

Flutter 中获取设备信息 以及 获取地理位置

主讲教师：（大地）

合作网站：www.itying.com （IT 营）

我的专栏：<https://www.itying.com/category-79-b0.html>

一、Flutter 中获取设备信息.....	1
二、Flutter 中使用高德定位准备工作获取 key.....	1
三、Flutter 实现用高德定位.....	1

一、Flutter 中获取设备信息

https://pub.dev/packages/device_info

二、Flutter 中使用高德定位准备工作 获取 key

- 1、申请成为开发者
- 2、创建应用配置获取 Key （参考教程演示）

<https://lbs.amap.com/api/android-sdk/guide/create-project/get-key>

三、Flutter 实现用高德定位

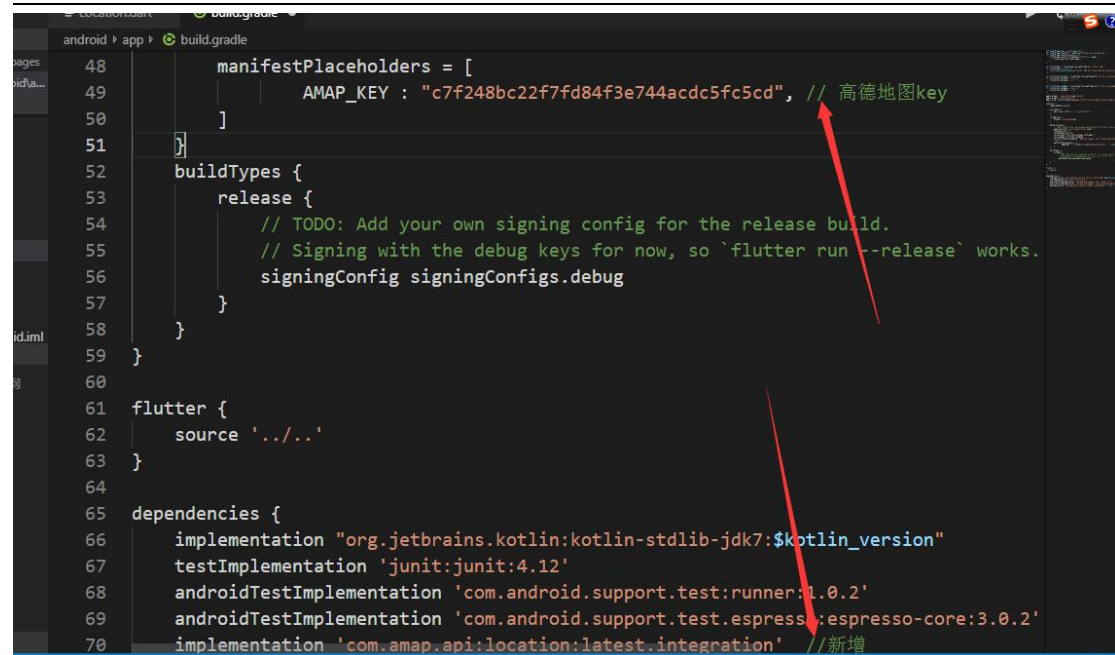
https://pub.dev/packages/amap_location

1、配置安装依赖

```
dependencies:  
    amap_location: ^0.2.0
```

2、修改 你的项目目录/app/build.gradle 在 android/defaultConfig 节点修改 manifestPlaceholders, 新增高德地图 key 配置

```
android {  
    .... 你的代码  
  
    defaultConfig {  
        .....  
        manifestPlaceholders = [  
            AMAP_KEY : "aa9f0cf8574400f2af0078392c556e25", // 高德  
地图 key  
        ]  
    }  
  
    ... 你的代码  
  
    dependencies {  
        /// 注意这里需要在主项目增加一条依赖, 否则可能发生编译不通过的  
情况  
        implementation 'com.amap.api:location:latest.integration'  
        ... 你的代码  
    }  
}
```



```
48     manifestPlaceholders = [  
49         AMAP_KEY : "c7f248bc22f7fd84f3e744acdc5fc5cd", // 高德地图key  
50     ]  
51 }  
52 buildTypes {  
53     release {  
54         // TODO: Add your own signing config for the release build.  
55         // Signing with the debug keys for now, so `flutter run --release` works.  
56         signingConfig signingConfigs.debug  
57     }  
58 }  
59 }  
60  
61 flutter {  
62     source '../..'  
63 }  
64  
65 dependencies {  
66     implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"  
67     testImplementation 'junit:junit:4.12'  
68     androidTestImplementation 'com.android.support.test:runner:1.0.2'  
69     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'  
70     implementation 'com.amap.api:location:latest.integration' //新增
```

具体参考视频教程配置

3、引入库使用

```
import 'package:flutter/material.dart';  
  
import 'package:amap_location/amap_location.dart';  
  
class LocationPage extends StatefulWidget {  
    LocationPage({Key key}) : super(key: key);  
  
    _LocationPageState createState() => _LocationPageState();  
}  
  
class _LocationPageState extends State<LocationPage> {  
  
    @override  
    void initState() {  
        super.initState();  
        this._getLocation();  
    }  
}
```

```
}
_getLocation() async{
    await AMapLocationClient.startup(new
AMapLocationOption( desiredAccuracy:CLLocationAccuracy.kCLLocationAcc
uracyHundredMeters ));
    var location=await AMapLocationClient.getLocation(true);
    print(location.accuracy);
    print(location.longitude);
}
@override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(
            title: Text("地理定位演示"),
        ),
        body: Text("获取定位"),
    );
}
}
```