Android开发笔记

# 序

本文档写给有一定android基础，独立开发app能力欠缺的新人。

开发者Java基本的语法要具备。面向对象设计思想要懂。反射注解高级应用需要了解。

假设app的packageName为 phoebe.frame.lib

# Appllication的功能设计

每一个App都应该有一个扩展的Application。这个类中主要写一些整个app广泛用到的方法或app相关配置操作

public class PhoebeApp extends android.os.Application

androidManifest.xml中配置

<application android:name=*"*phoebe.frame.lib*.PhoebeApp"*

application的生命周期是从app启动到退出。进程结束。所以在application定义的变量都命名为static

## Context

App的上线文引用，主要用于调用系统api

比如说 context.getSystemService(String name) 我们在app中调用的时候可以

PhoebeApp.getContext().getSystemService(Context.NOTIFICATION\_SREVICE)

## Handler

App中会频繁用到handler与线程的交互。有时候我们需要用Handler#postDelayed(Runnable r , int delayMills) 这个时候我们无需重新new handler(); 只需要调用全局的PhoebeApp#getHander() 即可

更高级的用法先不讲。后续再

至此一个基本的Application扩展类已经定义完成

# Log类的设计

正式代码设计之前我们先考虑一下日志类的使用。

日志主要用在调试app 一些复杂的业务流程，借助log 可以很容易问题所在

因为android.util.Log 定义为final类(不可继承) 所以我采用组合的方式才重写一个可以控制开关的工具类。考虑到Log应用的广泛性，所以定义为static，生命周期为全局

public class Log{

private staic boolean open = false; //默认关闭

public static boolean isOpen() {

return open;

}

public static void setOpen(boolean open) {

Log.open = open;

}

}

Log的开关控制，一般在app初始化的地方设置。后续再讨论

public static void d(String tag, String msg){

if(open == false){

return; // 不打印日志

}

android.util.Log.*d*(tag, msg);

}

其中tag和msg 考虑到传入的参数需要频繁转型

比如说像打印一个int型的变量值。这时候会很不方便。同理其他非String类型的都需要转化为String 。

因为优化一下当前方法

public static void d(Object tag, Object msg){

if(open == false){

return; // 不打印日志

}

android.util.Log.*d*(String.valueOf(tag), String.valueOf(msg));

}

这时候如果传入的 tag == null 或者 “” 或者 包含有” ” \n\r等字符 ，那么输出的结果可能不是我们想要的，

接着继续优化一下该方法

public static void d(Object tag, Object msg){

if(open == false){

return; // 不打印日志

}

tag = handleMsgOrTag(tag);

msg = handleMsgOrTag(msg);

android.util.Log.*d*(String.valueOf(tag), String.valueOf(msg));

}

private static Object handleMsgOrTag(Object msgOrTag) {

if (msgOrTag == null) {

msgOrTag = "[null]";

} else if (msgOrTag.toString().trim().length() == 0) {

msgOrTag = "[\"\"]";

} else {

msgOrTag = msgOrTag.toString().trim();

}

return msgOrTag;

}