

Protect SDK — Android

配置为 JSON 字符串。必填 `access_key`；其余可选，不传则使用默认。

键	默认	说明
<code>access_key</code>	(必填)	接入密钥
<code>fast_tick_enabled</code>	<code>false</code>	按接入要求开启
<code>log_enabled</code>	<code>false</code>	<code>true</code> 时输出调试日志（logcat 过滤 PROTECT）
<code>lifecycle_keep_enabled</code>	<code>false</code>	见下文「前后台」

示例：{"access_key": "YOUR_ACCESS_KEY"}

包内容

路径	说明
<code>protect.aar</code>	SDK
<code>demo/ProtectDemoApplication.java</code>	进程内一次 <code>start</code> / <code>stop</code> 的示例
<code>demo/MainActivity.java</code>	退出时再 <code>stop</code> 的示例

API

方法	含义
<code>ProtectLibrary.start(context, config)</code>	启动。成功返回 <code>0</code>
<code>ProtectLibrary.stop()</code>	停止。尚未启动时返回 <code>0</code>
<code>ProtectLibrary.error()</code>	最近一次错误说明；无错误为空串

`start` 传入 `Activity` 时，严重错误退出前可能弹出对话框；只传入 `Application` 时为提示条。

start 返回码：

码	含义
0	成功
1001	<code>config</code> 不是合法 JSON
1003	无法使用应用数据目录，或未传入 <code>Context</code>
1004	已经启动且尚未 <code>stop</code>
1009	启动失败

请检查返回值。`1004`、`1009` 表示未处于可用状态。失败后调用 `error()`。

使用约定

1. 同一进程内：`start` 成功后，再次 `start` 前必须先 `stop`，否则返回 **1004**。全进程只启动一次，不要在每个 `Activity` 里各自 `start`。
2. `start` 与 `stop` 会挡住其后的原有逻辑，大约阻塞 2 秒。在原流程里直接调用，返回后再继续，不要放到别的线程。
3. 旋转屏幕等配置变化会销毁 `Activity`，但进程还在。此时不要 `stop`。只在用户真正退出（`isFinishing()` 为 `true`）或主动结束进程前 `stop`。
4. 系统在后台直接杀掉进程时，不会执行 `stop`。下次冷启动再 `start` 即可。
5. SDK 不负责让应用在后台保持运行。
6. 前后台（启动成功后自动生效）：应用进入后台后默认停止；回到前台后重新启动。短暂失焦（例如旋转）不会当成进入后台。`"lifecycle_keep_enabled":true` 时，进入后台不停止，回到前台仍会重新启动一次。调用 `stop()` 之后，直到下次 `start` 成功，不会因回到前台而自动再启动。

推荐：

```
Application.onCreate      → ProtectLibrary.start(...)
用户退出 / isFinishing    → ProtectLibrary.stop()
```

Manifest:

```
<!-- protect.aar 已声明 INTERNET，合并进宿主即可。
      不要从合并结果里去掉该权限，否则启动容易失败（1009）。 -->
<uses-permission android:name="android.permission.INTERNET" />

<application
    android:name=".ProtectDemoApplication"
    ...>
    <activity android:name=".MainActivity" ... />
</application>
```

避免：

- 只在 `Activity.onCreate` 里 `start`、从不 `stop`，再次进入又 `start` → **1004**。
- 在 `onDestroy` 里不论是否真正退出都 `stop`（含旋转）→ 回到前台异常。
- 只靠 `Application.onTerminate()` 做清理 → 进程被强杀时不会执行。

集成步骤

1. 将 `protect.aar` 放入模块 `libs/`：

```
dependencies {  
    implementation files('libs/protect.aar')  
}
```

2. 复制 `demo/ProtectDemoApplication.java`、`demo/MainActivity.java`（包名可改），在 `Manifest` 里注册 `Application`。确认合并后的清单仍有 `INTERNET`。
3. 将 `YOUR_ACCESS_KEY` 换成实际密钥。用 `logcat` 过滤 `ProtectDemo` 或 `PROTECT` 查看返回码。

自测

1. 按上文加入 `protect.aar`。
2. 复制两个 `Demo` 源文件并注册 `Application`。
3. 运行后应看到启动成功；按返回键退出后应有停止日志；强制停止应用再打开，应能再次启动（不是 1004）。