

一、接入方式对比

对比项	封装式接入	SDK 接入
接入条件	不需要源代码, 无需编译	需要前端源代码及一定的开发能力
接入速度	秒级接入, 封装软件开箱即用	行级代码量接入, 几分钟搞定
接入位置	封装游戏登录器, 实现接入	任意集成方式, 登录器, 游戏本地皆可
适用性	目前仅支持 PC 平台的 EXE 封装	全平台支持, PC 移动端全适配
拓展性	无法使用 SDK 提供的 API 接口	更灵活的使用规则, 更强的稳定性

二、获取实例编号和密钥 KEY

三、Windows 应用接入

1、接入方法

PC 的 SDK 接入方法为 dll 模式，需要将 GameShield.dll 集成入项目中并在开启时调用。

2、接入示例

以下提供原生 unity、C++项目的实例 其他类型的同理。

如果有接入的问题欢迎联系我们的客服，将协助您进行接入。

● C/C++项目接入示例

(1)、引用 windows 标准库:

```
1 #include <windows.h>
2 #include <iostream>
```

(2)、加载 dll:

```
1 HMODULE hInstLibrary = LoadLibrary(TEXT("./GameshieldManager.dll")); //根据 dll 在项目里的路径
2     if (!hInstLibrary) {
3         std::cerr << "Unable to load DLL!\n";
4         return 1;
5     }
```

(3)、加载游戏盾 dll 函数:

```
1 //游戏盾开启函数
2 typedef bool (*QNDFunc)(const char* sdkkey, const char* recommendCode);
3 QNDFunc pfnStart = (QNDFunc)GetProcAddress(hInstLibrary, "GameshieldManager_Start");
4 //获取设备唯一 ID 函数
5 typedef const char* (*QNDFunc)();
6 QNDFunc pfnGetUniqueID = (QNDFunc)GetProcAddress(hInstLibrary, "GameshieldManager_GetUniqueID");
7 //重新获取动态 IP 函数
8 typedef void (*QNDFunc)();
9 QNDFunc pfnTryGetDynamicIP = (QNDFunc)GetProcAddress(hInstLibrary,
10 "GameshieldManager_TryGetDynamicIP");
11 //测试 IP 联通性
12 QNDFunc pfnTryTestConnectivity = (QNDFunc)GetProcAddress(hInstLibrary,
13 "GameshieldManager_TryTestConnectivity");
14 //获取本地 IP 接口
15 QNDFunc pfnTryGetLocalIP = (QNDFunc)GetProcAddress(hInstLibrary, "GameshieldManager_TryGetLocalIP");
16
17 if (!pfnStart) {
18     std::cerr << "Function GameshieldManager_Start not found!\n";
19     FreeLibrary(hInstLibrary);
20     return 1;
21 }
```

(4)、加载游戏盾 dll，调用开启函数，代码示例:

```
1 std::string sdkkey = "your-sdk-key"; //必填项，需要传入的 SDK 密钥（后台生成查看）
2 std::string recommendCode = "you-recommend-Code"; //非必填，校验实例的推荐码（传空则不校验）
3 // 调用函数
4 pfnStart(sdkkey.c_str(), recommendCode.c_str());
```

(5)、其他相关 API 接口，代码示例:

```

1  const char* uniqueID = pfnGetUniqueID();           //获取设备唯一 ID 函数
2  pfnTryGetDynamicIP();                             //重新获取动态 IP
3  pfnTryTestConnectivity();                         //测试端口联通性
4  const char* localIP = pfnTryGetLocalIP();         //获取本地 IP

```

• Python 项目接入示例

(1)、引用 ctypes 库, 代码示例:

```

1  from ctypes import cdll, c_char_p

```

(2)、加载 dll, 代码示例:

```

1  lib = cdll.LoadLibrary('./GameshieldManager.dll')

```

(3)、获取游戏盾函数指针, 代码示例:

```

1  fnStart = lib.GameshieldManager_Start
2  pfnStop = lib.GameshieldManager_Stop
3  pfnGetUniqueID = lib.GameshieldManager_GetUniqueID
4  pfnTryGetDynamicIP = lib.GameshieldManager_TryGetDynamicIP
5  pfnTryTestConnectivity = lib.GameshieldManager_TryTestConnectivity
6  pfnTryGetLocalIP = lib.GameshieldManager_TryGetLocalIP

```

(4)、设置参数类型和返回类型, 代码示例:

```

1  pfnStart.argtypes = [c_char_p, c_char_p]
2  pfnGetUniqueID.restype = c_char_p
3  pfnTryGetLocalIP.restype = c_char_p

```

(5)、调用 dll 函数, 代码示例:

```

1  try:
2      pfnStart(
3          b"your-sdk-key",           ##必填项, 需要传入的 SDK 密钥 (后台生成查看)
4          b"you-recommend-Code",    ##非必填, 校验实例的推荐码 (传空则不校验)
5          ) ##游戏盾开启函数
6
7      uniqueID = pfnGetUniqueID()     ##获取设备唯一 ID
8      if uniqueID is not None:
9          print("Unique ID:", uniqueID.decode('utf-8'))
10
11     pfnTryGetDynamicIP()            ##重新获取动态 IP
12     pfnTryTestConnectivity()       ##测试端口联通性
13
14     localIP = pfnTryGetLocalIP()    ##获取本地 IP
15     if localIP is not None:
16         print("Local IP:", localIP.decode('utf-8'))
17
18 except Exception as e:
19     print("An error occurred:", e)

```

• 易语言接入示例

(1)、引用 Windows 库, 代码示例:

```
1 在易语言中，不需要显式引用标准库，它默认包含必要的系统库
```

(2)、加载 dll，代码示例：

```
1  .版本 2
2
3  .全局变量
4  库句柄, 整数型
5
6  库句柄 = 取库 (“./GameshieldManager.dll”)
7  如果 (库句柄 = 0)
8      调试输出 (“无法加载DLL!”)
9      返回
```

(3)、加载游戏盾 DLL 函数，代码示例：

```
1  // 在易语言中，不需要显式声明函数类型，只需使用 导入函数 来获取函数地址。
2
3  .全局变量
4  pfnStart, 整数型
5  pfnGetUniqueID, 整数型
6  pfnTryGetDynamicIP, 整数型
7  pfnTryTestConnectivity, 整数型
8  pfnTryGetLocalIP, 整数型
9
10 pfnStart = 导入函数(库句柄, “GameshieldManager_Start”, , 2)
11 pfnGetUniqueID = 导入函数(库句柄, “GameshieldManager_GetUniqueID”, , 0)
12 pfnTryGetDynamicIP = 导入函数(库句柄, “GameshieldManager_TryGetDynamicIP”, , 0)
13 pfnTryTestConnectivity = 导入函数(库句柄, “GameshieldManager_TryTestConnectivity”, , 0)
14 pfnTryGetLocalIP = 导入函数(库句柄, “GameshieldManager_TryGetLocalIP”, , 0)
15
16 如果 (pfnStart = 0)
17     调试输出 (“未找到函数 GameshieldManager_Start!”)
18     卸载库 (库句柄)
19     返回
```

(4)、调用游戏盾 DLL 的开启函数，代码示例：

```
1  .局部变量
2  sdkkey, 文本型
3  recommendCode, 文本型
4
5  sdkkey = “your-sdk-key” // 必填项，需要传入的 SDK 密钥（后台生成查看）
6  recommendCode = “you-recommend-code” // 非必填，校验实例的推荐码（传空则不校验）
7
8  // 调用函数
9  调用命令(pfnStart, sdkkey, recommendCode)
```

(5)、调用其他相关 API 接口，代码示例：

```

1  .局部变量
2  uniqueID, 文本型
3  localIP, 文本型
4
5  // 获取设备唯一 ID 函数
6  uniqueID = 调用命令(pfnGetUniqueID)
7  调试输出 ("Unique ID: " + uniqueID)
8
9  // 重新获取动态 IP
10 调用命令(pfnTryGetDynamicIP)
11
12 // 测试端口联通性
13 调用命令(pfnTryTestConnectivity)
14
15 // 获取本地 IP
16 localIP = 调用命令(pfnTryGetLocalIP)
17 调试输出 ("Local IP: " + localIP)

```

• Delphi 接入示例

(1)、引用 windows 标准库

```

1  uses
2    Windows, SysUtils;

```

(2)、加载 dll:

```

1  var
2    hInstLibrary: HMODULE;
3
4  hInstLibrary := LoadLibrary('./GameshieldManager.dll');
5  if hInstLibrary = 0 then
6  begin
7    Writeln('Unable to load DLL!');
8    Exit;
9  end;

```

(3)、声明函数指针类型:

```

1  type
2    TGameshieldManager_Start = procedure(sdkkey, recommendCode: PAnsiChar); stdcall;
3    TGameshieldManager_GetUniqueID = function(): PAnsiChar; stdcall;
4    TGameshieldManager_TryGetDynamicIP = procedure(); stdcall;
5    TGameshieldManager_TryTestConnectivity = procedure(); stdcall;
6    TGameshieldManager_TryGetLocalIP = function(): PAnsiChar; stdcall;

```

(4)、加载游戏盾 dll 函数:

```

1  var
2    pfnStart: TGameShieldManager_Start;
3    pfnGetUniqueID: TGameShieldManager_GetUniqueID;
4    pfnTryGetDynamicIP: TGameShieldManager_TryGetDynamicIP;
5    pfnTryTestConnectivity: TGameShieldManager_TryTestConnectivity;
6    pfnTryGetLocalIP: TGameShieldManager_TryGetLocalIP;
7
8  pfnStart := GetProcAddress(hInstLibrary, 'GameShieldManager_Start');
9  pfnGetUniqueID := GetProcAddress(hInstLibrary, 'GameShieldManager_GetUniqueID');
10 pfnTryGetDynamicIP := GetProcAddress(hInstLibrary, 'GameShieldManager_TryGetDynamicIP');
11 pfnTryTestConnectivity := GetProcAddress(hInstLibrary, 'GameShieldManager_TryTestConnectivity');
12 pfnTryGetLocalIP := GetProcAddress(hInstLibrary, 'GameShieldManager_TryGetLocalIP');
13
14 if not Assigned(pfnStart) then
15 begin
16   Writeln('Function GameShieldManager_Start not found!');
17   FreeLibrary(hInstLibrary);
18   Exit;
19 end;

```

(5)、加载游戏盾 dll，调用开启函数，代码示例：

```

1  var
2    sdkkey: AnsiString;
3    recommendCode: AnsiString;
4
5  sdkkey := 'your-sdk-key';           // 必填项，需要传入的 SDK 密钥（后台生成查看）
6  recommendCode := 'you-recommend-Code'; // 非必填，校验实例的推荐码（传空则不校验）
7
8  // 调用函数
9  pfnStart(PAnsiChar(sdkkey), PAnsiChar(recommendCode));

```

(6)、其他相关 API 接口，代码示例

```

1  var
2    uniqueID: PAnsiChar;
3    localIP: PAnsiChar;
4
5  // 获取设备唯一 ID 函数
6  uniqueID := pfnGetUniqueID();
7  Writeln('Unique ID: ', uniqueID);
8
9  // 重新获取动态 IP
10 pfnTryGetDynamicIP();
11
12 // 测试端口联通性
13 pfnTryTestConnectivity();
14
15 // 获取本地 IP
16 localIP := pfnTryGetLocalIP();
17 Writeln('Local IP: ', localIP);

```

• **Unity 项目接入示例**

- (1)、将 32 位和 64 位的 GameShield.dll 文件分别放到 Plugins/GameShield/x86 或 x64 目录下
- (2)、在连接服务器前开启游戏盾（建议在程序运行后就可开启），代码示例：

```
1 [DllImport("GameShield.dll", CallingConvention = CallingConvention.Cdecl)]
2 private static extern void GameshieldManager_Start(string sdkkey, string recommendCode);
3
4 // 必填项, 需要传入的 SDK 密钥 (后台生成查看)
5 string sdkkey = "your-sdk-key";
6 // 非必填, 校验实例的推荐码 (传空则不校验)
7 string recommendCode = "you-recommend-Code";
8 // 开启游戏盾
9 GameshieldManager_Start(sdkkey, recommendCode);
```

四、Android 应用接入

1、接入方式

Android 的 SDK 接入方法为 arr 模式, 需要将 gameshield.arr 集成入项目中并在开启时调用。

2、需获取设备存储权限 (通用步骤)

```
1 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
2 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

3、SDK 接入示例

以下提供 Android Studio 和 unity 项目的实例 其他类型的同理。

如果有接入的问题欢迎联系我们的客服, 将协助您进行接入。

• Android Studio 示例

(1)、将 gameshield.aar 文件放到工程 libs 目录下

(2)、在 app 级别的 build.gradle 中添加对 libs 目录的引用, 并包含 gameshield.aar 作为依赖项

```
1 repositories {
2     flatDir{
3         dirs 'libs'
4     }
5 }
6 implementation(name: 'gameshield', ext: 'aar')
```

(3)、执行“File/Sync Project with Gradle Files”同步配置

(4)、在连接服务器前开启游戏盾 (建议在程序运行后就可开启), 代码示例:

```

1 import com.beiguard.gameshield.GameshieldManager;
2
3 // SDK KEY
4 String sdkKey = "your-sdk-key";
5 // 推荐码 (非必填), 若没有可以不传该参数
6 String recommendCode = "you-recommend-code";
7 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
8 int result = GameshieldManager.start(this, sdkKey, recommendCode);
9
10 //如果程序开启混淆, Proguard 增加规则
11
12 //保持 tlschannel 的实现类不被混淆
13 //-keep class tlschannel.** { *; }
14 // 保持 java9.util 的实现类不被混淆
15 //-keep class java9.util.** { *; }

```

(5)、在程序结束前释放游戏盾, 代码示例:

```

1 // 关闭游戏盾
2 GameshieldManager.stop();

```

• Unity 项目接入示例

(1)、将 gameshield.aar 文件放到 Plugins/Android 目录下

(2)、在连接服务器前开启游戏盾 (建议在程序运行后就可开启), 代码示例:

```

1 // SDK KEY
2 string sdkKey = "your-sdk-key";
3 // 推荐码 (非必填), 若没有可以不传该参数
4 string recommendCode = "you-recommend-code";
5 // 需传入 UnityPlayer 当前 Activity
6 AndroidJavaClass unityPlayer = new AndroidJavaClass("com.unity3d.player.UnityPlayer");
7 AndroidJavaObject currentActivity = unityPlayer.GetStatic<AndroidJavaObject>("currentActivity");
8 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
9 AndroidJavaClass gameshieldManager = new
    AndroidJavaClass("com.beiguard.gameshield.GameshieldManager");
10 int result = gameshieldManager.CallStatic<int>("start", currentActivity, sdkKey, recommendCode);
11
12 //如果程序开启混淆, Proguard 增加规则
13
14 //保持 tlschannel 的实现类不被混淆
15 //-keep class tlschannel.** { *; }
16 // 保持 java9.util 的实现类不被混淆
17 //-keep class java9.util.** { *; }

```

(3)、在程序结束前释放游戏盾, 代码示例:

```

1 // 关闭游戏盾
2 AndroidJavaClass gameshieldManager = new
    AndroidJavaClass("com.beiguard.gameshield.GameshieldManager");
3 gameshieldManager.CallStatic("stop");

```

4、其他项目接入示例 (补充中)

5、开放接口 API (前端)

```
1 // 1.开启游戏盾
2 public static int start(Context context, String sdkKey)
3 // 若带参数 recommendCode
4 public static int start(Context context, String sdkKey, String recommendCode)
5 // 2.关闭游戏盾
6 public static void stop()
7 // 3.获取设备唯一码 (游戏盾专用)
8 public static String getUniqueID()
9 // 4.重新请求调度 IP (调度间隔 10s)
10 public static void tryGetDynamicIP()
11 // 5.检查服务器端口连通性 (发现不通时会立刻触发调度)
12 public static void tryTestConnectivity()
13 // 6.获取用户本地 IP (前端取值)
14 public static String tryGetLocalIP()
```

五、IOS 应用接入

1、接入方法

ios 的 SDK 接入方法为 Frameworks 模式，需要将 GameShield.framework 集成入项目中并在开启时调用。

2、接入示例

以下提供原生 Objective-C 和 unity 项目的实例 其他类型的同理。

如果有接入的问题欢迎联系我们的客服，将协助您进行接入。

● Objective-C 示例

(1)、选择 "File/Add Files to"将 GameShield.framework 文件放到项目目录 Frameworks 下

(2)、在弹出的对话框中选择"Copy items if needed"

(3)、在 Build Settings 中关闭 Enable Bitcode

(4)、代码中引入头文件

```
1 #import <GameShield/GameshieldManager.h>
```

(5)、在连接服务器前开启游戏盾 (建议在程序运行后就可开启)，代码示例：

```
1 // SDK KEY
2 NSString *sdkKey = @"your-sdk-key";
3 // 推荐码 (非必填), 若没有可以不传该参数
4 NSString *recommendCode = @"you-recommend-code";
5 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
6 int32_t result = [[GameshieldManager sharedInstance] start:[sdkKey UTF8String]
    recommendCode:[recommendCode UTF8String]];
```

(6)、在程序结束前释放游戏盾，代码示例：

```
1 // 关闭游戏盾
2 [[GameshieldManager sharedInstance] stop];
```

• Unity 项目接入示例

(1)、将 GameShield.framework 文件放到 Plugins/iOS 目录下

(2)、在连接服务器前开启游戏盾（建议在程序运行后即可开启），代码示例：

```
1 [DllImport("__Internal")]
2 private static extern int GameshieldManager_start(string sdkKey);
3 // private static extern int GameshieldManager_startWithCode(string sdkKey, string recommendCode);
4
5 // SDK KEY
6 string sdkKey = "your-sdk-key";
7 // 推荐码 (非必填), 若没有可以不传该参数
8 string recommendCode = "you-recommend-code";
9 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
10 int result = GameshieldManager_start(sdkKey);
11 // 若带参数 recommendCode
12 // int result = GameshieldManager_startWithCode(sdkKey, recommendCode);
```

(3)、在程序结束前释放游戏盾，代码示例：

```
1 [DllImport("__Internal")]
2 private static extern void GameshieldManager_stop();
3
4 // 关闭游戏盾
5 GameshieldManager_stop();
```

3、其他项目接入示例（补充中）

4、开放接口 API（前端）

OC 示例：

```
1 // 开启游戏盾
2 - (int32_t)start:(const char *)sdkKey;
3 // 若带参数 recommendCode
4 - (int32_t)start:(const char *)sdkKey recommendCode:(const char *)recommendCode;
5 // 关闭游戏盾
6 - (void)stop;
7 // 获取设备唯一码（游戏盾专用）
8 - (const char *)getUniqueID;
9 // 重新请求调度 IP（调度间隔 10s）
10 - (void)tryGetDynamicIP;
11 // 检查服务器端口连通性（发现不通时会立刻触发调度）
12 - (void)tryTestConnectivity;
13 // 获取用户本地 IP（前端取值）
14 - (const char *)tryGetLocalIP;
```

Unity 示例：

```
1 // 1.开启游戏盾
2 extern "C" int GameshieldManager_start(const char *sdkKey)
3 // 若带参数 recommendCode
4 extern "C" int GameshieldManager_startWithCode(const char *sdkKey, const char *recommendCode)
5 // 2.关闭游戏盾
6 extern "C" void GameshieldManager_stop()
7 // 3.获取设备唯一码 (游戏盾专用)
8 extern "C" const char * GameshieldManager_getUniqueID()
9 // 4.重新请求调度 IP (调度间隔 10s)
10 extern "C" void GameshieldManager_tryGetDynamicIP()
11 // 5.检查服务器端口连通性 (发现不通时会立刻触发调度)
12 extern "C" void GameshieldManager_tryTestConnectivity()
13 // 6.获取用户本地 IP (前端取值)
14 extern "C" const char * GameshieldManager_tryGetLocalIP()
```

六、MacOS 应用接入

1、接入方法

Mac 的 SDK 接入方法为 dylib/framework 模式，需要将 GameShield.dylib/framework 集成入项目中并在开启时调用。

2、接入示例

以下提供原生 Objective-C 和 unity 项目的实例 其他类型的同理。

如果有接入的问题欢迎联系我们的客服，将协助您进行接入。

• Objective-C 示例

- (1)、将 GameShield.dylib 文件放到项目根目录 Frameworks 下
- (2)、在弹出的对话框中选择"Copy items if needed"
- (3)、在 Build Settings 中关闭 Enable Bitcode
- (4)、代码中引入头文件

```
1 #import <GameShield/GameshieldManager.h>
```

- (5)、在连接服务器前开启游戏盾 (建议在程序运行后就可开启)，代码示例：

```
1 // SDK KEY
2 NSString *sdkKey = @"your-sdk-key";
3 // 推荐码 (非必填), 若没有可以不传该参数
4 NSString *recommendCode = @"you-recommend-code";
5 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
6 int32_t result = [[GameshieldManager sharedInstance] start:[sdkKey UTF8String] recommendCode:[recommendCode UTF8String]];
```

- (6)、在程序结束前释放游戏盾，代码示例：

```
1 // 关闭游戏盾
2 [[GameshieldManager sharedInstance] stop];
```

• Unity 项目接入示例

- (1)、将 GameShield.dylib 文件放到 Plugins/GameShield 目录下
- (2)、在连接服务器前开启游戏盾（建议在程序运行后就可开启），代码示例：

```
1 [DllImport("GameShield", CallingConvention = CallingConvention.Cdecl)]
2 private static extern int GameshieldManager_start(string sdkKey);
3 // private static extern int GameshieldManager_startWithCode(string sdkKey, string recommendCode);
4
5 // SDK KEY
6 string sdkKey = "your-sdk-key";
7 // 推荐码 (非必填), 若没有可以不传该参数
8 string recommendCode = "you-recommend-code";
9 // 开启游戏盾 result 为 1 表示开启成功 其他为开启失败
10 int result = GameshieldManager_start(sdkKey);
11 // 若带参数 recommendCode
12 // int result = GameshieldManager_startWithCode(sdkKey, recommendCode);
```

- (3)、在程序结束前释放游戏盾，代码示例：

```
1 [DllImport("GameShield", CallingConvention = CallingConvention.Cdecl)]
2 private static extern void GameshieldManager_stop();
3
4 // 关闭游戏盾
5 GameshieldManager_stop();
```

3、开放接口 API (前端)

OC 示例:

```
1 // 开启游戏盾
2 - (int32_t)start:(const char *)sdkKey;
3 // 若带参数 recommendCode
4 - (int32_t)start:(const char *)sdkKey recommendCode:(const char *)recommendCode;
5 // 关闭游戏盾
6 - (void)stop;
7 // 获取设备唯一码 (游戏盾专用)
8 - (const char *)getUniqueID;
9 // 重新请求调度 IP (调度间隔 10s)
10 - (void)tryGetDynamicIP;
11 // 检查服务器端口连通性 (发现不通时会立刻触发调度)
12 - (void)tryTestConnectivity;
13 // 获取用户本地 IP (前端取值)
14 - (const char *)tryGetLocalIP;
```

Unity 示例:

```
1 // 1.开启游戏盾
2 extern "C" int GameshieldManager_start(const char *sdkKey)
3 // 若带参数 recommendCode
4 extern "C" int GameshieldManager_startWithCode(const char *sdkKey, const char *recommendCode)
5 // 2.关闭游戏盾
6 extern "C" void GameshieldManager_stop()
7 // 3.获取设备唯一码 (游戏盾专用)
8 extern "C" const char * GameshieldManager_getUniqueID()
9 // 4.重新请求调度 IP (调度间隔 10s)
10 extern "C" void GameshieldManager_tryGetDynamicIP()
11 // 5.检查服务器端口连通性 (发现不通时会立刻触发调度)
12 extern "C" void GameshieldManager_tryTestConnectivity()
13 // 6.获取用户本地 IP (前端取值)
14 extern "C" const char * GameshieldManager_tryGetLocalIP()
```