定位的方式就可以很好的解决问题。

定位到LocationManagerClient类，getBestProvider方法。发现代码的实现方式，先通过locationManager.getProviders获取支持的定位方式。

```
1 | if (Strings.isEmptyOrWhitespace(provider)) {  
2 |     List<String> providers = locationManager.getProviders(true);  
3 |     if (providers.size() > 0) provider = providers.get(0);  
4 | }
```

官方提供了几种Provider：

1. LocationManager.NETWORK_PROVIDER，网络定位，网络定位则更具有实时性，在精度要求不高以及室内
2. LocationManager.GPS_PROVIDER，GPS定位，首次采集数据较慢，定位精度高
3. LocationManager.PASSIVE_PROVIDER，被动定位，被动定位并不会做任何获取位置信息的尝试，它只是被动的接收位置信息的更新，只有其他应用使用了网络定位或GPS定位获取到了新的位置信息后，被动定位的监听者才能获取当前位置。

修改源码

官方实现方式是直接获取第一个定位方式，如果这个方式刚好是GPS定位，那在室内调试的时候，半天获取不到数据。只要改成，都使用网络定位的方式就好了。

```
1 | Location currentLocation = locationManager.getLastKnownLocation(  
2 |     List<String> providers = locationManager.getProviders(true);  
3 |  
4 |     if (providers.contains(LocationManager.NETWORK_PROVIDER)) {  
5 |         provider = LocationManager.NETWORK_PROVIDER;  
6 |     } else if (providers.contains(LocationManager.GPS_PROVIDER)) {  
7 |         provider = LocationManager.GPS_PROVIDER;  
8 |     }
```

如果存在NETWORK_PROVIDER，就使用网络定位的方式。这样就解决问题了。

再进行一些其他优化，源码中onLocationChanged回调函数，是获取定位的回调，里面还进行了一些误差范围的限制，由于使用网络定位，误差可能会比原本的限制值偏大，所以也进行了一点微调。在精确度中，加入一个lowest: 1000的值。

```
1 | private static float accuracyToFloat(LocationAccuracy accuracy) {
```

```
2     switch (accuracy) {
3         case lowest:
4             return 1000;
5         case low:
6             return 500;
7         case medium:
8             return 250;
9         case best:
10        case bestForNavigation:
11            return 50;
12        default:
13            return 100;
14    }
15 }
```

使用方式

- [基于源码v7.0.3修改](#) 分支geolocator_v7.0.3_zzb_0421

pubspec.yaml

```
1 dependencies:
2   flutter-geolocator:
3     git:
4       url: "https://github.com/Super-Bin/flutter-geolocator"
5       r e f: geolocator_v7.0.3_zzb_0421
```

使用兼容android12的版本- [基于源码v8.0.3修改](#) 分支geolocator_v8.0.3_20220121

pubspec.yaml

```
dependencies:
  flutter-geolocator:
    git:
      url: "https://github.com/Super-Bin/flutter-geolocator"
      r e f: geolocator_v8.0.3_20220121
```

flutter调用

```
1
2
```

最好先把代码拉下来，在demo运行验证通过了再集成到项目中。

2022.5.5更新

最近在一个新项目中按照以上方式引入，发现修改的代码没有生效，我明明改了android源码。原因是在v8.0库开始，插件分类管理，分类引入，在引入库的配置上有问题。

主入口geolocator中的pubspec.yaml文件可以看到，统一接口引入了geolocator_platform_interface库，android相关功能使用geolocator_android库，ios相关功能使用geolocator_apple库。但是文章介绍修改源码的地方都是在geolocator_android中完成，然而在主入口geolocator中的pubspec.yaml，依旧引入geolocator_android: ^3.0.2，这个表示从pub仓库下载geolocator_android库，所以引用的依然是pub仓库中原来的代码，我修改的源码没有生效。

```
1 |
2 |
```

解决方案1:本地引入

既然修改的是geolocator_android源码，那就整个geolocator都使用本地引入的方式，将geolocator移到项目根目录中，然后引入。

```
1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
```

解决方案2:远程依赖

远程依赖需要分2种方式，第1是自己搭建的pub，这种方便。第2是使用github类比pub仓库的方式。这里就用github来举例，思路是一样。

我们在本地修改了geolocator_android，然后geolocator又需要引用这个修改过的geolocator_android库，那就把geolocator_android推送到github上，修改中geolocator中的pubspec.yaml，引入GitHub上的geolocator_android库。

```
1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 |
13 |
```

2022.7.6更新

最近测试，发现在某些机型上通过Geolocator.getCurrentPosition方法获取不到定位，暂不清楚原因。可以尝试先获取getLastKnownPosition，没有数据的话再去请求getCurrentPosition方法。

```
1 |
2 |
3 |
4 |
5 |
6 |
7 |
8 |
9 |
10 |
11 |
12 |
13 |
14 |
15 |
```