

Evan

Only let oneself become strong enough, good enough, can afford the life that you want to.

☰ 目录视图

☰ 摘要视图

🔒 订阅专栏

从创业到再就业，浅述对程序员职业生涯的看法 征文 | 你会为 AI 转型么？ 赠书：7月大咖新书机器学习/Android/python

Android之实现定位

标签： android 定位

2016-01-10 23:16 313人阅读 评论(0) 收藏 举报

☰ 分类： 移动开发 (38) ▾

基于Android的定位无非就两种：network、gps。两者各有优劣。

Network:定位快，准确度低，受环境影响小。

GPS：定位慢，准确度高，受环境影响大。

本文要解决的问题：

- 1. locationManager.getLastKnownLocation方法返回null。
- 2. 如何实现快速而又精确的定位。

E文好的话，直接看官网就好了<http://developer.android.com/guide/topics/location/strategies.html>

在你的程序里如果有这样的代码你就要注意了(现在看来这些倒是多余了)

[java]



[java]

```
01. <span style="font-size:14px;"> Criteria criteria = new Criteria();
02.     criteria.setAccuracy(Criteria.ACCURACY_FINE); //高精度
03.     criteria.setAltitudeRequired(false); //无海拔要求
04.     criteria.setBearingRequired(false); //无方位要求
05.     criteria.setCostAllowed(true); //允许产生资费
06.     criteria.setPowerRequirement(Criteria.POWER_LOW); //低功耗
07.
08.     // 获取最佳服务对象
09.     String provider = locationManager.getBestProvider(criteria,true);
10.     locationManager.getLastKnownLocation(provider); </span>
```

locationManager.getBestProvider(criteria,true);方法看起来很完美，但其实返回值就network、gps二选一。而且如果你要求高精度，它会优先检查GPS，如果手机开启了GPS就返回GPS，否则返回network。如果都没开启则返回null。

结合Network、GPS两种定位方式的优劣不难看出为什么getLastKnownLocation方法会返回null（这只针对第一次定位）。

当你开启GPS，provider的值为GPS。这时的定位方式为GPS，由于GPS定位慢(我测试的时间大约为50秒)，所以它不可能立即返回你一个Location对象，所以就返回null了。还有人用下面的方法解决这个问题：

```
[java]
```



```
[java]
```

```
01. <span style="font-size:14px;"> while (location ==null) {  
02.     location = locationManager.getLastKnownLocation(provider);  
03. } </span>
```

这绝对是个愚蠢的做法！举个例子：如果你在室内，gps无法定位到，你的程序将陷入死循环。当然使用requestLocationUpdates可以做到定位且不让程序陷入死循环，但是定位耗时长，甚至得不到定位。

如果使用网络定位呢，不得不说这也是一个不错的选择。locationManager.requestLocationUpdates(

```
LocationManager.NETWORK_PROVIDER, 0, 0, networkListener);
```

网络定位耗时一般在2秒左右（网络差，时间会更长），只要你接入网络，基本上都能获得定位。唯一的缺点就是精度不高。

那能不能将两者结合，这也是本文的重点。既然结合两者，就要同时为两者添加监听

```
[java]
```



```
[java]
```

```
01. <span style="font-size:14px;"> locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER, 1000 * 2, 50, gpsListener);  
02.     locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, 0, 0, networkListener); </span>
```

这样，大概2秒我们就可以得到来自网络的定位，一分钟后再得到来自GPS定位。这时用GPS定位替换网络定位就好了。当然这只是个理想的情况，现实要复杂的多。

比如：

你第一次定位成功返回location，由于网络问题第二次返回null。这时会发现，更新的location没有上次的精确，甚至是null，无法使用，这时我们要判断当前的location和新获得的location哪个更好。可能你获得GPS定位后，由于天气、进入隧道等原因GPS服务器丢失，无法更新location（这时一个好的做法是切换到network定位）。还有可能用户没有开启GPS和network，根本就谈不上定位（其实每次定位成功都会有个定位缓存的，可以使用getLastKnownLocation获得）。

综上所述，我们要做的就是：

1. 尝试通过getLastKnownLocation获取上次定位信息
2. 开启network和gps监听
3. 获得network定位信息location
4. 比较当前location和新获取的location哪个更好（来自network）
5. 获得gps定位信息location
6. 停掉network监听
7. 比较当前location和新获取的location哪个更好（来自gps）
8. 如果gps服务器丢失，重新开启network监听

以GPS监听为例

```
[java]
```



[java]

```

01. <span style="font-size:14px;"> // GPS监听的回调函数
02.     private class GPSLocationListener implements LocationListener {
03.
04.         private boolean isRemove = false; //判断网络监听是否移除
05.
06.         @Override
07.         public void onLocationChanged(Location location) {
08.             // TODO Auto-generated method stub
09.             boolean flag = betterLocation.isBetterLocation(location,
10.                 currentBestLocation);
11.
12.             if (flag) {
13.                 currentBestLocation = location;
14.                 updateLocation(currentBestLocation);
15.             }
16.             // 获得GPS服务后，移除network监听
17.             if (location != null && !isRemove) {
18.                 locationManager.removeUpdates(networkListener);
19.                 isRemove = true;
20.             }
21.         }
22.
23.         @Override
24.         public void onProviderDisabled(String provider) {
25.             // TODO Auto-generated method stub
26.         }
27.
28.         @Override
29.         public void onProviderEnabled(String provider) {
30.             // TODO Auto-generated method stub
31.         }
32.
33.         @Override
34.         public void onStatusChanged(String provider, int status, Bundle extras) {
35.             // TODO Auto-generated method stub
36.             if (LocationProvider.OUT_OF_SERVICE == status) {
37.                 Toast.makeText(MainActivity.this, "GPS服务丢失, 切换至网络定位",
38.                     Toast.LENGTH_SHORT).show();
39.                 locationManager
40.                     .requestLocationUpdates(
41.                         locationManager.NETWORK_PROVIDER, 0, 0,
42.                         networkListener);
43.             }
44.         }
45.     } </span>

```

其中isBetterLocation是用来判断哪个location更好的。这个方法来自[android](#)官网的，通过location获取的时间，精度等信息进行判断。

因为之前上传的demo，大家觉得意义不大，所以就不再提供了。

下图的‘微秒’单位错了，应该是毫秒



后来我接触的项目中对定位的要求发现并不是那么高，只是需要简单的获取到当前位置就够了，而且只是在每次启动时获取一次。

Android简单定位的实现可以参见<http://blog.csdn.net/limb99/article/details/18819925>

from:<http://blog.csdn.NET/limb99/article/details/8765584#comments>

顶

1

踩

0

- [上一篇](#) 详解ListView加载网络图片的优化
- [下一篇](#) Android实现自动定位城市并获取天气信息

相关文章推荐

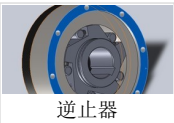
- | | |
|--|---|
| • 在Android系统上定位功能实现方法 | • Android中级第五讲--GPRS定位的实现 |
| • 教程：实现Android的不同精度的定位（基于网络... | • 转：教程：实现Android的不同精度的定位（基于... |
| • Android 定位实现（1） | • Android之3.5版的百度地图的定位、覆盖物实现... |
| • android实现百度地图定位 | • 百度Android定位SDK实现获取当前经纬度及位置 |
| • Android实现高德地图定位详细流程 | • Android下实现GPS定位服务 |



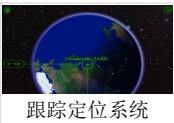
定量控制器



断路器测试仪



逆止器



跟踪定位系统



便携式超声波流量



会计师培训学校



脉冲控制仪

猜你在找

- | | |
|---------------------------------------|---|
| • 机器学习之概率与统计推断 | • 机器学习之数学基础 |
| • 机器学习之凸优化 | • 机器学习之矩阵 |
| • 响应式布局全新探索 | • 探究Linux的总线、设备、驱动模型 |
| • 深度学习基础与TensorFlow实践 | • 深度学习之神经网络原理与实战技巧 |
| • 前端开发在线峰会 | • TensorFlow实战进阶：手把手教你做图像识别应用 |