

Protect SDK — iOS

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
<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.xcframework</code>	SDK（真机与模拟器，Xcode 自动选择）
<code>demo/ProtectDemo.m</code>	调用示例

头文件在 `XCFramework` 内：`Protect.framework/Headers/sdk_ios.h`。

API

函数	含义
<code>protect_start(config)</code>	启动。 <code>config</code> 为 JSON 字符串
<code>protect_stop()</code>	停止
<code>protect_error()</code>	最近一次错误（UTF-8）；无错误为 ""。返回的指针在下次 <code>start / stop / error</code> 之前有效，需要保留请立刻拷贝

返回 `0` 表示成功，其它为错误码。失败后调用 `protect_error()`。

在应用启动完成后调用 `protect_start`，在进程退出前调用 `protect_stop`。例如 `UIKit` 的启动完成与即将终止回调，或 `SwiftUI` 里对应的启动与退出时机。

前后台（启动成功后自动生效）：进入后台后默认停止，回到前台后重新启动。短暂失焦（例如控制中心）不会触发。调用在回到调用方之前完成，可能短时间卡住。`"lifecycle_keep_enabled":true` 时，进入后台不停止，回到前台仍会重新启动一次。主动 `protect_stop` 之后，直到下次 `protect_start` 成功，不会因回到前台而自动再启动。进程退出时仍须调用 `protect_stop`。

Demo 在启动和停止之间等待 30 秒，只是留出一段运行时间。正式 App 按上面的时机调用即可，不必保留这段等待。

集成步骤

1. 将 `Protect.xcframework` 拖入 Xcode 工程，选择 **Embed & Sign**。
2. 在启动与退出处调用：

```
#import <Protect/sdk_ios.h>
```

```
const char* config = "{\"access_key\": \"YOUR_ACCESS_KEY\"}";  
int rc = protect_start(config);  
if (rc != 0) {  
    NSLog(@"protect_error => %s", protect_error());  
}
```

```
protect_stop();
```

3. 将 `YOUR_ACCESS_KEY` 换成实际密钥。

自测

1. 将 `demo/ProtectDemo.m` 加入 App Target。
2. 确认已嵌入 `Protect.xcframework`（**Embed & Sign**）。
3. 无需再配头文件搜索路径。
4. 替换源码中的密钥后编译运行。该示例只演示调用顺序，不挂在应用生命周期上。