

```
import Tabs from '@theme/Tabs';
import TabItem from '@theme/TabItem';
```

## 集成 GioKit

GioKit 代码已托管在 [Github](#) 上，欢迎 star,fork。

:::info

**Gradle**插件版本： 3.2.1及以上

**Android**系统版本： Android 5.0 及以上, **GioKit**仅支持 **AndroidX**

请在 **Android SDK** 基础上使用 **GioKit**

...

## 添加依赖

在 project 级别的 build.gradle 文件中添加 giokit-plugin 依赖和 maven 仓库。

```
buildscript {
    repositories {
        // 添加maven仓库
        mavenCentral()
        maven { url "https://s01.oss.sonatype.org/content/repositories/snapshots/" }
    }
    dependencies {
        // GioKit plugin
        classpath "com.growingio.giokit:giokit-plugin:1.1.0-SNAPSHOT"
    }
}

allprojects {
    repositories {
        // 添加maven仓库
        mavenCentral()
        maven { url "https://s01.oss.sonatype.org/content/repositories/snapshots/" }
    }
}
```

在 app 级别的 build.gradle 文件中添加 com.growingio.giokit 插件、 giokit 依赖。

:::info

**GrowingIO Android SDK 3.0**及以上版本，请添加

```
apply plugin: 'com.growingio.giokit'
```

...

:::info

**GrowingIO Android SDK 2.x 版本，请添加**

```
apply plugin: 'com.growingio.giokit.saas'
```

...

:::caution 注意

以上二者根据相应的 GrowingIO Android SDK 选择对应的插件，请不要一起使用！！

...

```
...
dependencies {
    ...
    // Giokit
    debugImplementation "com.growingio.giokit:giokit:1.1.0-SNAPSHOT"
    releaseImplementation "com.growingio.giokit:giokit-no-op:1.1.0-SNAPSHOT"
}
```

:::caution 注意

为了避免在正式环境下出现不必要的错误，请务必只在 Debug 环境下使用 Giokit 工具。

...

## 初始化

请将 Giokit 的初始化代码放入 Application 的 onCreate 中。

```
<Tabs
    groupId="code-language"
    defaultValue="kotlin"
    values=[
        {label: 'java', value: 'java'},
        {label: 'kotlin', value: 'kotlin'},
    ]
}>
```

```
public class MyApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        GioKit.with(this).build();
    }
}
```

```
class MyApplication : Application() {
    override fun onCreate() {
        super.onCreate()
        GioKit.with(this).build()
    }
}
```

## 插件配置

为了满足用户的需求，我们在 Giokit 插件中添加了多项配置。

```
giokitExt {
    debugMode false
    enableIncremental false
    enableRelease false
    // 统计该域值下所有埋点信息，如 com.growingio 表示统计 com.growingio 包名下的埋点代码
    trackFinder {
        domain = ["com.growingio.giokit.demo"]
        // 若用户自己封装了SDK customEvent 方法，可通过在此设置来查找封装类调用的代码
        // className "com.growingio.giokit.demo.AutotrackerUtils"
        // methodName "trackCustomEvent"
    }
}
```

1. debugMode 为true时，项目编译的时候会输出相应的 Debug 信息；
2. enableIncremental 是否打开增量编译，默认为true。由于 Giokit 会查找代码中的所有埋点信息，所以在增量编译时有时候会出现无法找到埋点代码的情况，这时候可以将其设置false，来保证每次查找的准确性；
3. enableRelease 是否支持release打包。giokit 是只推荐在 debug 环境下使用，若一定要在release环境下使用，则需要打开此开关来使插件生效；
4. 为了方便统一查看用户的手动埋点信息，我们通过 trackFinder 配置来查找在应用中手动埋点调用的位置。默认查找域名为项目的 ApplicationId