

MNCrashMonitor

MNCrashMonitor 监听程序崩溃日志,直接页面展示崩溃日志列表,调试方便,测试人员可以随时给程序猿查看日志详情,可以动态添加日志内容,手机直接查看日志内容可以分享,复制,生成长截图。

JitPack V1.0.7

截图

MNCrashMonitor

抛出自定义异常

日志列表页面

下一页

添加额外的LOG信息

获取日志保存路径

崩溃日志文件夹的路径：

```
/storage/emulated/0/Android/data/com.maning.  
crashmonitor/cache/crashLogs
```

崩溃啦~





重启应用

查看崩溃日志

傍晚6:44
4月20日星期四



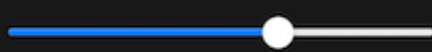

CYXX12301


蓝牙



移动数据


振动


自动旋转



自动 

 设置 • 下午1:36
正在通过 USB 充电
点击查看更多选项

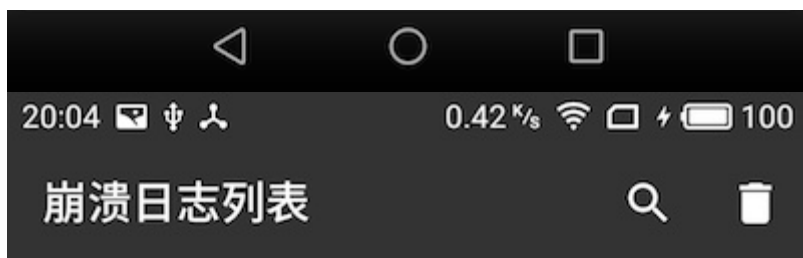
 Android 系统
已连接 USB 调试
点击可停用 USB 调试

 CrashMonitor 

2017-04-20_18-44-10

java.lang.IllegalStateException: Could not execute
method for android:onClick
at
android.support.v7.app.AppCompatActivity\$Inflater\$D
eclaredOnClickListener.onClick(AppCompatActivity.java:293)
at android.view.View.performClick(View.java:5646)
at android.view.View\$PerformClick.run(View.java:
22458)
at
android.os.Handler.handleCallback(Handler.java:
761)
at





CrashLog_V1.0_T2017-11-30_19-50-20.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/
cache/crashLogs/CrashLog_V1.0_T2017-11-30_19-50-20.txt

CrashLog_V1.0_T2017-11-30_19-47-48.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/
cache/crashLogs/CrashLog_V1.0_T2017-11-30_19-47-48.txt

CrashLog_V1.0_T2017-11-30_19-47-18.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/
cache/crashLogs/CrashLog_V1.0_T2017-11-30_19-47-18.txt

CrashLog_V1.0_T2017-11-30_19-46-37.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/
cache/crashLogs/CrashLog_V1.0_T2017-11-30_19-46-37.txt

CrashLog_V1.0_T2017-11-30_19-44-06.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/
cache/crashLogs/CrashLog_V1.0_T2017-11-30_19-44-06.txt

CrashLog_V1.0_T2017-11-30_19-43-20.txt

/storage/emulated/0/Android/data/com.maning.crashmonitor/



用户手机号码: 16666666666
用户网络环境: wifi

额外内容, 动态设置

Crash Time : 2017-11-30_19-29-29

Device Manufacturer : Meizu

Device CPU ABI : arm64-v8a

Device Model : m2 note

Android OS Version : 5.1

Android SDK : 22

App VersionName : 1.0

App VersionCode : 1

```
java.lang.IllegalStateException: Could not execute
method for android:onClick
    at android.support.v7.app.
AppCompatActivity$DeclaredOnClickListener.
onClick(AppCompatActivity.java:293)
    at android.view.View.performClick(View.java:4900)
    at android.view.View$PerformClick.run(View.java:
20381)
    at android.os.Handler.handleCallback(Handler.java:815)
    at android.os.Handler.dispatchMessage(Handler.java:
104)
    at android.os.Looper.loop(Looper.java:194)
    at android.app.ActivityThread.main(ActivityThread.java:
5877)
    at java.lang.reflect.Method.invoke(Native Method)
    at java.lang.reflect.Method.invoke(Method.java:372)
    at com.android.internal.os.
ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:
1020)
    at com.android.internal.os.ZygoteInit.main(ZygoteInit.
java:815)
Caused by: java.lang.reflect.InvocationTargetException
    at java.lang.reflect.Method.invoke(Native Method)
```

如何添加

Gradle添加:

1.在Project的build.gradle中添加仓库地址

```
allprojects {
    repositories {
        ...
        maven { url "https://jitpack.io" }
    }
}
```

2.在app目录下的build.gradle中添加依赖

```
dependencies {  
    compile 'com.github.maning0303:MNCrashMonitor:v1.0.7'  
}
```

使用方法:

1.Application 的 onCreate()方法 初始化:

```
/**  
 * 初始化日志系统  
 * context : 上下文  
 * isDebug : 是不是Debug模式,true:崩溃后显示自定义崩溃页面 ;false:关闭应用,  
不跳转奔溃页面(默认)  
 * CrashCallback : 回调执行  
 */  
MNCrashMonitor.init(this, true, new MNCrashCallback() {  
    @Override  
    public void onCrash(File file) {  
        //可以在这里保存标识,下次再次进入把日志发送给服务器  
        Log.i(TAG, "CrashMonitor回调:" + file.getAbsolutePath());  
    }  
});
```

2.动态设置其它额外信息到日志中 (如用户手机号, wifi状态等信息) :

```
String extraInfo = "用户手机号码: 16666666666" +  
    "\n用户网络环境: wifi";  
MNCrashMonitor.setCrashLogExtraInfo(extraInfo);
```

3.其它相关方法介绍:

```
/**  
 * Created by maning on 2017/4/20.  
 * 主类  
 */  
public class MNCrashMonitor {  
  
    /**  
     * 初始化  
     *  
     * @param context 上下文  
     */  
    public static void init(Context context) {  
        MNCrashHandler crashHandler = MNCrashHandler.getInstance();  
        crashHandler.init(context);  
    }  
  
    /**  
     * 初始化  
     *  
     */  
}
```

```

    * @param context 上下文
    * @param isDebug 是否处于debug状态
    */
    public static void init(Context context, boolean isDebug) {
        MCrashHandler crashHandler = MCrashHandler.getInstance();
        crashHandler.init(context, isDebug);
    }

    /**
     * 初始化
     *
     * @param context 上下文
     * @param crashCallbacks 回调
     */
    public static void init(Context context, MCrashCallback
crashCallbacks) {
        MCrashHandler crashHandler = MCrashHandler.getInstance();
        crashHandler.init(context, crashCallbacks);
    }

    /**
     * 初始化
     *
     * @param context 上下文
     * @param isDebug 是否处于debug状态
     * @param crashCallbacks 回调
     */
    public static void init(Context context, boolean isDebug,
MCrashCallback crashCallbacks) {
        MCrashHandler crashHandler = MCrashHandler.getInstance();
        crashHandler.init(context, isDebug, crashCallbacks);
    }

    /**
     * 日志列表页面
     *
     * @param context
     */
    public static void startCrashListPage(Context context) {
        Intent intent = new Intent(context.getApplicationContext(),
CrashListActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.getApplicationContext().startActivity(intent);
    }

    /**
     * 打开奔溃展示页面
     *
     * @param context
     */
    public static void startCrashShowPage(Context context) {
        Intent intent = new Intent(context.getApplicationContext(),
CrashShowActivity.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.getApplicationContext().startActivity(intent);
    }

    /**

```



```

        * 获取日志的路径
        *
        * @param context
        * @return
        */
    public static String getCrashLogFilesPath(Context context) {
        return MFileUtils.getCrashLogPath(context);
    }

    /**
     * 设置额外的日志内容，当发生崩溃的时候会写入当前内容到文件开头
     * 例如L用户手机号码，Token ，网络环境等定制化东西
     *
     * @param content 内容
     */
    public static void setCrashLogExtraInfo(String content) {
        if (!TextUtils.isEmpty(content)) {
            MCrashHandler crashHandler = MCrashHandler.getInstance();
            crashHandler.setExtraContent(content);
        }
    }
}

```

文件的位置:

```

/Android/data/包名/cache/crashLogs/
或者
/data/data/包名/cache/crashLogs/

```

生成截图的位置:

```

Environment.getExternalStorageDirectory() + "/crash_pic_" +
System.currentTimeMillis() + ".jpg"

```

注意:

当应用已启动就崩溃的无法打开页面,直接看通知或者去文件夹里面查看: