

KOZEN Terminal Management
SDK Development Documentation
V1.7

Shanghai Xiangcheng Communication Technology Co., Ltd.

Table of Contents

- 1. Revision History 8
- 2. Overview 11

2.1 Introduction 11
2.2 Android version and IDE version supported by the SDK 11
2.3 Feature Introduction 11
2.3.1 Terminal Management Module 11
2.3.2 Certificate Management Module 11
2.3.3 Device Management Module 11
2.3.4 Device Information Management Module 11
2.3.5 Location Management Module 12
2.3.6 Network Management Module 12
2.3.7 Resource Management Module 12
2.3.8 Device Log Module 12
2.4 Importing the TerminalManagerService SDK 12
2.5 Initializing the TerminalManagerService SDK 13

- 3. API Interface Introduction 13

➢ 3.1 Terminal manager SDK initialization 14
3.1.1 Initialize the TerminalManagerService instance 14
-- TerminalManagerService instance callback - InitCallBack -- 14
3.1.2 Initialization results 14
➢ 3.2 Certification module 14
-- Get Certification module - getCertificationManager-- 14
3.2.1 Updates the application's signature certificate 15
3.2.2 Deletes the application's signature certificate 15
3.2.3 Get the application's signature certificate information 15
3.2.4 Get the public key from the certificate chain 15
3.2.5 Get the client name from the certificate chain 16
➢ 3.3 Device information module 16
-- Get device information module - getDeviceInfoManager-- 16
3.3.1 Get the SDK service version number 17
3.3.2 Get Terminal serial number 17
3.3.3 Get the IMSI number 17
3.3.4 Get the IMEI number 17
3.3.5 Get the equipment supplier name (XC, Kozen) 17
3.3.6 Get the device model 18
3.3.7 Get the OS version number 18
3.3.8 Get the Linux kernel version number 18
3.3.9 Get the MCU version number 18
3.3.10 Get the hardware version number 18
3.3.11 Get the EMV kernel version number 19
3.3.12 Get TUSN (Only for the China region) 19
3.3.13 Get customer serial number 19

3.3.14 Get the Terminal Management System server address 19
3.3.15 Get the ICCID of the SIM card in the specified slot 19
3.3.16 Get the phone number of the SIM card in the specified slot 20
3.3.17 Get the MAC address of the device's Bluetooth adapter(Deprecated) 20
3.3.18 Get the specified system device information 20
3.3.19 Get Device MAC Address 20
➢ 3.4 Device module 20
-- Get device module - getDeviceManager-- 21
3.4.1 Sets the system time 22
3.4.2 Returns the current system time as a timestamp 22
3.4.3 Sets the time zone 22
3.4.4 Get the current time zone 22
3.4.5 Reboots the device 22
3.4.6 Shuts down the device 23
3.4.7 Schedules a timed reboot(Deprecated) 23
3.4.8 Cancels the scheduled reboot 23
3.4.9 Enable or disable silent installation 23
3.4.10 Enforce runtime permission checks 24
3.4.11 Set the screen timeout duration 24
3.4.12 Turn off the screen (sleep mode) 24
3.4.13 Enable or disable various device wake-up methods 24
3.4.14 Get PCI reboot countdown time 25
3.4.15 Get the configured PCI reboot time 25
3.4.16 Get application permissions 25
3.4.17 Set application permission 25
3.4.18 Check whether developer mode is enabled 26
3.4.19 Synchronize system time from NTP 26
3.4.20 Synchronize time zone from network 26
3.4.21 Sets automatic system time synchronization 26
3.4.22 Sets automatic system time-zone synchronization 27
3.4.23 Show or hide a launcher application icon 27
3.4.24 Get the launcher application list 27
3.4.25 Enable or disable the screen reader 27
3.4.26 Get screen-reader switch state 28
3.4.27 Set the primary-screen wallpaper 28
3.4.28 Check whether an EEPROM battery is supported 28
3.4.29 Get battery charge cycle count 28
3.4.30 Get battery health percentage 29
3.4.31 Read EEPROM battery fields 29
3.4.32 Schedule a timed reboot with mode configuration 29
3.4.33 Set App Auto-Start 30
➢ 3.5 Location module 30
Declaration on Location Interface 30
Our location interface is designed solely to provide location-related services to applications. 30

● It does not actively collect device information such as SN, IMEI, or hardware identifiers. 30
● It does not transmit any device information or location data to external servers without explicit application request and authorization. 30
● All data access is limited to the scope necessary for location service functionality. 30
This ensures that the interface complies with user privacy protection and security requirements. 30
-- Get Location module - getLocationManager-- 30
3.5.1 Enables the location module and initializes related resources 31
3.5.2 Enables the location module and initializes related resources by key 31
3.5.3 Set location parameters 31
3.5.4 Start a single location request 32
3.5.5 Registers a location callback listener 32
3.5.6 Removes the location listener 32
3.5.7 Stop the location service 32
3.5.8 Creates a circular geofence 33
3.5.9 Registers a listener for geofence creation results 33
3.5.10 Set the action for receiving geofence status broadcasts 33
3.5.11 Removes all geofences 33
3.5.12 Add an APK to a block list 33
3.5.13 Remove an APK from the blocked list 34
3.5.14 Check if an APK is in block 34
3.5.15 Retrieves the list of blocked APKs 34
3.5.16 Disables the geofence feature 34
3.5.17 Enable the location module and initialize with the specified method 34
3.5.18 Enable the location module and initialize with the specified key and method.35
-- Location changed listener - ILocationChangedListener-- 35
3.5.19 Location changed callback 35
-- GeoFence create listener - IGeoFenceCreateListener -- 36
3.5.20 Callback for geofence creation completion 36
➢ 3.6 Network module 36
-- Get network module - getNetworkManager-- 36
3.6.1 Adds an APN configuration 37
3.6.2 Enables an APN configuration 37
3.6.3 Add a Wi-Fi network 37
3.6.4 Connects to a saved Wi-Fi network 38
3.6.5 Disconnect Wi-Fi 38
3.6.6 Forget a Wi-Fi network 38
3.6.7 Get Wi-Fi data usage statistics 38
3.6.8 Get saved Wi-Fi networks 38
3.6.9 Enable or disable Wi-Fi 39
3.6.10 Check Bluetooth support 39
3.6.11 Get Bluetooth state 39
3.6.12 Enable or disable Bluetooth 39

3.6.13 Set the mobile-data switch 40
3.6.14 Get the mobile-data switch state 40
3.6.15 Set data saver mode 40
3.6.16 Get data saver mode state 40
3.6.17 Get application data usage 41
3.6.18 Check SIM-card presence 41
3.6.19 Delete an APN 41
3.6.20 Get the APN list 41
3.6.21 Disable the active APN 42
3.6.22 Set the preferred network type for cellular data 42
3.6.23 Get the preferred network type for cellular data 42
3.6.24 Enable or disable data roaming for the specified SIM card 42
3.6.25 Get the data roaming status for the specified SIM card 43
➢ 3.7 Resource module 43
-- Get resource module - getResourceManager-- 43
3.7.1 Installs or updates an app 43
3.7.2 Uninstalls an app 44
3.7.3 Updates the device system version or MCU firmware version 44
3.7.4 Updates the resource package 44
3.7.5 Install or update an app with a listener 44
-- Upgrade listener - OnAppUpdateListener-- 45
3.7.6 Update the resource package with listener 45
3.7.7 Update the device system version / MCU firmware version with listener 45
3.7.8 Upgrade error/failure 45
3.7.9 Upgrade successful 46
-- Upgrade listener - OnUpdateCustomResListener -- 46
3.7.10 Upgrade error/failure 46
3.7.11 Upgrade successful 46
-- Upgrade listener - OnUpdateOTAListener -- 46
3.7.12 Upgrade error/failure 46
3.7.13 Upgrade successful 47
3.7.14 Callback for upgrade progress 47
➢ 3.8 Device Log module 47
-- Get device log module - getLogManager -- 47
3.8.1 Add package names to the blacklist 48
3.8.2 Add a tag filter 48
3.8.3 Add package names to the whitelist 48
3.8.4 Clear the package blacklist 48
3.8.5 Clear all tag filters 49
3.8.6 Clear the package whitelist 49
3.8.7 Commit or disable log filtering 49
3.8.8 Compress logs by date and time 49
3.8.9 Enable or disable background log collection 50
3.8.10 Get the package blacklist 50
3.8.11 Get the device log directory path 50

3.8.12 Get the log level filter 50
3.8.13 Get the tag filter list 51
3.8.14 Get the package filtering mode 51
3.8.15 Get the package whitelist 51
3.8.16 Initialize the log configuration 51
3.8.17 Set the log level filter 52
3.8.18 Set the package filtering mode 52
3.8.19 Check the log collection state 52
3.8.20 Business services obtain compressed logs and report upload results 53
-- Log compression/status callback - OnCompressListener -- 53
3.8.21 Log collection status changed 53
3.8.22 Log compression finished 53
3.8.23 Log ZIP is ready 54
➢ 3.9 Perception Info module 54
-- get Perception Information Module getPerceptionInfoManager-- 54
3.9.1 Get perception data as a file stream 54
3.9.2 Get large battery cycle count list 54
3.9.3 Get large battery design capacity list 55
3.9.4 Get current max capacity of large battery 55
3.9.5 Get battery health percentage list 55
3.9.6 Get battery health status list 55
3.9.7 Get small battery voltage list 55
3.9.8 Get print distance history list 56
● **4. Error Code Definition 56**
4.1 CertificationError 56
4.2 DeviceError 56
4.3 LocationError 56
4.4 NetworkError 58
4.5 ResourceError 58
4.6 LogError 59
4.7 CommonError 59
● **5. Entity Class Definition 59**
5.1 com.kozen.terminalmanager.Const 59
5.2 com.kozen.terminalmanager.location.constant.LocationConstant 59
5.3 LocationConstant.GeoLanguage 60
5.4 LocationConstant.LocationMode 60
5.5 LocationConstant.SignalMode 60
5.6 ApnConfiguration 61
5.7 WakeUpType 61
5.8 PackageFilterMode 61
5.9 NetworkMode 62
5.10 WifiSecurityType 62
5.11 AttributeValue 62
5.12 com.kozen.terminalmanager.location.constant.LocationConstant.LocationType 62
5.13 MacType 62

- 6. Access permission 62

1. Revision History

Version	Release	Modify Record	Adapted SDK version	Author
1.7	2026/11/04	✓ Add support for PCI scheduled reboot with multiple modes. ✓ Add support for App auto-start with a specified package name. ✓ Add support for retrieving the device MAC address across multiple types (Wi-Fi, Bluetooth, and Ethernet).	TerminalManagerService 1.6.x	Yiven Tong.liu Ye.Zhang
1.6	2026/09/18	✓ Add support for retrieving certificate chain information. ✓ Add support for retrieving the Terminal Management System server address. ✓ Add support for retrieving the SIM card ICCID and phone number. ✓ Add support for retrieving the Bluetooth MAC address. ✓ Add support for retrieving specified system device information. ✓ Add APIs to control preferred network. ✓ Add APIs to control data roaming.	TerminalManagerService 1.5.x	Luna, Yiven
1.5	2026/09/01	✓ Add Device Log module. ✓ Add APIs for obtaining and configuring PCI restart time. ✓ Add APIs to get and set application permissions. ✓ Add API to query developer mode status. ✓ Support manual synchronization of NTP time and network time zone, and configuration of automatic synchronization. ✓ Support show/hide desktop app	TerminalManagerService 1.4.x	Luna, Johnny

		icons and query desktop app list. ✓ Add APIs to control and query screen broadcast switch status. ✓ Support custom wallpaper setting for main display. ✓ Add APIs to query EEPROM battery support status, charge cycles, battery health and read all EEPROM battery data. ✓ Add NetworkManager APIs to support Wi-Fi, Bluetooth, mobile data, data saver, application traffic, cellular network mode, SIM card status and APN management.		
1.4	2026/04/22	✓ Adjust wakeupType to add support for double tap to wake and lift to wake	TerminalManagerService 1.3.x	Johnny
1.3	2026/01/30	✓ Add an information collection module. ✓ Add support for returning OTA upgrade progress monitoring callbacks. ✓ Add a main screen brightness timeout duration. ✓ Add a screen-off reminder setting. ✓ Add log retrieval. ✓ Adjust compatibility for retrieving the OS version number. ✓ Adjust the SDK icon	TerminalManagerService 1.2.x	Johnny
1.2	2025/09/18	✓ Add forcePermission API for mandatory permission check ✓ Add silentInstall API for silent installation ✓ Add CSN API ✓ Add TUSN reading ✓ Extend getLocationManager with 2 open interfaces to specify location provider ✓ Add application install API with listener ✓ Add OTA install API with compatibility support ✓ Add resource install API with	TerminalManagerService 1.1.x	Johnny

		listener ✓ Add module permission ✓ Add a security statement for location		
1.1	2025/03/31	✓ Positioning module constants supplement; ✓ Enhance the resource module with extended error codes and implement upgrade listener for the MCU	TerminalManagerService 1.0.x	Yue.Cui Tong.Liu Johnny
1.0	2025/03/19	✓ Version verification and release	TerminalManagerService 1.0.x	Yao.zhang Sunan Johnny
0.2	2025/03/15	✓ Update LocationManager module, add LocationError	TerminalManagerService 1.0.x	Johnny
0.1	2025/03/14	✓ Initial version	TerminalManagerService 1.0.x	Johnny

● 2. Overview

2.1 Introduction

This document describes the interfaces for TMS and AppStore clients, with broader OS operation permissions compared to the Financial SDK. Main functionalities include installing/uninstalling apps on the OS, retrieving system information, location data, consumable status, and performing operations such as shutting down or rebooting the OS. After client integration, the cloud can be used to manage applications and information on terminals in bulk.

This document serves as the KozenTerminalManagerService API Reference.

2.2 Android version and IDE version supported by the SDK

System environment	Platform	Compile environment
Android 6.0 and above	ARM 64, ARM 32	Android Studio, Intellij

2.3 Feature Introduction

2.3.1 Terminal Management Module

- This class handles SDK initialization and provides access to operation classes for each module.
- The operation class object is: TerminalManager.

2.3.2 Certificate Management Module

- This module provides certificate management functionalities.
- The operation class object is: ICertificationManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getCertificationManager()
Kotlin: TerminalManager.certificationManager()

2.3.3 Device Management Module

- This module handles device-related functionalities.
- The operation class object is: IDeviceManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getDeviceManager()
Kotlin: TerminalManager.deviceManager()

2.3.4 Device Information Management Module

- This module provides functionalities for retrieving device information.
- The operation class object is: IDeviceInfoManager.

- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getDeviceInfoManager()
Kotlin: TerminalManager.deviceInfoManager()

2.3.5 Location Management Module

- This module handles location-related functionalities.
- The operation class object is: ILocationManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getLocationManager()
Kotlin: TerminalManager.locationManager()

2.3.6 Network Management Module

- This module handles network-related functionalities.
- The operation class object is: INetworkManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getNetworkManager()
Kotlin: TerminalManager.networkManager()

2.3.7 Resource Management Module

- This module handles resource management functionalities.
- The operation class object is: IResourceManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getResourceManager()
Kotlin: TerminalManager.resourceManager()

2.3.8 Device Log Module

- This module handles Log collection functionalities.
- The operation class object is: ILogManager.
- Example to obtain the operation class:

JAVA: TerminalManager.INSTANCE.getLogManager()
Kotlin: TerminalManager.logManager()

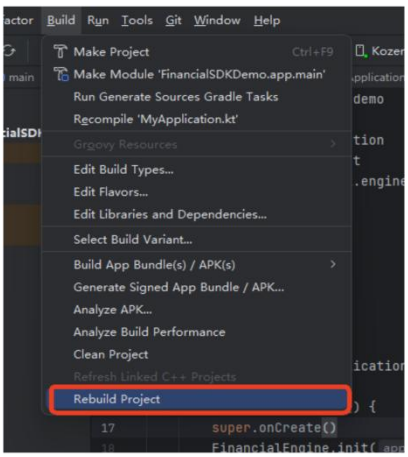
2.4 Importing the TerminalManagerService SDK

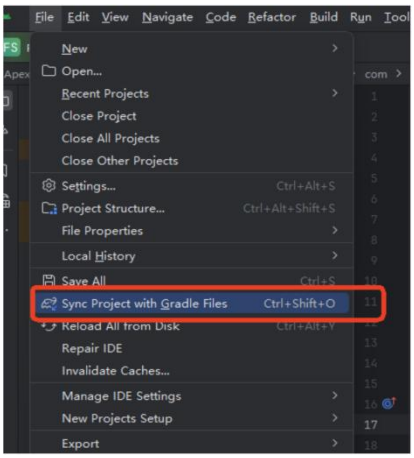
Local Dependency: Place the TerminalManagerLib-release-x.x.x.aar file in the libs directory of your Android Studio project.
Add the following code to the build.gradle file:

gradle 复制

```
dependencies {
    implementation(files("libs/TerminalManagerLib_x.x.x_release.aar"))
}
```

After importing the .aar file, sync and rebuild the project.





2.5 Initializing the TerminalManagerService SDK

Please initialize the TerminalManagerService SDK in your Application. Example:

kotlin 复制

```kotlin
class MyApplication : Application() {
    override fun onCreate() {
        super.onCreate()
        val terminal = TerminalManager
        terminal.init(
            applicationContext
        ) { result, error ->
            Log.d("Application", "[terminal-manager] init result=$result, error=$error")
        }
    }
}
```

If the initialization is successful, the callback will return result=true; if it fails, the callback will return result=false.
After the Terminal Management SDK is successfully initialized, use TerminalManager to obtain the operation objects for each module.

If the Terminal Management service is disconnected during use, the InitCallBack will also be triggered, with result=false. Upon receiving this callback, reinitialize the Terminal Management SDK.

● 3. API Interface Introduction

void init(Context context, InitCallBack callBack)	Terminal manager SDK initialization
ICertificationManager getCertificationManager()	Certification module
IDeviceInfoManager getDeviceInfoManager()	Device information module

IDeviceManager getDeviceManager()	Device module
ILocationManager getLocationManager()	Location module
INetworkManager getNetworkManager()	Network module
IResourceManager getResourceManager()	Resource module
ILogManager getLogManager()	Log module
IPerceptionInfoManager getPerceptionInfoManager()	Perception Info module

➢ 3.1 Terminal manager SDK initialization

3.1.1 Initialize the TerminalManagerService instance

Prototype	void init(Context context, InitCallBack callBack)
Function	Initialize the TerminalManagerService instance
Parameters	Parameters: context - android context callback - initialization callback
Return value	
Notes	

-- TerminalManagerService instance callback - InitCallBack --

void onInitResult(boolean result, String error)	Initialization results

3.1.2 Initialization results

Prototype	void onInitResult(boolean result, String error)
Function	Initialization results
Parameters	Parameters: result - Initialization result (true or false) error - error message
Return value	
Notes	

➢ 3.2 Certification module

-- Get Certification module - getCertificationManager--

int deleteAppSignature(String certData)	Delete the app signature certificate
List<String> getAppSignatureInfo()	Get the app signature certificate information

int updateAppSignature(String certData)	Update the app signature certificate
List <String> getCertificatePublicKeys()	Get the public key from the certificate chain
List <String> getCertificateCustomerNames()	Get the client name from the certificate chain

3.2.1 Updates the application's signature certificate

Prototype	int updateAppSignature(String certData)
Function	Updates the application's signature certificate.
Parameters	Parameters: certData- Certificate Data
Return value	Return: - 0: Success - Others: Failure (specific error codes refer to CertificationError).
Notes	For failure cases, refer to CertificationError for detailed error codes.

3.2.2 Deletes the application's signature certificate

Prototype	int deleteAppSignature(String certData)
Function	Deletes the application's signature certificate.
Parameters	Parameters: certData- Certificate Data to Be Deleted
Return value	Return: - 0: Success - Others: Failure (specific error codes refer to CertificationError).
Notes	For failure cases, refer to CertificationError for detailed error codes.

3.2.3 Get the application's signature certificate information

Prototype	List<String> getAppSignatureInfo()
Function	Get the application's signature certificate information.
Parameters	
Return value	Return: Certificate Details List
Notes	

3.2.4 Get the public key from the certificate chain

Prototype	List <String> getCertificatePublicKeys()
Function	Get the public key from the certificate chain.
Parameters	
Return value	Return: The list index starts at 0, and the last level contains the public key of the certificate currently in use.
Notes	

3.2.5 Get the client name from the certificate chain

Prototype	List <String> getCertificateCustomerNames()
Function	Get the customer name from the certificate chain.
Parameters	
Return value	Return: The list index starts at 0, and the last level is the name of the current customer.
Notes	

➢ 3.3 Device information module

-- Get device information module - getDeviceInfoManager--

String getEmvKernelVersion()	EMV kernel version number
String getHardwareVersion()	Get hardware version number
String[] getImei()	Get IMEI number; if multiple exist, return multiple
String[] getImsi()	Get IMSI number; if multiple exist, return multiple
String getKernelVersion()	Get Linux kernel version number
String getMcuVersion()	Get MCU version number
String getOsVersion()	Get Android version number
String getSdkServiceVersion()	Get SDK service version number
String getSerialNo()	Get device serial number
String getDeviceModel()	Get device model
String getVendorName()	Get vendor name(XC, Kozen)
String getCSN()	Get customer serial number
String getTUSN()	Get TUSN(Only for the China region)
String getTmsAddress()	Get the Terminal Management System server address
String getIccid(int slot)	Get the ICCID of the SIM card in the specified slot
String getPhoneNumber(int slot)	Get the phone number of the SIM card in the specified slot
String getBluetoothMac()	Get the MAC address of the device's Bluetooth adapter
String getAttributeValueByName(AttributeValue key)	Get the specified system device information
String getDeviceMac(MacType type)	Get Device MAC Address

3.3.1 Get the SDK service version number

Prototype	String getSdkServiceVersion()
Function	Get the SDK service version number.
Parameters	
Return value	Return: The SDK service version number.
Notes	

3.3.2 Get Terminal serial number

Prototype	String getSerialNo()
Function	Get Terminal serial number
Parameters	
Return value	Return: The device serial number.
Notes	

3.3.3 Get the IMSI number

Prototype	String[] getImsi()
Function	Retrieves the IMSI number; if multiple exist, returns multiple.
Parameters	
Return value	Return: The IMSI number(s).
Notes	

3.3.4 Get the IMEI number

Prototype	String[] getImei()
Function	Retrieves the IMEI number; if multiple exist, returns multiple.
Parameters	
Return value	Return: The IMEI number(s).
Notes	

3.3.5 Get the equipment supplier name (XC, Kozen)

Prototype	String getVendorName()
Function	Get the equipment supplier name (XC, Kozen)
Parameters	
Return value	Return: Equipment supplier name
Notes	

3.3.6 Get the device model

Prototype	String getDeviceModel()
Function	Get the device model.
Parameters	
Return value	Return: The device model.
Notes	

3.3.7 Get the OS version number

Prototype	String getOsVersion()
Function	Get the OS version number.
Parameters	
Return value	Return: If KozenOS is supported, return the KozenOS version number; if it is not supported, return the Android version number.
Notes	

3.3.8 Get the Linux kernel version number

Prototype	String getKernelVersion()
Function	Get the Linux kernel version number.
Parameters	
Return value	Return: The kernel version number.
Notes	

3.3.9 Get the MCU version number

Prototype	String getMcuVersion()
Function	Retrieves the MCU version number.
Parameters	
Return value	Return: The MCU version number.
Notes	

3.3.10 Get the hardware version number

Prototype	String getHardwareVersion()
Function	Get the hardware version number.
Parameters	
Return value	Return: The hardware version number.
Notes	

3.3.11 Get the EMV kernel version number

Prototype	String getEmvKernelVersion()
Function	Get the EMV kernel version number.
Parameters	
Return value	Return: The EMV kernel version number.
Notes	

3.3.12 Get TUSN (Only for the China region)

Prototype	String getTUSN()
Function	Get TUSN (Only for the China region)
Parameters	
Return Value	Return:TUSN
Notes	Returns the unique terminal serial number.

3.3.13 Get customer serial number

Prototype	String getCSN()
Function	Get customer serial number
Parameters	
Return Value	Return:CSN
Notes	Returns the customer serial number

3.3.14 Get the Terminal Management System server address

Prototype	String getTmsAddress()
Function	Get the Terminal Management System server address
Parameters	
Return Value	Return: Terminal Management System server address
Notes	This address can only be configured through the resource package and cannot be set via the API. Supports third-party Terminal Management Systems.

3.3.15 Get the ICCID of the SIM card in the specified slot

Prototype	String getIccid(int slot)
Function	Get the ICCID of the SIM card in the specified slot
Parameters	Parameters: slot - SIM card slot index, starting from 0.
Return Value	Return: SIM card ICCID; returns an empty string if retrieval fails.
Notes	

3.3.16 Get the phone number of the SIM card in the specified slot

Prototype	String getPhoneNumber(int slot)
Function	Get the phone number of the SIM card in the specified slot
Parameters	Parameters: slot - SIM card slot index, starting from 0.
Return Value	Return: SIM card phone number; returns an empty string if retrieval fails.
Notes	

3.3.17 Get the MAC address of the device's Bluetooth adapter(Deprecated)

Prototype	String getBluetoothMac()
Function	Get the MAC address of the device's Bluetooth adapter
Parameters	
Return Value	Return: Bluetooth MAC address; returns an empty string if retrieval fails.
Notes	

3.3.18 Get the specified system device information

Prototype	String getAttributeValueByName(AttributeValue key)
Function	Get the specified device attribute
Parameters	Parameters: key - An enumeration of system property values. See AttributeValue for details.
Return Value	Return: Specified system device information; returns null if unavailable.
Notes	

3.3.19 Get Device MAC Address

Prototype	String getDeviceMac(MacType type)
Function	Get Device MAC Address
Parameters	type - The type of MAC address: MacType.WIFI - Get the Wi-Fi MAC address. MacType.BLUETOOTH - Get the Bluetooth MAC address. MacType.ETHERNET - Get the Ethernet MAC address.
Return Value	Return: The retrieved MAC address.
Notes	

➢ 3.4 Device module

-- Get device module - getDeviceManager--

int cancelPCIReboot()	Cancel scheduled reboot
long getSystemTime()	Return the timestamp of the current system time
String getTimeZone()	Get the current time zone
void reboot()	Reboot the device
int setPCIReboot(long time)	Schedule a reboot
int setSystemTime(long time)	Set the system time
int setTimeZone(String timeZone)	Set the time zone
void shutdown()	Shut down the device
void forcePermission(boolean isOpen)	Forced permission request
void setSilentInstall(boolean isOpen)	Silent app installation
int setScreenTimeOut(int timeOut)	Set screen timeout duration
int sleep(boolean isConfirm)	Turn off (sleep) the screen
int wakeUp(WakeUpType type, boolean open)	Enable or disable various device wake-up
long getPCIRebootCountdownTime()	Get PCI reboot countdown time.
long getPCIReboot()	Get the configured PCI reboot time.
List<AppPermissionInfo> getAppPermissions(String packageName)	Get application permissions.
int setAppPermission(String packageName, String permissionName, boolean enable)	Set application permission.
int isDeveloperMode()	Check whether developer mode is enabled.
int syncSystemTimeFromNtp()	Synchronize system time from NTP.
int syncTimeZoneFromNetwork()	Synchronize time zone from network.
int setAutoSystemTimeSyncEnabled(boolean enabled)	Sets automatic system time synchronization.
int setAutoTimeZoneSyncEnabled(boolean enabled)	Sets automatic system time-zone synchronization.
int hideAppFromLauncher(String packageName, boolean isHide)	Show or hide a launcher application icon.
Map<String, Boolean> getAppListFromLauncher()	Gets the launcher application list.
void setScreenReader(boolean enabled)	Enables or disables the screen reader.
boolean isScreenReaderEnabled()	Get screen-reader switch state.
int setPrimaryScreenWallpaper(String path)	Set the primary-screen wallpaper.
int isSupportEepromBattery()	Check whether an EEPROM battery is supported.
int getBatteryCycleCount()	Gets the battery charge cycle count.
int getBatteryHealth()	Gets battery health as a percentage.
Map<String, String> readEepromBatteryField()	Reads EEPROM battery data fields.
int setPCIReboot(int mode,long rebootTime)	Schedule a timed reboot with mode

	configuration
int setAppAutoStart(String packageName)	Set App Auto-Start

3.4.1 Sets the system time

Prototype	int setSystemTime(long time)
Function	Sets the system time.
Parameters	time - Timestamp(ms).
Return value	Return: 0: Success, Others: Failure (refer to DeviceError).
Notes	

3.4.2 Returns the current system time as a timestamp

Prototype	long getSystemTime()
Function	Returns the current system time as a timestamp(ms).
Parameters	
Return value	Return: The timestamp.
Notes	

3.4.3 Sets the time zone

Prototype	int setTimeZone(String timeZone)
Function	Sets the time zone.
Parameters	timeZone - Time zone ID, supports two formats: 1. Region/City (e.g., Europe/Moscow) 2. GMT (e.g., GMT+9).
Return value	Return: 0: Success, Others: Failure (refer to DeviceError).
Notes	

3.4.4 Get the current time zone

Prototype	String getTimeZone()
Function	Retrieves the current time zone.
Parameters	
Return value	Return: The time zone.
Notes	

3.4.5 Reboots the device

Prototype	void reboot()

Function	Reboots the device
Parameters	
Return value	
Notes	

3.4.6 Shuts down the device

Prototype	void shutdown()
Function	Shuts down the device.
Parameters	
Return value	
Notes	

3.4.7 Schedules a timed reboot(Deprecated)

Prototype	int setPCIReboot(long time)
Function	Schedules a timed reboot.
Parameters	time - Device will reboot after running for this duration (e.g., 1000 * 60 * 60 for 1 hour).
Return value	Return: 0: Success, Others: Failure (refer to DeviceError).
Notes	1. The policy will run automatically unless it is canceled by calling cancelPCIReboot 2. A reboot prompt will be shown before rebooting. 3. It is recommended not to exceed 24 hours. 4. If the set time is less than the device's uptime, it will reboot immediately.

3.4.8 Cancels the scheduled reboot

Prototype	int cancelPCIReboot()
Function	Cancels the scheduled reboot.
Parameters	
Return value	Return: 0: Success, Others: Failure (refer to DeviceError).
Notes	

3.4.9 Enable or disable silent installation

Prototype	void setSilentInstall(boolean isOpen)
Function	Enable or disable silent installation
Parameters	Parameters: isOpen - true: Do not show app installation prompt (silent install) false: Show installation prompt
Return Value	
Notes	When enabled, applications can be installed silently without prompting the user.

3.4.10 Enforce runtime permission checks

Prototype	void forcePermission(boolean isOpen)
Function	Enforce runtime permission checks
Parameters	Parameters: isOpen - true: The app must request permission to access a module false: No permission enforcement
Return Value	
Notes	When enabled, modules cannot be accessed unless the app has explicitly been granted the required permissions.

3.4.11 Set the screen timeout duration

Prototype	int setScreenTimeOut(int timeOut)
Function	Set the screen timeout duration in seconds
Parameters	Parameters: timeOut - 0: Always on 1800: Maximum value (30 minutes)
Return Value	Return: 0:Success -1:Failure,Others , See DeviceError for details
Notes	This method sets how long the screen stays on before turning off due to inactivity.

3.4.12 Turn off the screen (sleep mode)

Prototype	int sleep(boolean isConfirm)
Function	Turn off the screen (sleep mode)
Parameters	Parameters: isConfirm - Whether to show a confirmation dialog before sleeping true - Show confirmation dialog false - Enter sleep directly
Return Value	Return: 0:Success -1:Failure
Notes	After calling this method, the screen will enter sleep mode in 5 seconds. The user can tap the screen to wake it up.This setting only takes effect for the next sleep.

3.4.13 Enable or disable various device wake-up methods

Prototype	int wakeUp(WakeUpType type, boolean open)
Function	Enable or disable various device wake-up methods
Parameters	Parameters: type - Wake-up method enum value (see WakeUpType)

	open- true: Enable the wake-up method false: Disable the wake-up method
Return Value	Return: 0:Success <0:Failure(Refer to DeviceError)
Notes	This method configures whether a specific device wake-up method is enabled or disabled.

3.4.14 Get PCI reboot countdown time

Prototype	long getPCIRebootCountdownTime()
Function	Get PCI reboot countdown time.
Parameters	
Return Value	Return: Remaining time in milliseconds.
Notes	

3.4.15 Get the configured PCI reboot time

Prototype	long getPCIReboot()
Function	Gets the configured PCI reboot time.
Parameters	
Return Value	Return: Configured PCI restart time in milliseconds.
Notes	

3.4.16 Get application permissions

Prototype	List<AppPermissionInfo> getAppPermissions(String packageName)
Function	Get application permissions.
Parameters	Parameters: packageName - Application package name.
Return Value	Return: A list containing all permissions used by the application. See AppPermissionInfo.
Notes	Supports Android native permissions and OS-defined permissions.

3.4.17 Set application permission

Prototype	int setAppPermission(String packageName, String permissionName, boolean enable)
Function	Set application permission.
Parameters	Parameters: packageName - Application package name. permissionName - Android native or OS-defined permission name.

	enable - true: grant/enable; false: revoke/disable.
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	Supports Android native permissions and Kozen-defined permissions.

3.4.18 Check whether developer mode is enabled

Prototype	int isDeveloperMode()
Function	Check whether developer mode is enabled.
Parameters	
Return Value	Return: 0: Production mode; 1: Developer mode.
Notes	

3.4.19 Synchronize system time from NTP

Prototype	int syncSystemTimeFromNtp()
Function	Synchronize system time from NTP.
Parameters	
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	Immediately synchronizes system time from an NTP server.

3.4.20 Synchronize time zone from network

Prototype	int syncTimeZoneFromNetwork()
Function	Synchronize time zone from network.
Parameters	
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	Immediately synchronizes time zone from network

3.4.21 Sets automatic system time synchronization

Prototype	int setAutoSystemTimeSyncEnabled(boolean enabled)
Function	Sets automatic system time synchronization.
Parameters	Parameters: enabled - true: enable; false: disable.
Return Value	Return: 0: Success;

	<0: Failure (refer to DeviceError).
Notes	

3.4.22 Sets automatic system time-zone synchronization

Prototype	int setAutoTimeZoneSyncEnabled(boolean enabled)
Function	Sets automatic system time-zone synchronization.
Parameters	Parameters enabled - true: enable; false: disable.
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	

3.4.23 Show or hide a launcher application icon

Prototype	int hideAppFromLauncher(String packageName, boolean isHide)
Function	Show or hide a launcher application icon.
Parameters	Parameters: packageName - Application package name. isHide - true: hide; false: show.
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	

3.4.24 Get the launcher application list

Prototype	Map<String, Boolean> getAppListFromLauncher()
Function	Gets the launcher application list.
Parameters	
Return Value	Return String - APP package name Boolean - isHide (true: hide; false: show)
Notes	Returns list of Launcher visible apps.

3.4.25 Enable or disable the screen reader

Prototype	void setScreenReader(boolean enabled)
Function	Enables or disables the screen reader.
Parameters	Parameters: enabled - true: enable;

	false: disable.
Return Value	
Notes	

3.4.26 Get screen-reader switch state

Prototype	boolean isScreenReaderEnabled()
Function	Get screen-reader switch state.
Parameters	
Return Value	Return: true: enabled; false: disabled.
Notes	

3.4.27 Set the primary-screen wallpaper

Prototype	int setPrimaryScreenWallpaper(String path)
Function	Set the primary-screen wallpaper.
Parameters	Parameters: path - Wallpaper file path.
Return Value	Return: 0: Success; <0: Failure (refer to DeviceError).
Notes	Replaces the built-in wallpaper

3.4.28 Check whether an EEPROM battery is supported

Prototype	int isSupportEepromBattery()
Function	Checks whether an EEPROM battery is supported.
Parameters	
Return Value	Return: 1: Supported; 0: Not supported.
Notes	Include both the device and battery

3.4.29 Get battery charge cycle count

Prototype	int getBatteryCycleCount()
Function	Gets the battery charge cycle count.
Parameters	
Return Value	Return int: Charge cycle count.
Notes	

3.4.30 Get battery health percentage

Prototype	int getBatteryHealth()
Function	Gets battery health as a percentage.
Parameters	
Return Value	Return int: from 0 to 100(Battery health)
Notes	

3.4.31 Read EEPROM battery fields

Prototype	Map<String, String> readEepromBatteryField()
Function	Reads EEPROM battery data fields.
Parameters	
Return Value	Return Map<data key, data value> key - EEPROM battery key information value - EEPROM battery value
Notes	

3.4.32 Schedule a timed reboot with mode configuration

Prototype	int setPCIReboot(int mode, long rebootTime)
Function	Schedule a timed reboot with mode configuration.
Parameters	mode - The reboot mode. Default: 0. 0: When the device uptime reaches the specified duration, the reboot is triggered. 1: When the device reaches the specified system time (e.g., 8:00 AM daily), the reboot mechanism is triggered, but the first reboot prompt is displayed at a random time within the two-hour window starting at the configured time (i.e., at some time between 8:00 AM and 10:00 AM). rebootTime - The time of reboot, expressed as a millisecond value in the range of 0 - 24 hours. Default: 23h55m, which triggers the dialog box.
Return value	Return: 0: Success. <0: Failure (refer to DeviceError).
Notes	1. The policy will run automatically unless it is canceled by calling cancelPCIReboot. 2. A dialog box will be displayed before the reboot. If no action is taken on the dialog box, or if the user clicks "Confirm," the device will reboot after 5 minutes. 3. The dialog box can be postponed up to 5 times, with a new prompt triggered at a 1-hour interval for each postponement. On the 6th occurrence, the "Postpone" option will be unavailable, and the device will reboot after 5 minutes. 4. If mode 0 is set and the configured time is less than the device's current uptime, the reboot dialog box will be displayed immediately.

3.4.33 Set App Auto-Start

Prototype	int setAppAutoStart(String packageName)
Function	Set App Auto-Start.
Parameters	packageName - The package name of the app to auto-start. Takes effect on the next boot. Passing an empty string will clear the app auto-start setting.
Return value	Return: 0: Success. <0: Failure (refer to DeviceError).
Notes	

➢ 3.5 Location module

Declaration on Location Interface

Our location interface is designed solely to provide location-related services to applications.

- It does not actively collect device information such as SN, IMEI, or hardware identifiers.
- It does not transmit any device information or location data to external servers without explicit application request and authorization.
- All data access is limited to the scope necessary for location service functionality.

This ensures that the interface complies with user privacy protection and security requirements.

-- Get Location module - getLocationManager--

int addToBlockOpenAppList(String pkgName)	Add an APK to a block list
List<String> getBlockOpenAppList()	Get the list of blocked APKs
boolean isInBlockOpenAppList(String pkgName)	Check if an APK is in block
void onDestroy()	Destroy location and geofencing resources
int open()	Start the location module and initialize related resources.
int open(String key)	Start the location module and initialize related resources by key.
int open(LocationConstant.LocationType locationType)	Enable the positioning module, specify the positioning method, and initialize the related resources.
int open(String key, LocationConstant.LocationType locationType)	Enable the positioning module, specify the positioning key and method, and initialize the related resources.
int registerGeoFenceCreateListener(IGeoFenceCreateListener listener)	Register a listener for geofence creation results
int registerLocationListener(ILocationChangedListener listener)	Register a location callback listener

int removeAllGeoFence()	Remove all geofences
int removeFromBlockOpenAppList(String pkgName)	Remove an APK from the blocked list
int setGeoFenceResultAction(String action, String pkgName)	Set the action to receive geofence status broadcasts.
int setLocationOption(LocationClientOption locationClientOption)	Set location parameters
int startOnceLocation()	Start a single location request
int stopLocation()	Stop location services
int unRegisterLocationListener()	Remove the location listener
int addGeoFence(double longitude, double latitude, float radius, String customId)	Create a circular geofence

3.5.1 Enables the location module and initializes related resources

Prototype	int open()
Function	Enables the location module and initializes related resources.
Parameters	
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	

3.5.2 Enables the location module and initializes related resources by key

Prototype	int open(String key)
Function	Enables the location module and initializes related resources.
Parameters	key - The API key for location. By default, the system-signed key is used.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	If the system-signed key fails (location fails with error code ERROR_CODE_FAILURE_AUTH), Please contact Kozen FAE to generate a special API key. Then call onDestroy first, and use this API to reinitialize the location module by passing the API key.

3.5.3 Set location parameters

Prototype	int setLocationOption(LocationClientOption locationClientOption)
Function	Set location parameters. Must be used after open is successful.
Parameters	locationClientOption - Location parameters.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Default configuration parameters:

	1. High-accuracy mode; 2. Location request interval is 2000ms; 3. Network location timeout is 30s; 4. No reverse geocoding address information; 5. Reverse geocoding language is selected based on the region; 6. Signal selection is network plus GPS

3.5.4 Start a single location request

Prototype	int startOnceLocation()
Function	Start a single location request. Must be used after open is successful.
Parameters	
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	The location request will automatically stop after completion. You can also call stopLocation to terminate the location request during the process

3.5.5 Registers a location callback listener

Prototype	int registerLocationListener(ILocationChangedListener listener)
Function	Registers a location callback listener
Parameters	listener - Callback interface for location updates.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful. Only one listener can be registered globally; new registrations will overwrite previous ones. Call unRegisterLocationListener to remove the listener when it is no longer needed

3.5.6 Removes the location listener

Prototype	int unRegisterLocationListener()
Function	Removes the location listener.
Parameters	
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful.

3.5.7 Stop the location service

Prototype	int stopLocation()
Function	Stop the location service.
Parameters	
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful.

3.5.8 Creates a circular geofence

Prototype	int addGeoFence(double longitude, double latitude, float radius, String customId)
Function	Creates a circular geofence.
Parameters	longitude - Longitude of the geofence center; latitude - Latitude of the geofence center; radius - Radius of the geofence (minimum 100 meters); customId - Custom geofence ID, must be unique.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful.

3.5.9 Registers a listener for geofence creation results

Prototype	int registerGeoFenceCreateListener(IGeoFenceCreateListener listener)
Function	Registers a listener for geofence creation results.
Parameters	listener - Callback interface for geofence creation.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful.

3.5.10 Set the action for receiving geofence status broadcasts

Prototype	int setGeoFenceResultAction(String action, String pkgName)
Function	Set the action for receiving geofence status broadcasts.
Parameters	action - Broadcast action; pkgName - Package name for receiving the broadcast.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Multiple settings will use the last one. Must be used after open is successful.

3.5.11 Removes all geofences

Prototype	int removeAllGeoFence()
Function	Removes all geofences.
Parameters	
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	Must be used after open is successful.

3.5.12 Add an APK to a block list

Prototype	int addToBlockOpenAppList(String pkgName)

Function	Add an APK to a block list
Parameters	pkgName - Package name of the APK.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	

3.5.13 Remove an APK from the blocked list

Prototype	int removeFromBlockOpenAppList(String pkgName)
Function	Remove an APK from the blocked list
Parameters	pkgName - Package name of the APK.
Return value	0: Operation successful; Others: Operation failed. Refer to LocationError for specific error codes.
Notes	

3.5.14 Check if an APK is in block

Prototype	boolean isInBlockOpenAppList(String pkgName)
Function	Check if an APK is in block
Parameters	pkgName - Package name of the APK.
Return value	true: APK is disabled; false: APK is not disabled.
Notes	

3.5.15 Retrieves the list of blocked APKs

Prototype	List<String> getBlockOpenAppList()
Function	Retrieves the list of blocked APKs.
Parameters	
Return value	Returns the list of disabled APK package names.
Notes	

3.5.16 Disables the geofence feature

Prototype	void onDestroy()
Function	Destroys the location and geofence resources.
Parameters	
Return value	
Notes	Call this method when the location module is no longer needed.

3.5.17 Enable the location module and initialize with the specified method

Prototype	int open(LocationConstant.LocationType locationType)
Function	Enable the location module and initialize with the specified method
Parameters	Parameters: locationType - The desired location method

	(from LocationConstant.LocationType)
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see LocationError for detailed error codes)
Notes	Initializes the location module and allocates necessary resources according to the specified method.

3.5.18 Enable the location module and initialize with the specified key and method

Prototype	int open(String key, LocationConstant.LocationType locationType)
Function	Enable the location module and initialize with the specified key and method
Parameters	Parameters: key - API key for location services. By default, the system-signed key is used. If the system key fails (error code ERROR_CODE_FAILURE_AUTH), contact support for a special API key. After obtaining the key, call onDestroy() first, then re-initialize using this API with the new key. locationType - The desired location method (from LocationConstant.LocationType)
Return Value	Return: 0 - Operation succeeded Others - Operation failed (see LocationError for detailed error codes)
Notes	This method starts the location module and allocates necessary resources for the specified location method.

-- Location changed listener - ILocationChangedListener--

void onLocationChanged(MapLocation mapLocation, int errorCode, String errorDetail)	Location changed callback

3.5.19 Location changed callback

Prototype	void onLocationChanged(MapLocation mapLocation, int errorCode, String errorDetail)
Function	Location changed callback
Parameters	mapLocation - Location result information; errorCode - Error code; errorDetail - Detailed error information.
Return value	
Notes	This method is called when the location is completed.

-- GeoFence create listener - IGeoFenceCreateListener --

void onGeoFenceCreateFinished(List<GeoFence> geoFenceList, int errorCode, String customId)	Callback for geofence creation completion

3.5.20 Callback for geofence creation completion

Prototype	void onGeoFenceCreateFinished(List<GeoFence> geoFenceList, int errorCode, String customId)
Function	Callback for geofence creation completion.
Parameters	geoFenceList - List of created geofences (only available if creation is successful, otherwise empty); errorCode - Error code; customId - Custom business ID associated with this operation.
Return value	
Notes	

➢ 3.6 Network module

-- Get network module - getNetworkManager--

int addApn(ApnConfiguration config)	Add an APN.
int enableApn(String name)	Enable an APN.
void addWifiNetwork(String ssid, String password, WifiSecurityType security)	Add a Wi-Fi network.
void connectWifi(String ssid)	Connects to a saved Wi-Fi network.
void disconnectWifi()	Disconnect Wi-Fi.
void forgetWifi(String ssid)	Forget a Wi-Fi network.
List<DataUsageStats> getWifiDataUsageStats(Date fromTimestamp, Date toTimestamp)	Get Wi-Fi data usage statistics.
WifiNetwork[] getWifiNetworks()	Get saved Wi-Fi networks.
void toggleWifi(boolean enabled)	Enable or disable Wi-Fi.
boolean hasBluetooth()	Check Bluetooth support.
boolean isBluetoothActive()	Get Bluetooth state.
void toggleBluetooth(boolean enabled)	Enable or disable Bluetooth.
int setMobileDataEnabled(boolean enabled)	Set the mobile-data switch.
int isMobileDataEnabled()	Get the mobile-data switch state.

void setDataSaverAllowed(boolean allowed)	Set data saver mode.
boolean isDataSaverAllowed()	Get data saver mode state.
AppDataUsageInfo getAppDataUsage(String packageName, long startTime, long endTime)	Get application data usage.
int isSimCardPresent(int slotIndex)	Check SIM-card presence.
int deleteApn(String name)	Delete an APN.
List<ApnConfiguration> getApnList()	Get the APN list.
int disableApn()	Disable the active APN.
int setCellularNetworkMode(int slotIndex, int mode)	Set the preferred cellular network type.
int getCellularNetworkMode(int slotIndex)	Get the preferred cellular network type.
int setDataRoamingEnabled(int slot, boolean enabled)	Enable or disable data roaming for the specified SIM card.
int getDataRoamingEnabled(int slot)	Get the data roaming status for the specified SIM card.

3.6.1 Adds an APN configuration

Prototype	int addApn(ApnConfiguration config)
Function	Adds an APN configuration.
Parameters	config - The APN configuration to add.
Return value	Return: 0: Success; Others: Failure (refer to NetworkError).
Notes	

3.6.2 Enables an APN configuration

Prototype	int enableApn(String name)
Function	Enables an APN configuration.
Parameters	name - The name of the APN. Pass null to disable the currently used APN.
Return value	Return: 0: Success; Others: Failure (refer to NetworkError).
Notes	

3.6.3 Add a Wi-Fi network

Prototype	void addWifiNetwork(String ssid, String password, WifiSecurityType security)
Function	Adds a Wi-Fi network.
Parameters	Parameters: ssid - Wi-Fi network name.

	password - Password. security - Encryption type; see WifiSecurityType.
Return Value	
Notes	The Wi-Fi toggle state can affect remote configuration.

3.6.4 Connects to a saved Wi-Fi network

Prototype	void connectWifi(String ssid)
Function	Connects to a saved Wi-Fi network.
Parameters	Parameters: ssid - Wi-Fi network name.
Return Value	
Notes	

3.6.5 Disconnect Wi-Fi

Prototype	void disconnectWifi()
Function	Disconnect Wi-Fi.
Parameters	
Return Value	
Notes	

3.6.6 Forget a Wi-Fi network

Prototype	void forgetWifi(String ssid)
Function	Forget a Wi-Fi network.
Parameters	Parameters: ssid - Wi-Fi network name.
Return Value	
Notes	

3.6.7 Get Wi-Fi data usage statistics

Prototype	List<DataUsageStats> getWifiDataUsageStats(Date fromTimestamp, Date toTimestamp)
Function	Get Wi-Fi data usage statistics.
Parameters	Parameters: fromTimestamp - Start time. toTimestamp - End time.
Return Value	Return: Wi-Fi data usage details.
Notes	

3.6.8 Get saved Wi-Fi networks

Prototype	WifiNetwork[] getWifiNetworks()

Function	Get saved Wi-Fi networks.
Parameters	
Return Value	Return: Saved Wi-Fi networks.
Notes	

3.6.9 Enable or disable Wi-Fi

Prototype	void toggleWifi(boolean enabled)
Function	Enable or disable Wi-Fi.
Parameters	Parameters: enabled - true: enable Wi-Fi; false: disable Wi-Fi.
Return Value	
Notes	This toggle can affect remote Wi-Fi configuration.

3.6.10 Check Bluetooth support

Prototype	boolean hasBluetooth()
Function	Check Bluetooth support.
Parameters	
Return Value	Return: true: supported; false: not supported.
Notes	

3.6.11 Get Bluetooth state

Prototype	boolean isBluetoothActive()
Function	Get Bluetooth state.
Parameters	
Return Value	Return: true: enabled; false: disabled.
Notes	

3.6.12 Enable or disable Bluetooth

Prototype	void toggleBluetooth(boolean enabled)
Function	Enable or disable Bluetooth.
Parameters	Parameters: enabled - true: enable; false: disable.
Return Value	

Notes	

3.6.13 Set the mobile-data switch

Prototype	int setMobileDataEnabled(boolean enabled)
Function	Set the mobile-data switch.
Parameters	Parameters: enabled - true: enable; false: disable.
Return Value	Return: 0: Success; <0: Failure(Refer to NetworkError).
Notes	

3.6.14 Get the mobile-data switch state

Prototype	int isMobileDataEnabled()
Function	Get the mobile-data switch state.
Parameters	
Return Value	Return: 1: enable; 0: disable. <0: error, Refer to NetworkError.
Notes	

3.6.15 Set data saver mode

Prototype	void setDataSaverAllowed(boolean allowed)
Function	Set data saver mode.
Parameters	Parameters: allowed - true: enable data saver; false: disable.
Return Value	
Notes	

3.6.16 Get data saver mode state

Prototype	boolean isDataSaverAllowed()
Function	Get data saver mode state.
Parameters	
Return Value	Return: true: enabled; false: disabled.
Notes	

3.6.17 Get application data usage

Prototype	AppDataUsageInfo getAppDataUsage(String packageName, long startTime, long endTime)
Function	Get application data usage.
Parameters	Parameters: packageName - Application package name. startTime - Start timestamp(ms). endTime - End timestamp(ms).
Return Value	Return: Execution result containing application data usage.
Notes	

3.6.18 Check SIM-card presence

Prototype	int isSimCardPresent(int slotIndex)
Function	Check SIM-card presence.
Parameters	Parameters: slotIndex - SIM Card slot
Return Value	Return: 1: Present; 0: Not present <0: Failure(Refer to NetworkError).
Notes	

3.6.19 Delete an APN

Prototype	int deleteApn(String name)
Function	Delete an APN.
Parameters	Parameters: name - APN entry name.
Return Value	Return: 0: Success; <0: Failure(Refer to NetworkError).
Notes	

3.6.20 Get the APN list

Prototype	List<ApnConfiguration> getApnList()
Function	Get the APN list.
Parameters	
Return Value	Return: APN configuration list.
Notes	

3.6.21 Disable the active APN

Prototype	int disableApn()
Function	Disable the active APN.
Parameters	
Return Value	Return: 0: Success; <0: Failure(Refer to NetworkError).
Notes	

3.6.22 Set the preferred network type for cellular data

Prototype	int setCellularNetworkMode(int slotIndex, int mode)
Function	Set the preferred network type for cellular data.
Parameters	Parameters: slotIndex - SIM card slot index mode - Preferred network mode(Refer to NetworkMode).
Return Value	Return: 0: Success; <0: Failure(Refer to NetworkError).
Notes	

3.6.23 Get the preferred network type for cellular data

Prototype	int getCellularNetworkMode(int slotIndex)
Function	Get the preferred network type for cellular data.
Parameters	Parameters: slotIndex - SIM card slot index.
Return Value	Return: NetworkMode.AUTO: Auto choose NetworkMode.MODE_2G_ONLY: Only 2G mode NetworkMode.MODE_3G_ONLY: Only 3G mode NetworkMode.MODE_4G_ONLY: Only 4G mode NetworkMode.MODE_4G_PREFERRED: preferred 4G mode NetworkMode.MODE_5G_PREFERRED: preferred 5G mode <0: Failure(Refer to NetworkError).
Notes	

3.6.24 Enable or disable data roaming for the specified SIM card

Prototype	int setDataRoamingEnabled(int slot, boolean enabled)
Function	Enable or disable data roaming for the specified SIM card.
Parameters	Parameters: slot - SIM card slot index, starting from 0.

	enabled - true: Enabled; false: Disabled.
Return Value	Return: 0: Success; <0: Failure(Refer to NetworkError).
Notes	

3.6.25 Get the data roaming status for the specified SIM card

Prototype	int getDataRoamingEnabled(int slot)
Function	Get the data roaming status for the specified SIM card.
Parameters	Parameters: slot - SIM card slot index, starting from 0.
Return Value	Return: 1: Enabled; 0: Disabled; <0: Failure(Refer to NetworkError).
Notes	

➢ 3.7 Resource module

-- Get resource module - getResourceManager--

int installOrUpdate(String path)	Install or update an app
int unInstall(String pkgName)	Uninstall an app
int updateCustomRes(String path)	Update the resource package
int updateOTA(String path)	Update the device system version / MCU firmware version
int installOrUpdateWithListener(String path, OnAppUpdateListener listener)	Install or update an app with listener
int updateCustomResWithListener(String path, OnUpdateCustomResListener listener)	Update the resource package with listener
int updateOTAWithListener(String path, OnUpdateOTAListener listener)	Update the device system version / MCU firmware version with listener

3.7.1 Installs or updates an app

Prototype	int installOrUpdate(String path)
Function	Installs or updates an app.
Parameters	path - The path of the app installation package.
Return value	Return: 0: Success

	Others: Failure (refer to ResourceError).
Notes	

3.7.2 Uninstalls an app

Prototype	int unInstall(String pkgName)
Function	Uninstalls an app.
Parameters	pkgName - The package name of the app to uninstall.
Return value	Return: 0: Success, Others: Failure (refer to ResourceError).
Notes	

3.7.3 Updates the device system version or MCU firmware version

Prototype	int updateOTA(String path)
Function	Updates the device system version or MCU firmware version.
Parameters	path - The path of the update package (supports /storage/emulated/0/ or /sdcard/). - .zip: Recognized as a system version package. - .bin: Recognized as an MCU firmware package.
Return value	Return: 0: Package verification successful, starting upgrade. Others: Failure (refer to ResourceError).
Notes	

3.7.4 Updates the resource package

Prototype	int updateCustomRes(String path)
Function	Updates the resource package.
Parameters	path - The path of the resource package.
Return value	Return: 0: Package verification successful, starting resource update. Others: Failure (refer to ResourceError).
Notes	

3.7.5 Install or update an app with a listener

Prototype	int installOrUpdateWithListener(String path, OnAppUpdateListener listener)
Function	Install or update an app with listener
Parameters	Parameters: path - Path to the app package (either a file URI string or absolute accessible file path) listener - Callback for app installation result (see OnAppUpdateListener)
Return Value	Return: 0 - API call successful

	Others - API call failed (see CommonError for details)
Notes	

-- Upgrade listener - OnAppUpdateListener--

void onError(String msg, int code)	Upgrade error
void onSuccess()	App upgrade successful

3.7.6 Update the resource package with listener

Prototype	int updateCustomResWithListener(String path, OnUpdateCustomResListener listener)
Function	Update the resource package with listener
Parameters	Parameters: path - Path to the resource package (either a file URI string or absolute accessible file path) listener - Callback for update result (see OnUpdateCustomResListener)
Return Value	Return: 0 - API call successful Others - API call failed (see CommonError for details)
Notes	

3.7.7 Update the device system version / MCU firmware version with listener

Prototype	int updateOTAWithListener(String path, OnUpdateOTAListener listener)
Function	Update the device system version / MCU firmware version with listener
Parameters	Parameters: path - Path to the OTA package (either a file URI string or absolute accessible file path) listener - Callback for update result (see OnUpdateOTAListener)
Return Value	Return: 0 - API call successful Others - API call failed (see CommonError for details)
Notes	

3.7.8 Upgrade error/failure

Prototype	void onError(String msg, int code)
Function	Upgrade failed
Parameters	msg - Error message code - Error code (refer to ResourceError)
Return value	
Notes	

3.7.9 Upgrade successful

Prototype	void onSuccess()
Function	Upgrade successful
Parameters	
Return value	
Notes	

-- Upgrade listener - OnUpdateCustomResListener --

void onError(String msg, int code)	Upgrade error/failure
void onSuccess(String resultPath)	Upgrade successful

3.7.10 Upgrade error/failure

Prototype	void onError(String msg, int code)
Function	Upgrade failed
Parameters	msg - Error message code - Error code (refer to ResourceError)
Return value	
Notes	

3.7.11 Upgrade successful

Prototype	void onSuccess(String resultPath)
Function	Upgrade successful
Parameters	
Return value	
Notes	

-- Upgrade listener - OnUpdateOTAListener --

void onError(String msg, int code)	Upgrade error/failure
void onSuccess()	Upgrade successful
void onUpdateInProgress(int percent)	Callback for upgrade progress

3.7.12 Upgrade error/failure

Prototype	void onError(String msg, int code)
Function	Upgrade failed
Parameters	msg - Error message code - Error code (refer to ResourceError)
Return value	
Notes	

3.7.13 Upgrade successful

Prototype	void onSuccess()
Function	Upgrade successful
Parameters	
Return value	
Notes	

3.7.14 Callback for upgrade progress

Prototype	void onUpdateInProgress(int percent)
Function	Firmware or application update progress callback
Parameters	Parameters: percent - Update progress percentage (0 - 100)
Return Value	
Notes	This method is called to report the current progress of an ongoing update.

➢ 3.8 Device Log module

-- Get device log module - getLogManager --

int addBlackList(String packageNames)	Add package names to the blacklist.
int addFilterTag(String tag)	Add a tag filter.
int addWhiteList(String packageNames)	Add package names to the whitelist.
int clearBlackList()	Clear the package blacklist.
int clearFilterTag()	Clear all tag filters.
int clearWhiteList()	Clear the package whitelist.
int confirmLogFilter(boolean logFilter)	Commit or disable log filtering.
int createZipByDateTime(long startDateTime, long endDateTime, OnCompressListener listener)	Compress logs by date and time.
int enableLog(boolean enable)	Enable or disable background log collection.
ArrayList<String> getBlackList()	Get the package blacklist.
String getDeviceLogsPath()	Get the device log directory path.
String getFilterLevel()	Get the log level filter.
ArrayList<String> getFilterTag()	Get the tag filter list.
PackageFilterMode getPackageFilterMode()	Get the package filtering mode.
ArrayList<String> getWhiteList()	Get the package whitelist.
int initLogConfig(int bufferSize, int totalSize, int fileSize, String logDir)	Initialize the log configuration.

int setFilterLevel(String level)	Set the log level filter.
int submitPackageFilter(PackageFilterMode mode)	Set the package filtering mode .
boolean isLogEnabled()	Check the log collection state.
int reportUploadResult(boolean result)	Business services obtain compressed logs and report upload results.

3.8.1 Add package names to the blacklist

Prototype	int addBlackList(String packageNames)
Function	Add package names to the blacklist.
Parameters	Parameters: packageNames - Package name(s).
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.2 Add a tag filter

Prototype	int addFilterTag(String tag)
Function	Add a tag filter.
Parameters	Parameters: tag - Tag to filter.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.3 Add package names to the whitelist

Prototype	int addWhiteList(String packageNames)
Function	Add package names to the whitelist.
Parameters	Parameters: packageNames - Package name.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.4 Clear the package blacklist

Prototype	int clearBlackList()
Function	Clears the package blacklist.
Parameters	
Return Value	Return:

	0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.5 Clear all tag filters

Prototype	int clearFilterTag()
Function	Clear all tag filters.
Parameters	
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.6 Clear the package whitelist

Prototype	int clearWhiteList()
Function	Clears the package whitelist.
Parameters	
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.7 Commit or disable log filtering

Prototype	int confirmLogFilter(boolean logFilter)
Function	Commit or disable log filtering.
Parameters	Parameters: logFilter - true: Commit and enable the current filter configuration; false: Disable tag filtering (set to all).
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	true commits changes added through add/clear tag, level, black/white list, and submitPackageFilter. false immediately disables tag filtering.

3.8.8 Compress logs by date and time

Prototype	int createZipByDateTime(long startDateTime, long endDateTime, OnCompressListener listener)
Function	Compress logs by date and time.
Parameters	Parameters: startDateTime - Start timestamp(ms). endDateTime - End timestamp(ms).

	listener - Status/compression callback. Do not manipulate Android Views directly on the Binder callback thread.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	Only one ZIP export is retained; the previous ZIP is deleted before a new one is created. If no logs exist in the entire time range, an error is returned through the callback. Final result is delivered through OnCompressListener.

3.8.9 Enable or disable background log collection

Prototype	int enableLog(boolean enable)
Function	Enable or disable background log collection.
Parameters	Parameters: enable - true: start recording logs; false: stop recording logs.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	Enabled by default on both production and developer devices.

3.8.10 Get the package blacklist

Prototype	ArrayList<String> getBlackList()
Function	Get the package blacklist.
Parameters	
Return Value	Return: ArrayList<String> - package blacklist,will be null when service is not bound or an exception occurs.
Notes	

3.8.11 Get the device log directory path

Prototype	String getDeviceLogsPath()
Function	Get the device log directory path.
Parameters	
Return Value	Return: String - Readable log directory path,will be null when the service is not bound or an exception occurs.
Notes	

3.8.12 Get the log level filter

Prototype	String getFilterLevel()

Function	Get the log level filter.
Parameters	
Return Value	Return: String — Log level filter; returns null when the service is not bound or an exception occurs.
Notes	

3.8.13 Get the tag filter list

Prototype	ArrayList<String> getFilterTag()
Function	Get the tag filter list.
Parameters	
Return Value	Return: ArrayList<String> - Get the tag filter list (may be empty); null when the service is not bound or an exception occurs.
Notes	

3.8.14 Get the package filtering mode

Prototype	PackageFilterMode getPackageFilterMode()
Function	Get the package filtering mode.
Parameters	
Return Value	Return: PackageFilterMode.WHITE_LIST; PackageFilterMode.BLACK_LIST; PackageFilterMode.OFF (default).
Notes	

3.8.15 Get the package whitelist

Prototype	ArrayList<String> getWhiteList()
Function	Get the package whitelist.
Parameters	
Return Value	Return: ArrayList<String> - Get the package whitelist (may be empty); null when the service is not bound or an exception occurs.
Notes	

3.8.16 Initialize the log configuration

Prototype	int initLogConfig(int bufferSize, int totalSize, int fileSize, String logDir)
Function	Initialize the log configuration.
Parameters	Parameters: bufferSize - Buffer size in bytes. totalSize - Total log storage in MB. fileSize - Single log file size in MB.

	logDir - Pass null/empty to use the SDK default path.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The configuration takes effect only after confirmLogFilter(true). Keep at least 10% of device storage free.

3.8.17 Set the log level filter

Prototype	int setFilterLevel(String level)
Function	Set the log level filter.
Parameters	Parameters: level - LogFilterLevel.DEBUG; LogFilterLevel.INFO; LogFilterLevel.WARN; LogFilterLevel.EXCEPTION (default) LogFilterLevel.OFF.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.18 Set the package filtering mode

Prototype	int submitPackageFilter(PackageFilterMode mode)
Function	Set the package filtering mode.
Parameters	Parameters: mode - PackageFilterMode.WHITE_LIST; PackageFilterMode.BLACK_LIST; PackageFilterMode.OFF(default)
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	The change takes effect only after confirmLogFilter(true).

3.8.19 Check the log collection state

Prototype	boolean isLogEnabled()
Function	Check the log collection state.
Parameters	
Return Value	Return: true: collecting logs;

	false: not collecting or unavailable.
Notes	

3.8.20 Business services obtain compressed logs and report upload results

Prototype	int reportUploadResult(boolean result)
Function	Business services obtain compressed logs and report upload results.
Parameters	Parameters: result - true: Called upon successful upload, the zip file will be deleted; false: Called upon upload failure, the zip file will be retained.
Return Value	Return: 0: Success; <0: failure(Refer to LogError)
Notes	For business services to retrieve compressed logs and report upload results. Logs are not automatically cleaned up without invoking the compression interface until the next compression trigger or successful upload report.

-- Log compression/status callback - OnCompressListener --

void onStatusChanged(int status)	Log collection status changed.
void onCompressFinished(boolean isCompleted, String message)	Log compression finished.
void onCompressReady(String zipFile)	Log ZIP is ready.

3.8.21 Log collection status changed

Prototype	void onStatusChanged(int status)
Function	Log collection status changed.
Parameters	Parameters: status - 0: stopped; 1: running.
Return Value	
Notes	

3.8.22 Log compression finished

Prototype	void onCompressFinished(boolean isCompleted, String message)
Function	Log compression finished.
Parameters	Parameters: isCompleted - Whether compression completed successfully. message - Compression result, no-file explanation, or failure reason.

Return Value	
Notes	

3.8.23 Log ZIP is ready

Prototype	void onCompressReady(String zipFile)
Function	Log ZIP is ready.
Parameters	Parameters: zipFile - Readable ZIP path, comma-separated.
Return Value	
Notes	

➢ 3.9 Perception Info module

-- get Perception Information Module getPerceptionInfoManager--

android.os.ParcelFileDescriptor collectPerceptionData()	Get perception data in file stream format
ArrayList<String> getBatteryCurrentMaxCapacity()	Get the current max capacity list of the main battery
ArrayList<String> getBatteryCycleCount()	Get the cycle count list of the main battery
ArrayList<String> getBatteryDesignCapacity()	Get the factory design capacity list of the main battery
ArrayList<String> getBatteryHealthPercent()	Get the battery health percentage list
ArrayList<String> getBatteryHealthStatus()	Get the battery health status list
ArrayList<String> getPrintDistance()	Get the print distance list
ArrayList<String> getSmallBatteryVoltage()	Get the small battery voltage list

3.9.1 Get perception data as a file stream

Prototype	android.os.ParcelFileDescriptor collectPerceptionData()
Function	Get perception data as a file stream
Parameters	
Return Value	Return: File descriptor for perception data; null if no data (file size < 2MB or not found)
Notes	Returns a file descriptor containing buried point (analytics) data.

3.9.2 Get large battery cycle count list

Prototype	ArrayList<String> getBatteryCycleCount()
Function	Get large battery cycle count list
Parameters	
Return Value	Return:

	List of cycle counts in the format: "yyyyMMdd,count" (e.g., "20250630,100")
Notes	Each item represents the total charge cycle count on a specific date.

3.9.3 Get large battery design capacity list

Prototype	ArrayList<String> getBatteryDesignCapacity()
Function	Get large battery design capacity list
Parameters	
Return Value	Return: List in the format: "yyyyMMdd,capacity" (e.g., "20250630,2800") capacity: Battery design capacity, in mAh.
Notes	Represents the designed capacity of the battery on a given date.

3.9.4 Get current max capacity of large battery

Prototype	ArrayList<String> getBatteryCurrentMaxCapacity()
Function	Get current max capacity of large battery
Parameters	
Return Value	Return: List in the format: "yyyyMMdd,capacity" (e.g., "20250630,2750") capacity: Actual current maximum charge capacity, in mAh.
Notes	Indicates the actual current maximum charge capacity over time.

3.9.5 Get battery health percentage list

Prototype	ArrayList<String> getBatteryHealthPercent()
Function	Get battery health percentage list
Parameters	
Return Value	Return: List in the format: "yyyyMMdd,healthPercent" (e.g., "20250630,100")
Notes	Battery health percentage over time.

3.9.6 Get battery health status list

Prototype	ArrayList<String> getBatteryHealthStatus()
Function	Get battery health status list
Parameters	
Return Value	Return: List in the format: "yyyyMMdd,status" (e.g., "20250630,0")
Notes	Health status values may refer to different health categories

3.9.7 Get small battery voltage list

Prototype	ArrayList<String> getSmallBatteryVoltage()

Function	Get small battery voltage list
Parameters	
Return Value	Return: List in the format: "yyyyMMdd,voltage" (e.g., "20250630,24"), where the actual voltage is voltage * 0.1 V (24 represents 2.4 V).
Notes	Voltage data for the small/internal battery. The returned voltage value is in units of 0.1 V.

3.9.8 Get print distance history list

Prototype	ArrayList<String> getPrintDistance()
Function	Get print distance history list
Parameters	
Return Value	Return:List in the format: "yyyyMMdd,distance" (e.g., "20250630,45000")
Notes	Represents cumulative printing distance (in meters) by date.

● 4. Error Code Definition

4.1 CertificationError

Error Code	Error Description	Error Value
CERTIFICATION_ERROR_INIT	Certification manager service initialization exception	-10000
CERTIFICATION_ERR_PARA_ERROR	Parameter exception	-10001
CERTIFICATION_CERT_LIST_CHECK_FAIL_ERROR	Certificate chain validation exception	-10002
CERTIFICATION_OTHER_ERROR	Other exceptions	-19999

4.2 DeviceError

Error Code	Error Description	Error Value
DEVICE_ERROR_INIT	Device manager service initialization exception	-20000
DEVICE_OTHER_ERROR	Other exceptions	-20001
DEVICE_PARAMETERS_INVALID	Invalid parameters	-20002

4.3 LocationError

Error Code	Error Description	Error Value
MANAGER_SERVICE_DISCONNECT	Device management service is not connected, please initialize the device management service first.	-30001
ERROR_INIT	Positioning module is not open or initialized.	-30002

ERROR_POSITIONING_PROGRESS	Positioning is in progress. If positioning is in progress, repeated positioning is prohibited. Stop positioning first and then restart it.	-30003
ERROR_UNKNOWN	Unknown error.	-30004
ERROR_PARAMETERS_INVALID	Invalid parameters.	-30005
ERROR_MISS_PERMISSIONS	Missing location permissions.	-30006
ERROR_CODE_INVALID_PARAMETER	Some important parameters are empty.	-31001
ERROR_CODE_FAILURE_WIFI_INFO	Positioning failed because the device only scanned a single WiFi, and the location information cannot be accurately calculated.	-31002
ERROR_CODE_FAILURE_LOCATION_PARAMETER	The obtained request parameters are empty, possibly due to an exception during the process.	-31003
ERROR_CODE_FAILURE_CONNECTION	Network connection exception. Detailed information can be obtained through AMapLocation.getLocationDetail().	-31004
ERROR_CODE_FAILURE_PARSER	XML parsing error.	-31005
ERROR_CODE_FAILURE_LOCATION	Positioning result error.	-31006
ERROR_CODE_FAILURE_AUTH	KEY error.	-31007
ERROR_CODE_UNKNOWN	Other errors.	-31008
ERROR_CODE_FAILURE_INIT	Initialization exception.	-31009
ERROR_CODE_SERVICE_FAIL	Positioning service startup failed. Please check if the service is configured and if the service tag in the manifest is placed inside the application tag.	-31010
ERROR_CODE_FAILURE_CELL	Incorrect base station information. Please check if the SIM card is installed.	-31011
ERROR_CODE_FAILURE_LOCATION_PERMISSION	Missing location permissions. Please check if location permissions are configured and enable location permissions in security software and settings.	-31012
ERROR_CODE_FAILURE_NOWIFIANDAP	Network positioning failed. Please check if the device has a SIM card inserted, mobile network is enabled, or the WiFi module is turned on.	-31013
ERROR_CODE_FAILURE_NOENOUGHSATELLITES	Satellite positioning failed due to insufficient available satellites.	-31014
ERROR_CODE_FAILURE_SIMULATION_LOCATION	The location may be simulated.	-31015
ERROR_CODE_AIRPLANEMODE_WIFIOFF	Positioning failed. Airplane mode is on	-31018

	and the WiFi switch is off. Please turn off airplane mode or turn on the WiFi switch.	
ERROR_CODE_NOCGI_WIFIOFF	Positioning failed. No SIM card detected and the WiFi switch is off. Please turn on the WiFi switch or insert a SIM card.	-31019
ERROR_CODE_FAILURE_COARSE_LOCATION	Positioning failed due to an exception under coarse permissions.	-31020
GEO_ERROR_CODE_INVALID_PARAMETER	Parameter error.	-32001
GEO_ERROR_CODE_FAILURE_CONNECTION	Network connection exception.	-32004
GEO_ERROR_CODE_FAILURE_PARSER	Data parsing failed (possibly due to connecting to a network that requires login but not logged in).	-32005
GEO_ERROR_CODE_FAILURE_AUTH	Authentication failed.	-32007
GEO_ERROR_CODE_UNKNOWN	Other unknown errors.	-32008
GEO_ERROR_NO_VALIDFENCE	No valid geofence available.	-32016
GEO_ERROR_CODE_EXISTS	The same fence already exists and does not need to be added again. Applies when the geofence's customID is the same.	-32017

4.4 NetworkError

Error Code	Error Description	Error Value
NETWORK_ERROR_INIT	Network module initialization error	-40000
NETWORK_OTHER_ERROR	Other exceptions	-40001
NETWORK_PARAMETERS_INVALID	Invalid parameters	-40002
NETWORK_NO_ACTIVE_SIM	No SIM card available	-40003

4.5 ResourceError

Error Code	Error Description	Error Value
RESOURCE_ERROR_INIT	Resource module initialization error	-50000
RESOURCE_OTHER_ERROR	Other error	-50001
RESOURCE_REGISTER_UPGRADE_LISTENER_ERROR	Failed to register MCU upgrade listener	-50002
RESOURCE_MCU_UPGRADE_FILE_PATH_ERROR	MCU upgrade file path error	-50003
RESOURCE_MCU_UPGRADE_FILE_ERROR	MCU upgrade file error	-50004
RESOURCE_MCU_UPGRADE_ERROR	MCU upgrade error	-50005
RESOURCE_APP_UPGRADE_ERROR	App upgrade error	-50006
RESOURCE_APP_UPGRADE_ERROR_ABORTED	App upgrade error — aborted	-50007
RESOURCE_APP_UPGRADE_ERROR_BLOCKED	App upgrade error — blocked	-50008
RESOURCE_APP_UPGRADE_ERROR_CONFLICT	App upgrade error — conflict	-50009
RESOURCE_APP_UPGRADE_ERROR_INCOMPATIBLE	App upgrade error — incompatible	-50010
RESOURCE_APP_UPGRADE_ERROR_INVALID	App upgrade error — invalid	-50011
RESOURCE_APP_UPGRADE_ERROR_STORAGE	App upgrade error — storage error	-50012

4.6 LogError

Error Code	Error Description	Error Value
ERR_NO_LOG	Log does not exist	-70000
ERR_LOG_ZIP	Failed to compress/export logs	-70001
ERR_LOG_FORMAT	Invalid format	-70002
ERR_NO_CONFIG	No configurable items (no pending configuration changes to submit)	-70003
ERR_TOTAL_SIZE	Failed to configure TotalSize (partial effect: other fields applied, totalSizeMb rejected)	-70004
ERR_NO_UPLOAD	No pending upload tasks	-70005

4.7 CommonError

Error Code	Error Description	Error Value
FEATURE_NOPERMISSION	Permission not requested	-10010
FEATURE_UNSUPPORTED	Feature not supported	-10011
OTHER_ERROR	Other error	-19999
PARAMETERS_INVALID	Invalid parameters	-10002
SERVICE_DISCONNECT	Service disconnected,please initialize the service	-10001
VERSION_NOT_MATCH	SDK version mismatch	-10000

● 5. Entity Class Definition

5.1 com.kozen.terminalmanager.Const

Constant Name	Type	Value
FUN_ERROR	int	-1
FUN_SUCC	int	0
TERMINAL_MANAGER_SERVICE_ACTION	String	"android.intent.action.XC_TERMINAL_MANAGER_SERVICE"
TERMINAL_MANAGER_SERVICE_CLASS	String	"com.kozen.terminalmanager.service.TerminalManagerService"
TERMINAL_MANAGER_SERVICE_PACKAGE	String	"com.kozen.terminalmanager.service"

5.2 com.kozen.terminalmanager.location.constant.LocationConstant

Constant Name	Type	Value
GEO_BUNDLE_KEY_CUSTOMID	String	"customId"
GEO_BUNDLE_KEY_FENCESTATUS	String	"event"

GEO_BUNDLE_KEY_LOCERRORCODE	String	"location_errorcode"
GEO_STATUS_IN	int	1
GEO_STATUS_LOCFAIL	int	4
GEO_STATUS_OUT	int	2
GPS_ACCURACY_BAD	int	0
GPS_ACCURACY_GOOD	int	1
GPS_ACCURACY_UNKNOWN	int	-1
GPS_STATUS_MODE_SAVING	int	3
GPS_STATUS_NOGPSPERMISSION	int	4
GPS_STATUS_NOGPSPROVIDER	int	1
GPS_STATUS_OFF	int	2
GPS_STATUS_OK	int	0
LOCATION_PROVIDER_GPS	String	"gps"
LOCATION_PROVIDER_LBS	String	"lbs"
LOCATION_TYPE_CELL	int	6
LOCATION_TYPE_COARSE_LOCATION	int	11
LOCATION_TYPE_FIX_CACHE	int	4
LOCATION_TYPE_GPS	int	1
LOCATION_TYPE_LAST_LOCATION_CACHE	int	9
LOCATION_TYPE_OFFLINE	int	8
LOCATION_TYPE_SAME_REQ	int	2
LOCATION_TYPE_WIFI	int	5
TRUSTED_LEVEL_BAD	int	4
TRUSTED_LEVEL_HIGH	int	1
TRUSTED_LEVEL_LOW	int	3
TRUSTED_LEVEL_NORMAL	int	2

5.3 LocationConstant.GeoLanguage

Enum Constant	TYPE	Description
DEFAULT	ENUM	Returns reverse geocoding info in local language
EN	ENUM	Always returns reverse geocoding info in English
ZH	ENUM	Always returns reverse geocoding info in Chinese

5.4 LocationConstant.LocationMode

Enum Constant	TYPE	Description
Battery_Saving	ENUM	Low-power positioning mode
Device_Sensors	ENUM	Device-only positioning mode
Hight_Accuracy	ENUM	High-accuracy positioning mode

5.5 LocationConstant.SignalMode

Enum Constant	TYPE	Description
BEIDOU_FIRST	ENUM	In high-accuracy positioning mode, performs a single

		location operation and prioritizes returning BeiDou satellite positioning information.
DEFAULT	ENUM	In high-accuracy positioning mode, performs a single location operation, and the system returns the first available positioning result.
GPS_FIRST	ENUM	In high-accuracy positioning mode, performs a single location operation and prioritizes returning GPS satellite positioning information

5.6 ApnConfiguration

Constant Name	TYPE	Description
apn	String	APN name.
authType	int	Authentication type.
current	boolean	Enable current APN.
mcc	String	Mobile Country Code (MCC).
mmsc	String	MMSC URL.
mmsPort	String	MMS proxy port.
mmsProxy	String	MMS proxy address.
mnc	String	MNC Mobile Network Code (MNC).
name	String	Entry name.
numeric	String	Operator Network Identification.
password	String	APN password.
port	String	Proxy port.
protocol	String	The protocol to use to connect to this APN.
proxy	String	Proxy address.
roaming_protocol	String	The protocol to use to connect to this APN when roaming.
server	String	Server address.
type	String	Comma-delimited list of APN types.
user	String	APN username.

5.7 WakeUpType

Enum Constant	TYPE	Description
TAP_OR_INSERT_CARD	ENUM	Wake up by IC card insertion or NFC card tapping.
SCREEN_DOUBLE_TAP	ENUM	Wake up by screen double tap.
LIFT_TO_WAKE	ENUM	Lift to wake screen.

5.8 PackageFilterMode

Enum Constant	TYPE	Description
WHITE_LIST	ENUM	Whitelist mode: print logs only for whitelisted package names.
BLACK_LIST	ENUM	Blacklist mode: do not print logs for blacklisted package names.

OFF	ENUM	Disable package-name filtering (default).

5.9 NetworkMode

Enum Constant	TYPE	Description
AUTO	ENUM	AUTO.
MODE_2G_ONLY	ENUM	2G ONLY.
MODE_3G_ONLY	ENUM	3G ONLY.
MODE_4G_ONLY	ENUM	4G ONLY.
MODE_4G_PREFERRED	ENUM	Prefer 4G.
MODE_5G_PREFERRED	ENUM	Prefer 5G.

5.10 WifiSecurityType

Enum Constant	TYPE	Description
OPEN	ENUM	Open network (no password).
WEP	ENUM	WEP encryption (legacy protocol with lower security).
WPA_PSK	ENUM	WPA/WPA2 pre-shared key encryption.
WPA3_SAE	ENUM	WPA3 Personal (SAE); requires Android 11 / API level 30 or later.

5.11 AttributeValue

Enum Constant	TYPE	Description
ATTRIBUTE_VALUE_RESOURCE_INFO_PROTOCOL_VERSION	ENUM	Resource package protocol version.
ATTRIBUTE_VALUE_RESOURCE_INFO_VERSION	ENUM	Resource package version.

5.12 com.kozen.terminalmanager.location.constant.LocationConstant.LocationType

Enum Constant	TYPE	Description
AMAP	ENUM	Amap Positioning
BAIDU	ENUM	Baidu Positioning

5.13 MacType

Enum Constant	TYPE	Description
BLUETOOTH	ENUM	Bluetooth MAC address
ETHERNET	ENUM	Ethernet MAC address
WIFI	ENUM	Wi-Fi MAC address

- 6. Access permission

Permission Name	Related Module	Tooltips

android.permission.SUPER_PERMISSIONS_DEVICE	DeviceManager	Read and modify system time, timezone, and other system settings
android.permission.SUPER_PERMISSIONS_RESOURCE	ResourceManager	Install or update resource packages (e.g., fonts, images, applications)
android.permission.SUPER_PERMISSIONS_LOCATION	LocationManager	Get device location (GPS / Cell tower)
android.permission.SUPER_PERMISSIONS_DEVICE_INFO	DeviceInfoManager	Collect device info (SN, IMEI, hardware)
android.permission.SUPER_PERMISSIONS_NETWORK	NetworkManager	Manage network access and firewall rules