

TopVPN API 使用说明

目录

一、	SDK 简要介绍	2
二、	SDK 运行要求	2
三、	功能说明	3
四、	集成前准备	3
1.	准备测试环境.....	3
2.	ANDROID 平台准备.....	4
3.	IOS 平台准备.....	5
五、	VPN 集成步骤	7
1.	API 调用的基本流程.....	7
2.	补充说明.....	10
六、	API 说明	12
1.	主要数据结构介绍	12
2.	主要 API 类和接口说明.....	13
3.	API 接口详细说明.....	14
七、	附录	20
八、	备注	20
九、	常见问题及解答	21
1.	一般性问题.....	21
2.	ANDROID 平台问题.....	22
3.	IOS 平台问题.....	22
十、	注意事项	22

一、 SDK 简要介绍

考虑到现在用户开发自己的应用的需求，为了更大的发挥 VPN 网关功能来满足客户需求，我们提供 SSLVPN 协议的 SDK 开发包供客户进行二次开发。

为了方便用户使用，本 SDK 开发采用跨平台开发，目前支持 IOS 和 Android 系统。不同系统不同语言，但是接口一致。因此开发者调用 API 的步骤都是一样的。考虑到各种原因，所有 API 调用都为异步操作（每次 API 的调用都是一次异步操作），除个别例外。

每次升级只需进行简单的库文件的替换，不会因为 SDK 的功能升级（平台和 SDK 工作模式的增加）而需要用户修改额外代码。

SDK 包附带有示例 Demo，提供了最基本的 API 演示。在集成之前，请在 Demo 写代码进行测试。

二、 SDK 运行要求

1. VPN 网关要求

- 目前支持 VPN 网关版本 SM2 和 020 版本
- 不支持防火墙版本
- 针对很老的版本和不支持的版本如需使用请升级至 SM2 或 020 的最新版本

2. Android 平台

- 目前支持 4.0~9.x 系列系统。从 6.0 开始 HttpClient 因 Google 已将其从系统移除，默认不支持，因此我们也不支持。
- 我们只支持 JAVA 层面的网络处理，底层 C 库实现的网络通信不支持。
- 支持 CPU 的指令集包括 arm、arm-v7a，arm64-v8a，其他 CPU 架构不支持，其中 arm64-v8a 只支持国际协议国际算法，不支持国密。
- 支持 32 位系统和 64 位系统。若项目只支持 32 位系统，需在配置文件中指定只加载 32 位架构相关的库。

3. IOS 平台

- 支持 IOS9.0 以上任何系列设备，其中 IPV6 功能需要 iOS11.0 以后方可支持
- 支持 CPU 的指令集包括 arm、armv7s、arm64 和 arm64e，不支持 i386 和 x86_64 架构的 CPU，即 SDK 不支持模拟器，因此测试请使用真机环境
- 要求 APP 和其插件启用 NetworkExtension，即新建 BundleID 时勾选 NetworkExtension 的功能
- APP 的 BundleID 必须为其插件的 BundleID 的子串

三、 功能说明

SDK 基本功能介绍：

- 1 目前支持全网接入模块。
- 2 用户认证支持口令认证、证书认证和双因子认证，其中证书认证的只支持国际协议国际算法，不支持国密协议国密算法。
- 3 支持修改密码和找回密码功能，前提需网关配置允许，配置问题请联系管理员
- 4 允许长时在线。

四、 集成前准备

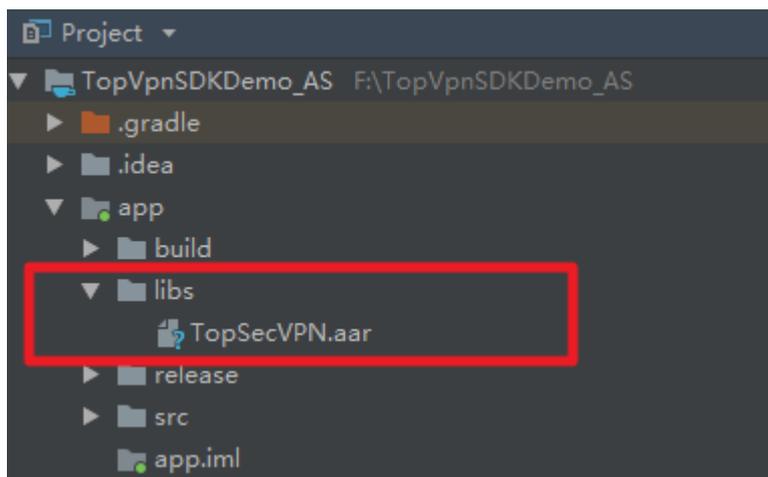
1. 准备测试环境

在使用之前请配置好用户和资源，确定 VPN 地址和用户可用，一切正常。如果要在 SDK 中使用同一 VPN 帐号，请给用户配置多点登录功能。资源的配置请配置全网接入类型的资源，针对资源配置方面有问题请联系管理员，如果贵单位管理员不知道，请天融信的客服或相关销售。

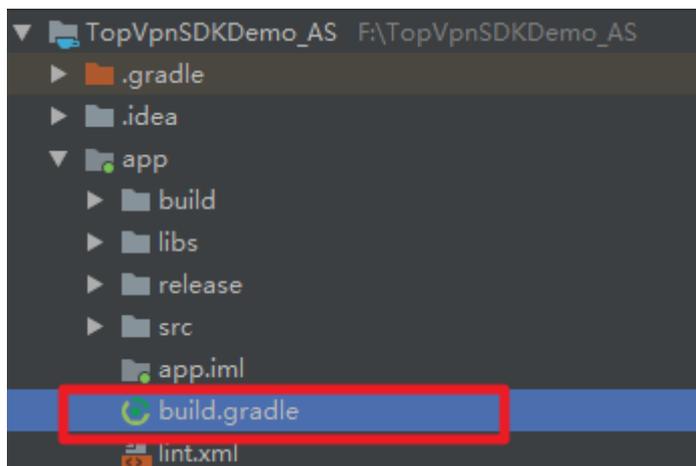
2. Android 平台准备

鉴于目前 Android 开发较多使用谷歌提供的集成开发环境 Android Studio,所以对 Android 平台提供 aar 库。SDK 中所包含示例 Demo 也为 Android Studio 环境下开发。故在开发时建议使用集成开发环境 Android Studio。在项目中集成 SDK 步骤如下:

- a) 将 SDK 中 TopSecVPN.aar 拷贝至项目中 libs 目录下。



- b) 修改项目中 app 目录下 build.gradle(aar 中包含 32 位库与 64 位库,若项目只支持 32 位架构,请在配置文件中指定使用 32 位库)



```
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation(name: 'TopSecVPN', ext: 'aar')
```

```
defaultConfig {
    applicationId "com.example.topsec.sslvpn.topvpnsdkdemo"
    minSdkVersion 8
    targetSdkVersion 26 // 指定加载架构的库，若只支持32位，需将arm64-v8a去掉
}

ndk {
    abiFilters 'armeabi', 'armeabi-v7a', 'arm64-v8a' // 指定需要加载的架构
}
```

c) 修改完成后 Sync 即可。

3. iOS 平台准备

为了方便后面的阅读，这里先对以下名称等信息做下解释，以避免一些用户后面看不明白，存有疑惑。如果您已经知道可以忽略下面这一段，跳过阅读即可。

a) 什么是扩展插件，我需要创建吗？

答：我们已经提供，您不需要插件。在这里我们指的是 SDK SDK 带的文件 “NetAccessProvider.appex”，其他的您不需要管。

b) 上面是 BundleID，什么是 AppID？

答：BundleID 是 Bundle identifier 的缩写，AppID 同理可知，两者可以理解为同一个。

c) 为什么 Demo 中的 BundleID 和实际要求的不一致？

答：为了规范调用，对外我们扩展插件（NetAccessProvider.appex）的 AppID 做了要求。这个请您不要纠结，直接忽略即可。

d) 什么是描述文件？

答: iOS Provisioning Profiles 可以称之为授权描述文件、授权文件或描述文件

e) 什么是代码重签名?

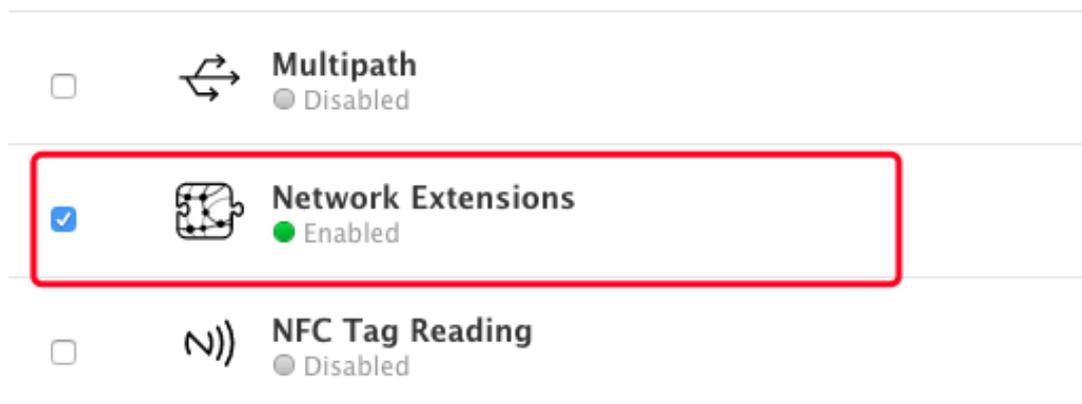
答: 一般您们是不需要了解的, 您只需要按要求进行过重签名, 问题都会解决。如果非常感兴趣请自己动手去查, 答案更全, 这里不做解释。

f) 为什么重新签名的截图中传入的描述文件名称为 TopTest.mobileprofile, 而事件的框架库中是 embedded.mobileprovision, 需要我们手动修改吗?

答: 您们不需要修改, 重签名程序会自动完成。

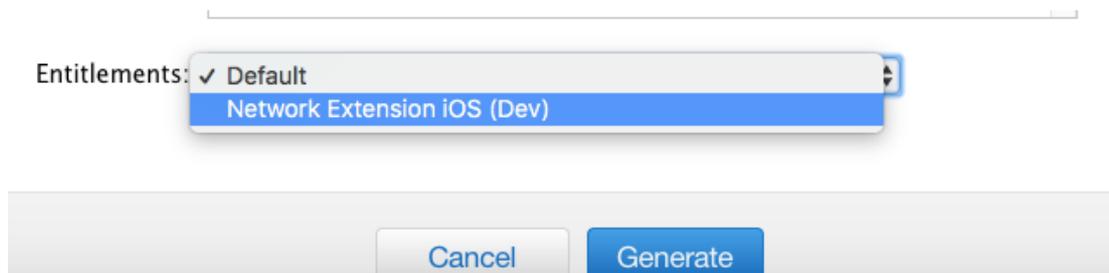
下面是我们 SDK 重签名需要的一些准备工作:

- 请准备一张签名证书, 并把它导入到钥匙串中, 主应用的签名、SDK 重签名和创建苹果授权文件(描述文件)时会用到
- 打开钥匙串, 获取签名证书的名称
- 确定主应用和插件的 BundleID, 其中插件的 BundleID 默认为主应用的 BundleID 加上 “.NetAccessProvider”
- 创建或修改主应用的 AppID, 开启其 NetworkExtension 功能



e) 创建插件的 AppID, 其 ID 必须是主应用的 AppID+ “.NetAccessProvider”, 并开通 NetworkExtension 功能 (开启方法同上)

f) 创建主应用 (已有可以更新) 和插件的授权文件, 此时选中该证书和对应的 AppID, 并选中 Entitlements 项目中的 Network Extension。(默认不选中时为 Default)



- g) 导出插件的苹果授权文件(描述文件)
- h) 把签名工具、SDK、插件的描述文件放在同一个目录下
- i) 打开终端，跳转到 SDK 所在目录“cd [SDK 目录]”
- j) 执行签名命令 “./AppResignTool [证书名称或标识] [主应用的 BundleID] [插件的描述文件名称]”

```

HECHAOdeMac-mini:Debug hechao$ ./AppResignTool 670E2A70B621047C9DB64A853F063F0F40790430 com.topsec.com.TopTest.mobileprovision
注意：由于我们没有对描述文件所用证书和您提供的签名证书进行一致性校验，可能会导致在对APP签名时出现失败，因此请你们自己确保所用证书一致。
完成重签名库： NetAccessProvider
真BundleID为： com.topsec.com.NetAccessProvider
真版本为： 3.4.0.14.0.5
真支持的最低系统版本为： 9.0

完成重签名库： TopSecVPN
真BundleID为： com.topsec.com.TopSecVPN
真版本为： 2.4.9
真支持的最低系统版本为： 8.0
    
```

- k) 请在主应用的项目中导入 SDK 库,禁止拖拽



- l) 项目中引入 SDK 的头文件 “#import "TopSecVPN/TopSecVPN.h"”

五、 VPN 集成步骤

1. API 调用的基本流程

SDK 的所有接口均默认均为异步接口，接口执行完成后会回调接口

OnAcceptResultListener，请用户根据操作 ID 做对应处理，针对一些情况请开发人员灵活处理。比如：

如果在断开 VPN 的时候只调用 `closeService` 接口关闭服务,之后就没有再调用 `logoutVOne` 接口,此时 VPN 隧道已经正常断开,但是几分钟之后又想重新建立 VPN 隧道时,调用了 `loginVone` 接口或 `setConfigInfo` 接口,此时发现调用失败,报“用户已经登录,操作被拒绝”(错误号: -18)的错误,您可以直接调用 `startService` 接口重新启动服务即可。如果报“服务器验证客户端 SessionID 非法”(错误号: -439)或“用户尚未登录,请登录”(错误号: -19),请调用 `loginVone` 接口重新登录。

API 接口被调用的大致流程如下图所示, :



Figure 1 VPN 工作流程

2. 补充说明

a) Android 平台

特别提示:

- 在 AndroidManifest.xml 中添加 VPN 服务

```

<service android:name="com.topsec.sslvpn.lib.na.NaVpnService"
  android:permission="android.permission.BIND_VPN_SERVICE">
  <intent-filter>
    <action android:name="com.topsec.sslvpn.lib.na.NaVpnService"/>
  </intent-filter>
</service>
  
```

图 1 在 AndroidManifest.xml 中添加对全网接入的服务

- 全网接入服务的启动请使用带参 startService

```

case R.id.btnStartService:
{
    //调用该函数启动全网接入服务
    //第一个参数用于接收用户返回选择结果的Activity，用户需要在该类中实现onActivityResult函数
    //第二个参数用于管理的Activity,Android5以后无法使用，可以不配置
    //TopVPNSdkDemo.m_ihVPNService.startService(this, DemoBrowser.class);
    TopVPNSdkDemo.m_ihVPNService.startService( activity: this, aClass: null);
}
  
```

图 2 启动全网接入

- 在处理返回值的类（startService 第一个参数）中实现 onActivityResult

```

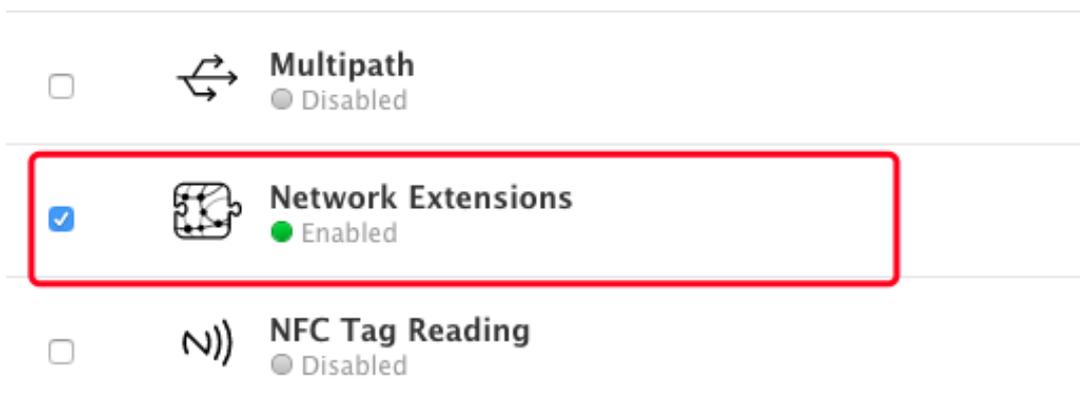
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    //启动VPN时会弹一个授权的对话框，该处用于接收授权的结果，请调用接口“toGrantStartVpnService”告知SDK
    TopVPNSdkDemo.m_ihVPNService.toGrantStartVpnService(resultCode);
    super.onActivityResult(requestCode, resultCode, data);
}
  
```

图 3 实现授权处理

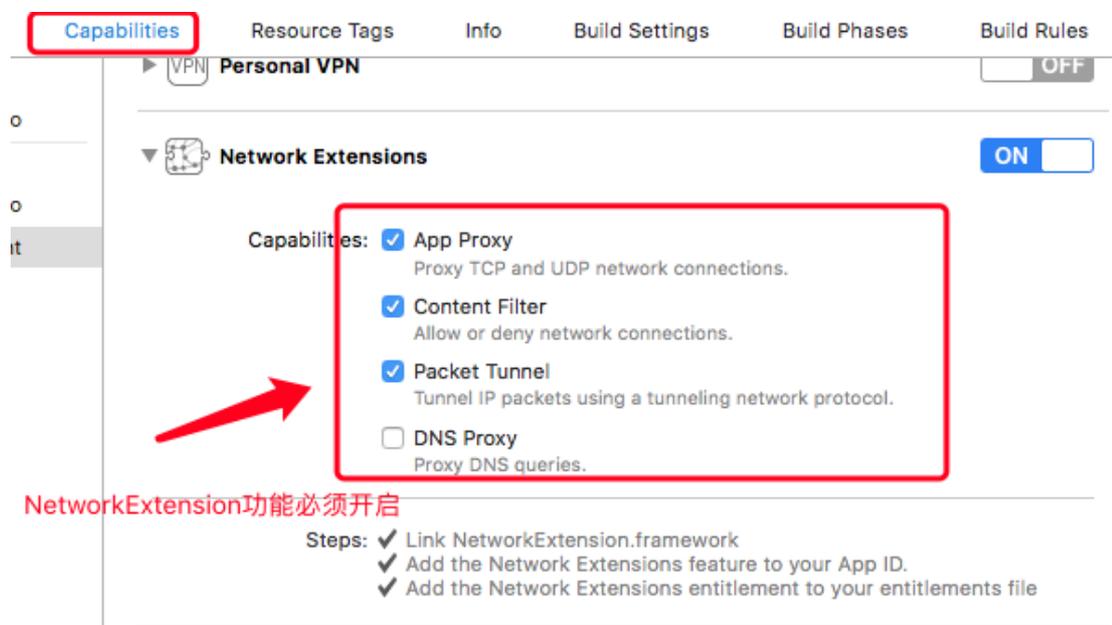
b) iOS 平台

特别提示:

- 新建插件的 APP ID(Bundle ID)时，必须为主 APP 的 APP ID(Bundle ID)后面加上“.NetAccessProvider”，同时启用 NetworkExtension 功能，如下图：



- APP 和插件必须开启 NetworkExtension 功能,开启后在项目-> 功能里面可以看到, 如下图:



- 在添加 SDK 到项目之前必须使用 “AppResignTool” 工具对 SDK 进行重签名

```
HECHAOdeMac-mini:Debug hechao$ ./AppResignTool 670E2A70B621047C9D864A853F063F0F4D790430 com.topsec.com.TopTest.mobileprovision
注意: 由于我们没有对描述文件所用证书和您提供的签名证书进行一致性校验, 可能会导致在对APP签名时出现失败, 因此请您们自己确保所用证书一致。
完成重签名库: NetAccessProvider
其BundleID为: com.topsec.com.NetAccessProvider
其版本为: 3.4.0.14.0.5
其支持的最低系统版本为: 9.0
完成重签名库: TopSecVPN
其BundleID为: com.topsec.com.TopSecVPN
其版本为: 2.4.9
其支持的最低系统版本为: 8.0
```

具体操作请参照 SDK 附带的 Demo 示例。

六、 API 说明

1. 主要数据结构介绍

表格 1 BaseModule (工作模式)数据说明

属性	类型	说明
SSLVPN_PORTFORWARDING	Int	端口转发模块（已废弃，不推荐）
SSLVPN_NETACCESS	Int	全网接入模块（请使用该模块）
SSLVPN_WEBFORWARDING	Int	Web 转发模块(尚未实现)

表格 2 WorkModel(工作模式) 数据说明

属性	类型	说明
WORKMODE_DEFAULT	Int	默认模式
WORKMODE_PORTMAP	Int	端口映射方式
WORKMODE_MASTER	Int	主动模式
WORKMODE_SLAVE	Int	被动模式（主从模式使用）
WORKMODE_SM2	Int	支持 SM 的服务器
WORKMODE_020	Int	支持 020 的服务器

表格 3 BaseConfigInfo (VPN 配置)数据说明

属性	类型	说明
m_blAutoReConnect	boolean	网络正常时是否重连
m_iLogLevel		日志级别，发布版本请关闭日志
m_iTimeOut	Int	超时时长
m_iRetryCount	Int	失败后重试次数
m_strVPNIP	String	VPN 服务器 IP 地址
m_iServerPort	Int	服务器提供服务的端口
m_iEnableModule	Int	设置开启的功能模块
m_iWorkMode	Int	工作模式
m_strExchangeData	String	用于数据交换，数据由主 SDK 提供
m_clsPendingCls	Class<?>	配置页面 Activity, 在 4.0 系列有效(可以为空)

m_aMainActivity	Activity	用于返回用户授权结果的接收 Acitivity
-----------------	----------	-------------------------

表格 4 BaseAccountInfo

属性	类型	说明
m_iLoginType	Int	登录类型(口令、证书和双因子)
m_iAuthType	Int	验证类型（无，软证书和硬证书）
m_iProtocolType	Int	协议类型：国密协议或国际协议
m_strAccount	String	登录用户名
m_strCerPath	String	软证文件名（路径）
m_strLoginPasswd	String	用户登录密码
m_strCerPasswd	String	软证的证书密码
m_iExtraCodeType	Int	附加码类型
m_strExtraCode	String	附加码
m_strPhoneFeatureCode	String	手机特征码
m_strPackageName	String	当前程序包名

2. 主要 API 类和接口说明

本 SDK 主要包括 3 个接口和 2 个类：

表格 5 SDK 类和接口

接口	类型	说明
IVPNService	interface	VPNSDK 接口
onAcceptSysLogInfo	interface	SDK 日志输出回调接口
OnAcceptResultListener	interface	SDK 每一个操作执行结果回调接口
VPNService	Class	IVPNService 实现类，必须实例化
VPNStaus	Class	VPN 状态检测

3. API 接口详细说明

- onAcceptSysLogInfo 接口说明

表格 6 onAcceptSysLogInfo 函数

函数	onAcceptSysLogInfo		
说明	实现该接口可接收系统抛出的日志		
返回值	无		
参数说明	iLogLevel	int	日志级别
	strLogTag	String	日志的标题
	strLogInfo	String	具体的日志内容

- OnAcceptResultListener 接口说明

表格 7 OnAcceptResultListener 函数

函数	OnAcceptResultListener		
说明	实现该接口用于接收执行结果		
返回值	无		
参数说明	iOperationID	int	操作 ID
	iRetVal	int	返回结果
	objExtraValue	Object	附加信息
	objReserved	Object	预留，如果失败，这里会返回服务器返回的错误代码

- IVPNService 接口说明

接口默认均异步执行，除函数名称InSyncMode的除外，一般地执行结果都通过 OnAcceptResultListener返回

表格 8 setConfigInfo 函数

函数	setConfigInfo		
说明	设置 VPN 配置信息		
返回值	调用成功返回 0		
参数说明	bcicfgInfo	BaseConfigInfo	配置信息

表格 9 loginVOne 函数

函数	loginVOne		
说明	登录 VPN 系统		
返回值	调用成功返回 0		
参数说明	baiActInfo	BaseAccountInfo	账户信息

表格 10 kickTheCurrentUserOffImmediately 函数

函数	kickTheCurrentUserOffImmediately		
说明	踢出当前用户并继续登录系统		
返回值	调用成功返回 0		
参数说明	无		

表格 11 continueToLoginWithExtraCode 函数

函数	continueToLoginWithExtraCode		
说明	使用附加码继续登录		
返回值	调用成功返回 0		
参数说明	strExtraCode	String	附加码
	ectExtraCodeType	eExtraCodeType	附加码类型

表格 12 logoutVOne 函数

函数	logoutVOne		
说明	退出 VPN 系统		
返回值	调用成功返回 0		
参数说明	无		

表格 13 startService 函数

函数	startService		
说明	启动 VPN 服务— 已废弃 ，请使用表 14 中接口		
返回值	调用成功返回 0		
参数说明	无		

表格 14 startService 函数

函数	startService		
说明	用于全网接入模块的 VPN 服务启动		
返回值	调用成功返回 0		
参数说明	aMainActivity	Activity	用于启动授权页面以及接收返回值页面(实现onActivityResult)
	clsCfgActivityCls	Class<?>	配置页类 4.0 系统VPN管理菜单“配置”实现 实现'配置',需要提供一个配置页,可以为空

表格 15 stopService 函数

函数	stopService
说明	暂停 VPN 服务 (暂时不可用)
返回值	调用成功返回 0
参数说明	无

表格 16 closeService 函数

函数	closeService
说明	关闭 VPN 服务
返回值	调用成功返回 0
参数说明	无

表格 17 requestVPNResInfo 函数

函数	requestVPNResInfo
说明	请求 VPN 资源
返回值	调用成功返回 0
参数说明	无

表格 18 requestCaptcha 函数

函数	requestCaptcha
----	----------------

说明	请求登录验证码
返回值	调用成功返回 0
参数说明	无

表格 19 queryVPNRunningState 函数

函数	queryVPNRunningState
说明	查询 VPN 状态,结果异步返回
返回值	调用成功返回 0
参数说明	无

表格 20 queryVPNRunningStateInSyncMode 函数

函数	queryVPNRunningStateInSyncMode
说明	同步模式下获取 VPN 状态
返回值	大于 0 为 VPN 的状态, 可通过 VPNStatus 进一步判断, 其他为错误
参数说明	无

表格 21 setOnAcceptResultListener 函数

函数	setOnAcceptResultListener		
说明	添加执行结果的事件侦听, 用于接收执行结果		
返回值	调用成功返回 0		
参数说明	oarSrc	OnAcceptResultListener	已实现事件

表格 22 setOnAcceptSysLogListener 函数

函数	setOnAcceptSysLogListener		
说明	添加日志的事件侦听, 用于接收日志, 用于可以选择过滤记录日志, 自定义显示		
返回值	调用成功返回 0		
参数说明	oasSrc	setOnAcceptSysLogListener	已实现事件

表格 23 getErrorInfoByCode 函数

函数	getErrorInfoByCode
----	--------------------

说明	调用该接口可查询错误原因		
返回值	调用成功返回错误原因		
参数说明	iErrorCode	Int	错误代码（服务器返回的和 VPN 系统的）

表格 24 getExchangeDataFromMode 函数

函数	getExchangeDataFromMode		
说明	获取模式交换数据，被动模式设置VPN参数时用到		
返回值	(String)指定模式下可以返回的数据，如果没有返回null		
参数说明	iWorkModel	Int	工作模式（WORKMODE_MASTER）

表格 25 getCertificateContentInSyncMode 函数

函数	getCertificateContentInSyncMode		
说明	登录后允许获取证书内容		
返回值	证书内容，失败返回null		

函数	modifyPassword		
说明	异步方式修改密码，执行完成后会调用回调接口返回		
返回值	(int)调用成功返回0		
参数说明	strNewPasswd	String	新密码
参数说明	strOldPasswd	String	旧密码

函数	modifyPasswordInSyncMode		
说明	同步模式修改密码，执行结果立即返回，请在线程中调用		
返回值	(int)执行成功返回0，说明密码修改成功，否则失败		
参数说明	strNewPasswd	String	新密码
参数说明	strOldPasswd	String	旧密码

函数	retrievePassword		
说明	异步方式找回密码，执行完成后会调用回调接口返回		
返回值	(int)调用成功返回0		
参数说明	strUserName	String	用户名
参数说明	strCapatcha	String	验证码

函数	retrievePasswordInSyncMode		
说明	同步模式找回密码，执行结果立即返回，请在线程中调用		
返回值	(int)执行成功返回0，说明找回成功，否则失败		
参数说明	strUserName	String	用户名
参数说明	strCapatcha	String	验证码

- VPNService 类说明

VPNService是IVPNService 接口的实现，因此接口基本是一样的。由于SDK采用单例模式，因此额外提供一个接口以便获取VPN实例。

表格 26 getVPNInstance 函数

函数	getVPNInstance		
说明	成功构造的 VPN 包实例		
返回值	VPN 包实例， IVPNService 类型		
参数说明	appContext	Context	系统上下文

- VPNStatus 类说明

表格 27 IsUserLoggedin 函数

函数	IsUserLoggedin		
说明	检测用户是否登录		
返回值	返回 true 则说明用户已经登录		
参数说明	iStatusValue	Int	VPN 状态值

表格 28 IsVPNServiceRunning 函数

函数	IsVPNServiceRunning		
说明	检测 VPN 服务是否正在运行		
返回值	返回 true 则说明服务运行正常		
参数说明	iStatusValue	Int	VPN 状态值

其他说明：

- a) Android 平台和 IOS 版本的接口基本是一样的，不做各平台特别说明了；
- b) IOS 的具体说明见头文件(.h)；
- c) 具体示例请看随同的 Demo 实现。

七、 附录

SDK 的操作均为异步操作，所有操作（函数）执行完成后会调用 OnAcceptResultListener 接口。下表是一些常用操作值及其说明：

表格 29 常用操作值及其说明

具体操作	标识 ID	备注	
获取服务器配置	OPERATION_GET_SERVERCFG	成功返回 0，objExtraValue 返回配	
获取验证码	OPERATION_GET_CAPTCHA	成功返回 0，objExtraValue 返回验	
登 录	账户验证	OPERATION_AUTH_LOGININFO	成功返回 0
	上传特征 码	OPERATION_UPLOAD_FEATURECODE	成功返回 0
	登录	OPERATION_LOGIN_SYSTEM	成功返回 0
登出	OPERATION_LOGOUT_SYSTEM	成功返回 0	
获取资源	OPERATION_GET_RESOURCE	成功返回 0，objExtraValue 返回资	
启动 VPN 服务	OPERATION_START_SERVICE	成功返回 0	
关闭 VPN 服务	OPERATION_CLOSE_SERVICE	成功返回 0	
获取 VPN 状态	OPERATION_GET_KEEPSTATUS	返回值可以通过 VPNStaus 类检	
修复 VPN 隧道	OPERATION_TRYFIX_VPNTUNNEL	成功返回 0，否则返回失败原因	

八、 备注

1. 所有异步接口调用结果都会在 onAcceptExecResultListener 中返回；
2. 所有日志均通过 onAcceptSysLogInfo 返回；
3. 刷新验证码请使用 requestCaptcha
4. VPN 运行状态可通过 queryVPNRunningState(异步获取)或

getVPNRunningStateInSyncMode(同步获取)得到

5. 对操作的返回值，成功都是 0，失败返回可以通过 `getErrorInfoByCode` 查询到具体的出错原因
6. 登录失败错误码为-40077 的时候可以进行踢人操作，当错误码为 40039 的时候可以调用 `continueToLoginWithExtraCode` 进行短信认证

九、 常见问题及解答

1. 一般性问题

- a) 集成 SDK 后无法正常访问资源怎么办？

答：在用户服务器运行正常的情况下，建议先使用 PC 端或 Demo 去进行情况核实，核实过程中请使用同一用户。如果情况一样肯定是网关配置异常，请复核网关配置。

- b) 在使用 Demo 测试时，VPN 启动后无法连接资源

答：可能的原因就是没有看提示和文档，看到“成功登入 VPN 系统”就以为完事。我们的 VPN 使用需要登录服务器，获取资源，启动 VPN 服务，用户的任何操作都需要在这些操作成功完成之后才能进行。请认真看下 Demo 登录页下方的提示。

- c) 使用 Demo 里头的 WebBrowser 无法访问 HTTPS 站点

答：WebView 目前我们没做 Https 处理，您可以使用 Socket 编码实现对 Https 站点的访问，这是完全没问题的。

- d) 无法访问资源

答：请确定内网可以正常访问；该资源已配置正确，该用户具有访问该资源的权限，同时保证电脑端在相应的工作模式下能正常访问；在自己的业务进行网络通信前一定要保证 VPN 已经正常启动。

- e) 调用 API 报“用户已经登录，操作被拒绝”

答：出现此类问题，一般出现在接口被重复调用时。原因是 SDK 做了防重复调用的检测。如果 SDK 连续被多次调用，但该操作只会执行真正一次。因此你只需要按流程执行下一步操作或者先退出登录再进行该操作。

2. Android 平台问题

- a) 运行时报: `java.lang.UnsatisfiedLinkError:dlopen failed: "**/*arm/*.so"`
答: 原因很简单, 库没放对。请检查下您的设备 CPU 架构和 SDK 的版本, 保持一致即可。目前没有提供 64 位库, 如果设备是 64 位的, 请修改配置强制加载 32 位库即可解决
- b) 执行 API 接口 `getVPNInstance` 返回 `null`, 即初始化 VPN 实例失败
答: 原因有多种, 第一种情况, 见问题 1. 第二种情况, 权限问题, 如果 APP 权限不够会导致 SDK 初始化失败, 因此请检查下您的 APP 具有 SDK 所需要的所有权限, 尤其是 Android 6 以后个别权限需要动态申请权限。

3. iOS 平台问题

- a) 编译时如果要到错误 (iOS 平台): `[Undefined symbols for architecture armv7: "_OBJC_CLASS_$_ASIdentifierManger", referenced from: objc-class-ref in libTopSecVPN.a]`,
答: 请添加框架 "AdSupport.framework"
- b) 编译失败报: "error: Embedded binary is not signed with the same certificate as the parent app."
答: 原因是插件所用签名证书与 APP 所用证书不一样, 请检查您插件的证书和 APP 的证书是否一致。

十、 注意事项

1. 请现在 Demo 中自己编写测试代码, 测试通过后再做集成;
2. 对 VPN 网关配置方面的问题, 请联系公司客服或相关销售;
3. Demo 实例为功能演示, 仅供参考;