

Decard-Logansdk

发布日志

V 2.2.02

2020-08-10

【新增】LogUtilsConfig 配置增加 setLogLevel(int level) 接口，不传入则默认DEBUG等级日志

【新增】LogUtilsConfig 增加日志等级配置常量，方便配置

V 2.2.01

2020-08-06

- 【新增】LogUtilsConfig 配置增加 setDeviceSn(deviceSn)接口，传入SN号, 传入 "" 或者不传，则使用 Build.SERIAL
- 【废弃】XCrash配置 setLogDir() 接口，改为由LogUtilsConfig 传入的projectNum获取Dir路径
- 【修复】底层so库，支持日志打印内容含有换行，json格式

V 2.2.00

2020-08-04

- 【修改】日志写入文件格式化
- 【新增】LoganUtils打印日志调用接口，增加各个Log等级
- 【新增】LoganUtils各个打印接口支持外部封装，Logcat中打印出文件及行号方法名
- 【废弃】public static void d(Class<?> object, String msg, int type, boolean isWriteNow)

V 2.1.01

2020-07-27

- 【更改】仅修改版本号，同 V 2.1.00

V 2.1.00

2020-07-21

- 【修改】日志写入引擎替换为tecent.mars
- 【修改】修改sdk兼容版本最低15
- 【去除】去除掉LogUtilConfig setMaxFile配置, 设置将不会生效

V 0.1.4

2020-06-24

- 【修改】修改日志写入目录为sd/Dcard/log/projectNum/
- 【修改】修改crash日志存储目录为sd/Dcard/crash/projectNum/

V 0.1.2

2020-05-09

- 【新增】新增服务器ip和port的配置

使用

1. 增加依赖

增加仓库地址

```
maven { url 'http://47.93.180.222:8081/repository/mobile-releases/' }
```

添加依赖

```
dependencies {  
    //version 2.2.00  
    implementation 'com.decard.mobile:logansdk-utils:version'  
}
```

2. 配置 Config

添加 Crash 日志的相关配置

```
XCrash.InitParameters xCrashParams = new XCrash.InitParameters()  
    //是否将异常重新抛出给系统  
    .setJavaRethrow(true)  
    //Java类型崩溃日志总数量  
    .setJavaLogCountMax(10)  
    .setJavaDumpAllThreadsWhitelist(new String[] { "^main$", "^Binder:.*",  
    ".*Finalizer.*" })  
    //是否将异常重新抛出给系统  
    .setNativeRethrow(true)  
    //Native类型  
    .setNativeLogCountMax(10)  
    //Native层的Thread白名单  
    .setNativeDumpAllThreadsWhitelist(new String[] { "^xcrash\\.sample$", "^Signal  
Catcher$", "^Jit thread pool$", ".*(R|r)ender.*", ".*Chrome.*" })  
    .setNativeDumpAllThreadsCountMax(10)  
    //是否将异常重新抛出给系统  
    .setAnrRethrow(true)  
    //ANR类型崩溃日志总数量  
    .setAnrLogCountMax(10)  
    //每个崩溃日志文件大小, 512kb  
    .setPlaceholderSizeKb(512);
```

添加普通日志的相关配置，其中：

- String key：key为密钥，web端可以获取到
- String url：ip地址，传入“”，则使用默认
- String port：端口号，传入“”，则使用默认
- String deviceSn：设备SN号，传入“”或者不传，则使用Build.SERIAL

.setProjectNum(projectNum) 必须设置

```
LoganUtilsConfig loganUtilsConfig = new LoganUtilsConfig.Builder()  
    .setProjectNum(projectNum)  
    //密钥  
    .setEncryptKey16(key)  
    .setXCrashParams(xCrashParams)  
    //http 请求的ip
```

```
.setUrl("dclog.decard.com")
//http 请求的端口port
.setPort("50015")
//mqtt服务器的完整url
.setMqttUrl("47.93.180.222:1883")
//设备SN号
.setDeviceSn(deviceSn)
//配置日志打印等级（V2.2.02版本）
.setLogLevel(LoganUtilsConfig.LEVEL_DEBUG)
.setContext(context)
.build();
```

3. 初始化 LoganSdk

在使用之前，必须先初始化 Logan

传入参数的说明：

- 传入上述日志配置的LoganUtilsConfig

```
//初始化Logan日志生成操作，需要在权限申请完成后调用
LoganUtils.getInstance().initLogan(loganUtilsConfig);
```

4. 写日志

release模式下仅打印至本地文件，debug下仅打印在控制台

V 2.2.02及之后

日志打印取决于setLogLevel 配置

V 2.1.01及之前打印日志接口不生效

V 2.1.01之后

```
LoganUtils.d("此处设置TAG", "测试dddd");
LoganUtils.v("此处设置TAG", "测试vvvv");
LoganUtils.i("此处设置TAG", "测试iiii");
LoganUtils.w("此处设置TAG", "测试www");
LoganUtils.e("此处设置TAG", "测试eeee");
```

V 2.1.01及之前

```
@Deprecated
LoganUtils.d(this.getClass(), "test", 2, false);
```

LoganUtils.d 方法有四个参数，详解如下：

```
Class<?> object, String msg, int type, boolean isWriteNow
```

- Class<?> object：传入当前类；
- String msg：写入的日志内容；
- int type：写入的日志类型，注意：type值1已被logan内部占用，建议业务方的日志类型使用新的type值用来区分；(V2.1.00版本已经废弃使用)
- boolean isWriteNow：是否立即写入。(V2.1.00版本已经废弃使用)

5.说明

项目中引入 Logansdk , 需要从写入日志信息到外部存储。

- 普通日志的路径为/sdcard/DCard/log/projectNum/
- 崩溃日志的路径为/sdcard/DCard/log/projectNum/crash/

LoganService.apk 为上传日志文件的组件, 需要 **单独安装**;

声明应用的权限

在 Android6.0 及以上的系统 需要手动授权 :

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
```