



链滴

# Android 全局异常捕获

作者: [Single](#)

原文链接: <https://ld246.com/article/1500603626962>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

```

import java.io.PrintWriter;
import java.io.StringWriter;
import java.io.Writer;
import java.lang.Thread.UncaughtExceptionHandler;
import java.lang.reflect.Field;
import java.util.LinkedHashMap;

import android.content.Context;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.os.Build;
import android.os.Process;

public class CrashHandler implements UncaughtExceptionHandler {
    private static CrashHandler instance;
    private UncaughtExceptionHandler defaultHandler;
    private Context context;
    /**
     * 保存异常信息集合
     */
    private LinkedHashMap<String, String> crashAppLog = new LinkedHashMap<String, String>(
    ;

    private CrashHandler() {}

    public static CrashHandler getInstance() {
        if (instance == null) {
            synchronized (CrashHandler.class) {
                if (instance != null) {
                    instance = new CrashHandler();
                }
            }
        }
        return instance;
    }

    public void init(Context context) {
        if (context == null)
            throw new NullPointerException("context不能为空");
        this.context = context;
        // 获取默认的异常处理器
        this.defaultHandler = Thread.getDefaultUncaughtExceptionHandler();
        // 设置自己的异常处理器
        Thread.setDefaultUncaughtExceptionHandler(this);
    }

    @Override
    public void uncaughtException(Thread t, Throwable e) {

        if (!handleException(e) && defaultHandler != null) {
            // 没有人为处理异常调用系统默认处理器处理
            defaultHandler.uncaughtException(t, e);
        } else {

```

```

        // 退出程序
        Process.killProcess(Process.myPid());
        System.exit(0);
    }
}

/**
 * 人为处理异常
 *
 * @param e
 * @return
 */
private boolean handleException(Throwable e) {
    if (e == null) {
        return false;
    }
    try {
        // 收集错误信息
        collectErrorINFo(e);
        // 保存错误信息
        saveErrorINFo();
    } catch (Exception e1) {
    }

    return false;
}

/**
 * 保存错误信息
 *
 * @throws Exception
 */
private void saveErrorINFo() throws Exception {

}

/**
 * 收集错误信息
 *
 * @throws Exception
 */
private void collectErrorINFo(Throwable e) throws Exception {
    // 获取包信息
    PackageManager packageManager = context.getPackageManager();
    if (packageManager != null) {
        PackageInfo packageInfo = packageManager.getPackageInfo(
            context.getPackageName(), PackageManager.GET_ACTIVITIES);
        if (crashAppLog != null && packageInfo != null) {
            crashAppLog.put("version_name", packageInfo.versionName);
            crashAppLog.put("version_code", packageInfo.versionCode + "");
        }
    }

    // 获取手机信息

```

```

Field[] fields = Build.class.getFields();
if (fields != null && fields.length > 0) {
    for (Field field : fields) {
        if (crashAppLog != null && field != null) {
            field.setAccessible(true);
            crashAppLog
                .put(field.getName(), field.get(null).toString());
        }
    }
}

// 获取异常信息
Throwable cause = e.getCause();
Writer writer = new StringWriter();
PrintWriter printWriter = new PrintWriter(writer);
while (cause != null) {
    cause.printStackTrace(printWriter);
    cause = e.getCause();
}
printWriter.flush();
printWriter.close();
crashAppLog.put("cause", writer.toString());
}
}

```

在Application中调用CrashHandler.getInstance().init(this);即可