

Timber: Android日志记录



sunnyaxin

关注

1 2017.10.17 19:48:48 字数 797 阅读 21,355

前言：Log日志在Android项目开发中的作用非常重要，但项目越做越大，开发者会面临很多问题，比如管理各种不同的Log，不同版本控制是否输入Log等。Timber是一个轻量级的第三方库，能够帮助开发者更好的使用Android Log。

面临问题

1. 项目使用Log规范不统一，例如：关于okhttp的实现使用了Logging Interceptor输出log，关于rxjava的实现使用了onError，其他使用了Android Log机制
2. 项目开发时，为了方便调试，需要输出Log，但发布后，Log中包含很多项目相关的信息，如果全部暴露会有安全隐患，但完全删除也不利于发生问题后的查看调试，因此需要一个开关，关闭开关则不能输出log，打开开关则可以输出log

Timber 介绍

Timber的使用主要通过添加Tree实例，添加Tree实例可以通过方法Timber.plant完成。实例化Tree需要尽量早，推荐在项目Application的onCreate()方法中完成。

其中，DebugTree自动实例化，是Timber中已经实现了tree的类，可直接拿来用，并在使用他的类中以类名为其TAG，但是其他的Tree不会自动实例化，需要进行手动实现扩展Timber.Tree实现。

下载

在build.gradle中

```
1 | compile 'com.jakewharton.timber:timber:4.5.1'
```

注册

在onCreate()中

```
1 | if (BuildConfig.DEBUG) {
2 |     Timber.plant(new Timber.DebugTree());
3 | } else {
4 |     Timber.plant(new CrashReportingTree());
5 | }
6 |
7 | private static class CrashReportingTree extends Timber.Tree {
8 |     @Override
9 |     protected void log(int priority, String tag, String message, Throwable t) {
10 |
11 |     }
12 | }
```

使用

```
1 | Timber.tag("activityName");
2 | Timber.d("Activity Created");
```

好处

- 不再重复使用TAG。之前使用Log方法每次使用都需要定义并传入一个TAG，在Timber中，若使用DebugTree，默认为当前类名，若使用自定义Tree，直接定义一次，到处使用，不需重复设置
- 在打Timber调用处，不仅可以只输出Log，还可以执行其他操作。比如：存到数据库，进行输出，Socket传输，甚至显示到UI界面上等，只需要自定义Tree，并将对应逻辑在 `log(int priority, String tag, String message, Throwable t)` 方法中实现
- 统一项目Log日志方法，输出形式，输出时间，更加符合面向对象思想

isLoggable 方法作为开关控制Log输出

语法

```
1 | boolean isLoggable (String tag, int level)
```

通过指定标签指定级别，控制日志是否显示。默认级别为 `Info`，表示高于或等于 `Info` 级别的日志都可以显示。

设置

1. 远程设置系统属性更改默认级别

```
1 | adb -d shell setprop log.tag.<YOUR_LOG_CAT> <LEVEL>
```

2. 系统目录 `/data/local.prop`，创建一个local.prop文件，文件内容：

```
1 | log.tag.<YOUR_LOG_TAG>=<LEVEL>
```

Level级别

LEVEL	NUMBER
VERBOSE	2
DEBUG	3
INFO	4
WARN	5
ERROR	6
ASSERT	7

例子

在自定义Tree中的log方法中进行控制，若为Debug环境下，可以产出log；若为Release环境，且打开开关，则可以产生log；若为Release环境，且关闭开关，则不可产生log。

1. Java代码如下

```
public class LogTree extends Timber.DebugTree {
    @Override
    protected void log(int priority, String tag, String message, Throwable t) {
        if (BuildConfig.DEBUG || Log.isLoggable("MyTag", Log.DEBUG)) {
            super.log(priority, tag, message, t);
        }
    }
}
```

2. 命令行中

若执行如下命令，则打开开关，可以产生log

```
1 | adb -d shell setprop log.tag.MyTag DEBUG
```

若执行如下命令，则关闭开关，不可产生log

```
1 | adb -d shell setprop log.tag.MyTag WARN
```