

[SCA] Crash 安全应急方案 - 使用文档

1. 背景

对于Android应用来说，运行过程中发生错误则会发生崩溃（Crash，即抛出未处理的异常），系统的默认处理是关闭或重启应用。这对于用户来说是一个相当糟糕的体验，如果启动过程中崩溃甚至会导致应用不可用。预防崩溃是开发工作的重中之重，但一旦崩溃发生，我们还需要一套应急方案，特别是对于启动崩溃，我们希望也能保证用户可以一定程度上使用我们的应用。

2. 功能

功能点	功能描述
捕获崩溃和启动异常	崩溃触发或启动异常触发时，跳转自定义安全页面查看异常信息，选择后续操作
发生异常后引导用户尝试恢复	后续操作默认包括，关闭应用、清除应用数据并重启或者打开降级兜底开关并重启
恢复操作支持多进程	可添加自定义的后续操作，操作支持在特定进程执行

3. 使用说明

demo工程：<https://git.garena.com/shopee/ssz-client/android/supplychain/corelib/-/tree/tech/yangtao-crash-security/crash/sample>

3.1 依赖引入

```
implementation com.shopee.sc:crash-security:{latest_version}
```

3.2 初始化

需要在Application的attachBaseContext初始化，让初始化尽量靠前。

```
CrashSecurity.init(this)
```

参数需传入Application对象。初始化支持多进程，故无需过滤进程执行。

3.3 自定义处理项

```
CrashSecurity.addSolveOperation(object : BaseOperation() {
    override fun displayName(): String {
        return "My Method"
    }

    override fun id(): String {
        return "my_id"
    }

    override fun execute() {
        Log.d("Crash", "execute pid: ${Process.myPid()}")
    }
})
```

建议紧接着初始化，添加自定义项。这里需要实现三个方法，displayName是返回操作页面展示项的显示名称，id是自定义项的自定义标识，execute方法是执行操作的回调。

注意考虑多进程，在不同的进程添加即可运行在不同进程。多个进程可以添加同一个自定义项，但需要在execute回调里考虑不同进程的回调。

3.4 启动异常测试

启动异常可以调用CrashSecurity类的两个方法来测试，原理是直接修改SP里的flag，并在退出应用时不重置。

testPerformStartError方法是立即触发启动异常，后续可以延时抛出Java异常，来模拟启动异常紧跟Java崩溃的场景。

testStartErrorAndQuit方法是立即杀掉应用，并且下次启动必定触发启动异常。

调试日志TAG为“CrashSecurityLogger”。