



KOZEN Terminal Management
SDK Development Documentation
V1.5

Shanghai Xiangcheng Communication Technology Co., Ltd.

Table of Contents

● 1. Revision History.....	7
● 2. Overview.....	9
2.1 Introduction.....	9
2.2 Android version and IDE version supported by the SDK.....	9
2.3 Feature Introduction.....	9
2.3.1 Terminal Management Module.....	9
2.3.2 Certificate Management Module.....	9
2.3.3 Device Management Module.....	9
2.3.4 Device Information Management Module.....	9
2.3.5 Location Management Module.....	10
2.3.6 Network Management Module.....	10
2.3.7 Resource Management Module.....	10
2.3.8 Device Log Module.....	10
2.4 Importing the TerminalManagerService SDK.....	10
2.5 Initializing the TerminalManagerService SDK.....	11
● 3. API Interface Introduction.....	11
➤ 3.1 Terminal manager SDK initialization.....	12
3.1.1 Initialize the TerminalManagerService instance.....	12
-- TerminalManagerService instance callback - InitCallBack --.....	12
3.1.2 Initialization results.....	12
➤ 3.2 Certification module.....	12
-- Get Certification module - getCertificationManager--.....	12
3.2.1 Updates the application's signature certificate.....	13
3.2.2 Deletes the application's signature certificate.....	13
3.2.3 Get the application's signature certificate information.....	13
➤ 3.3 Device information module.....	13
-- Get device information module - getDeviceInfoManager--.....	13
3.3.1 Get the SDK service version number.....	14
3.3.2 Get Terminal serial number.....	14
3.3.3 Get the IMSI number.....	14
3.3.4 Get the IMEI number.....	14
3.3.5 Get the equipment supplier name (XC, Kozen).....	15
3.3.6 Get the device model.....	15
3.3.7 Get the OS version number.....	15
3.3.8 Get the Linux kernel version number.....	15
3.3.9 Get the MCU version number.....	15
3.3.10 Get the hardware version number.....	16
3.3.11 Get the EMV kernel version number.....	16
3.3.12 Get TUSN (Only for the China region).....	16
3.3.13 Get customer serial number.....	16
➤ 3.4 Device module.....	16
-- Get device module - getDeviceManager--.....	16

3.4.1 Sets the system time.....	17
3.4.2 Returns the current system time as a timestamp.....	18
3.4.3 Sets the time zone.....	18
3.4.4 Get the current time zone.....	18
3.4.5 Reboots the device.....	18
3.4.6 Shuts down the device.....	18
3.4.7 Schedules a timed reboot.....	19
3.4.8 Cancels the scheduled reboot.....	19
3.4.9 Enable or disable silent installation.....	19
3.4.10 Enforce runtime permission checks.....	19
3.4.11 Enforce runtime permission checks.....	20
3.4.12 Turn off the screen (sleep mode).....	20
3.4.13 Enable or disable various device wake-up methods.....	20
3.4.14 Get PCI reboot countdown time.....	20
3.4.15 Get the configured PCI reboot time.....	21
3.4.16 Get application permissions.....	21
3.4.18 Check whether is the developer mode.....	21
3.4.19 Synchronize system time from NTP.....	22
3.4.20 Synchronize time zone from network.....	22
3.4.21 Sets automatic system time synchronization.....	22
3.4.22 Sets automatic system time-zone synchronization.....	22
3.4.23 Show or hide a launcher application icon.....	23
3.4.24 Get the launcher application list.....	23
3.4.25 Enable or disable the screen reader.....	23
3.4.26 Get screen-reader switch state.....	23
3.4.27 Set the primary-screen wallpaper.....	24
3.4.28 Checks whether support EEPROM battery.....	24
3.4.29 Get battery charge cycle count.....	24
3.4.30 Get battery health percentage.....	24
3.4.31 Read EEPROM battery fields.....	24
➤ 3.5 Location module.....	25
Declaration on Location Interface.....	25
Our location interface is designed solely to provide location-related services to applications.	25
● It does not actively collect device information such as SN, IMEI, or hardware identifiers.	25
● It does not transmit any device information or location data to external servers without explicit application request and authorization.	25
● All data access is limited to the scope necessary for location service functionality.	25
This ensures that the interface complies with user privacy protection and security requirements.	25
-- Get Location module - getLocationManager--	25
3.5.1 Enables the location module and initializes related resources.....	26
3.5.2 Enables the location module and initializes related resources by key.....	26

3.5.3 Set location parameters.....	26
3.5.4 Start a single location request.....	27
3.5.5 Registers a location callback listener.....	27
3.5.6 Removes the location listener.....	27
3.5.7 Stop the location service.....	27
3.5.8 Creates a circular geofence.....	27
3.5.9 Registers a listener for geofence creation results.....	28
3.5.10 Set the action for receiving geofence status broadcasts.....	28
3.5.11 Removes all geofences.....	28
3.5.12 Add an APK to a block list.....	28
3.5.13 Remove an APK from the blocked list.....	29
3.5.14 Check if an APK is in block.....	29
3.5.15 Retrieves the list of blocked APKs.....	29
3.5.16 Disables the geofence feature.....	29
3.5.17 Enable the location module and initialize with the specified method.....	29
3.5.18 Enable the location module and initialize with the specified key and method.....	30
-- Location changed listener - ILocationChangedListener--.....	30
3.5.19 Location changed callback.....	30
-- GeoFence create listener - GeoFenceCreateListener --.....	30
3.5.20 Callback for geofence creation completion.....	31
➤ 3.6 Network module.....	31
-- Get network module - getNetworkManager--.....	31
3.6.1 Adds an APN configuration.....	32
3.6.2 Enables an APN configuration.....	32
3.6.3 Add a Wi-Fi network.....	32
3.6.4 Connects to a saved Wi-Fi network.....	32
3.6.5 Disconnect Wi-Fi.....	33
3.6.6 Forget a Wi-Fi network.....	33
3.6.7 Get Wi-Fi data usage statistics.....	33
3.6.8 Get saved Wi-Fi networks.....	33
3.6.9 Enable or disable Wi-Fi.....	33
3.6.10 Check Bluetooth support.....	34
3.6.11 Get Bluetooth state.....	34
3.6.12 Enable or disable Bluetooth.....	34
3.6.13 Set the mobile-data switch.....	34
3.6.14 Get the mobile-data switch state.....	35
3.6.15 Set data saver mode.....	35
3.6.16 Get data saver mode state.....	35
3.6.17 Get application data usage.....	35
3.6.18 Check SIM-card presence.....	36
3.6.19 Delete an APN.....	36
3.6.20 Get the APN list.....	36
3.6.21 Disable the active APN.....	36
➤ 3.7 Resource module.....	36
-- Get resource module - getResourceManager--.....	37

3.7.1	Installs or updates an app.....	37
3.7.2	Uninstalls an app.....	37
3.7.3	Updates the device system version or MCU firmware version.....	37
3.7.4	Updates the resource package.....	38
3.7.5	Install or update an app with listner.....	38
3.7.6	Update the resource package with listener.....	38
3.7.7	Update the device system version / MCU firmware version with listener.....	38
--	Upgrade listener - OnAppUpgradeListener--.....	39
3.7.8	Upgrade error/failure.....	39
3.7.9	Upgrade successful.....	39
--	Upgrade listener - OnUpdateCustomResListener --.....	39
3.7.10	Upgrade error/failure.....	39
3.7.11	Upgrade successful.....	39
--	Upgrade listener - OnUpdateOTALListener --.....	40
3.7.12	Upgrade error/failure.....	40
3.7.13	Upgrade successful.....	40
3.7.14	Callback for upgrade progress.....	40
➤	3.8 Device Log module.....	40
--	Get device log module - getDeviceLogsManager--.....	40
3.8.1	Add package names to the blacklist.....	41
3.8.2	Add a tag filter.....	41
3.8.3	Add package names to the whitelist.....	42
3.8.4	Clear the package blacklist.....	42
3.8.5	Clear all tag filters.....	42
3.8.6	Clear the package whitelist.....	42
3.8.7	Commit or disable log filtering.....	43
3.8.8	Compress logs by date and time.....	43
3.8.9	Enable or disable background log collection.....	43
3.8.10	Get the package blacklist.....	44
3.8.11	Get the device log directory path.....	44
3.8.12	Get the log level filter.....	44
3.8.13	Get the tag filter list.....	44
3.8.14	Get the package filtering mode.....	44
3.8.15	Get the package whitelist.....	45
3.8.16	Initialize the log configuration.....	45
3.8.17	Set the log level filter.....	45
3.8.18	Set the package filtering mode.....	46
3.8.19	Check the log collection state.....	46
3.8.20	Business services obtain compressed logs and report upload results.....	46
--	Log compression/status callback - OnCompressListener --.....	47
3.8.21	Log collection status changed.....	47
3.8.22	Log compression finished.....	47
3.8.23	Log ZIP is ready.....	47
➤	3.9 Perception Info module.....	47
--	Get Certification module - getCertificationManager--.....	47

3.9.1 Get perception data as a file stream.....	48
3.9.2 Get large battery cycle count list.....	48
3.9.3 Get large battery design capacity list.....	48
3.9.4 Get current max capacity of large battery.....	48
3.9.5 Get battery health percentage list.....	49
3.9.6 Get battery health status list.....	49
3.9.7 Get small battery voltage list.....	49
3.9.8 Get print distance history list.....	49
● 4. Error Code Definition.....	49
4.1 CertificationError.....	49
4.2 DeviceError.....	50
4.3 LocationError.....	50
4.4 NetworkError.....	51
4.5 ResourceError.....	52
4.6 LogError.....	52
● 5. Entity Class Definition.....	52
5.1 com.kozen.terminalmanager.Const.....	52
5.2 com.kozen.terminalmanager.location.constant.LocationConstant.....	53
5.3 LocationConstant.GeoLanguage.....	53
5.4 LocationConstant.LocationMode.....	53
5.5 LocationConstant.SignalMode.....	54
5.6 ApnConfiguration.....	54
5.7 WakeUpType.....	54
5.8 PackageFilterMode.....	55
5.9 NetworkMode.....	55
5.10 WifiSecurityType.....	55
● 6. Access permission.....	55

● 1. Revision History

Version	Release	Modify Record	Adapted SDK version	Author
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 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.	TerminalManagerService 1.4.x	Luna, Johnny
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	TerminalManagerService 1.2.x	Johnny

		setting. ✓ Add log retrieval. ✓ Adjust compatibility for retrieving the OS version number. ✓ Adjust the SDK icon		
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 listener ✓ Add module permission ✓ Add a security statement for location	TerminalManagerService 1.1.x	Johnny
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 is 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.

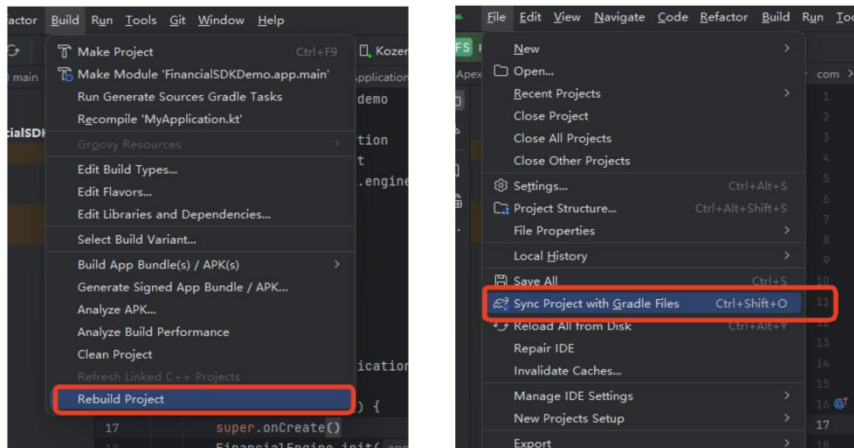
Add the following code to the `build.gradle` file:

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
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

<code>void init(Context context, InitCallBack callBack)</code>	Terminal manager SDK initialization
<code>ICertificationManager getCertificationManager()</code>	Certification module
<code>IDeviceInfoManager getDeviceInfoManager()</code>	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

<code>int updateAppSignature(String certData)</code>	Update the app signature certificate
--	--------------------------------------

3.2.1 Updates the application's signature certificate

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

3.2.2 Deletes the application's signature certificate

Prototype	<code>int deleteAppSignature(String certData)</code>
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 <code>CertificationError</code>).
Notes	For failure cases, refer to <code>CertificationError</code> for detailed error codes.

3.2.3 Get the application's signature certificate information

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

➤ 3.3 Device information module

-- Get device information module - `getDeviceInfoManager`--

<code>String getEmvKernelVersion()</code>	EMV kernel version number
<code>String getHardwareVersion()</code>	Get hardware version number
<code>String[] getImei()</code>	Get IMEI number; if multiple exist, return multiple
<code>String[] getImsi()</code>	Get IMSI number; if multiple exist, return multiple
<code>String getKernelVersion()</code>	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)

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

➤ 3.4 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

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

3.4.1 Sets the system time

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

3.4.2 Returns the current system time as a timestamp

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

3.4.3 Sets the time zone

Prototype	<code>int setTimeZone(String timeZone)</code>
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 <code>DeviceError</code>).
Notes	

3.4.4 Get the current time zone

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

3.4.5 Reboots the device

Prototype	<code>void reboot()</code>
Function	Reboots the device
Parameters	
Return value	
Notes	

3.4.6 Shuts down the device

Prototype	<code>void shutdown()</code>
Function	Shuts down the device.
Parameters	
Return value	
Notes	

3.4.7 Schedules a timed reboot

Prototype	<code>int setPCIR reboot(long time)</code>
Function	Schedules a timed reboot.
Parameters	<code>time</code> - Device will reboot after running for this duration (e.g., <code>1000 * 60 * 60</code> for 1 hour).
Return value	Return: 0: Success, Others: Failure (refer to <code>DeviceError</code>).
Notes	1. The policy will automatically run after reboot unless canceled by <code>cancelPCIR reboot</code> . 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	<code>int cancelPCIR reboot()</code>
Function	Cancels the scheduled reboot.
Parameters	
Return value	Return: 0: Success, Others: Failure (refer to <code>DeviceError</code>).
Notes	

3.4.9 Enable or disable silent installation

Prototype	<code>void setSilentInstall(boolean isOpen)</code>
Function	Enable or disable silent installation
Parameters	Parameters: <code>isOpen</code> - 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	<code>void forcePermission(boolean isOpen)</code>
Function	Enforce runtime permission checks
Parameters	Parameters: <code>isOpen</code> - 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 Enforce runtime permission checks

Prototype	<code>int setScreenTimeOut(int timeOut)</code>
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 - FailureOthers , 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	<code>int sleep(boolean isConfirm)</code>
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	<code>int wakeUp(WakeUpType type, boolean open)</code>
Function	Enable or disable various device wake-up methods
Parameters	Parameters: type - Wake-up method enum value (see WakeUpType)enable true: Enable the wake-up method false: Disable the wake-up method
Return Value	Return: 0 - Success 1 - FailureOthers , See DeviceError for details
Notes	This method configures whether a specific device wake-up method is enabled or disabled.

3.4.14 Get PCI reboot countdown time

Prototype	<code>long getPCIRebootCountdownTime()</code>
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	<code>long getPCIReboot()</code>
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	<code>List<AppPermissionInfo> getAppPermissions(String packageName)</code>
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	<code>int setAppPermission(String packageName, String permissionName, boolean enable)</code>
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 is the developer mode

Prototype	<code>int isDeveloperMode()</code>
Function	Check whether is the developer mode.
Parameters	
Return Value	Return: 0: Production mode;

	1: Developer mode.
Notes	

3.4.19 Synchronize system time from NTP

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

3.4.20 Synchronize time zone from network

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

3.4.21 Sets automatic system time synchronization

Prototype	<code>int setAutoSystemTimeSyncEnabled(boolean enabled)</code>
Function	Sets automatic system time synchronization.
Parameters	Parameters: enabled – true: enable; false: disable.
Return Value	Return: 0: Success; <0: Failure (refer to <code>DeviceError</code>).
Notes	

3.4.22 Sets automatic system time-zone synchronization

Prototype	<code>int setAutoTimeZoneSyncEnabled(boolean enabled)</code>
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	<code>int setPrimaryScreenWallpaper(String path)</code>
Function	Set the primary-screen wallpaper.
Parameters	Parameters: path - Wallpaper file path.
Return Value	Return: 0: Success; <0: Failure (refer to <code>DeviceError</code>).
Notes	Will replaces the built-in wallpaper

3.4.28 Checks whether support EEPROM battery

Prototype	<code>int isSupportEepromBattery()</code>
Function	Checks whether support EEPROM battery.
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	<code>int getBatteryCycleCount()</code>
Function	Gets the battery charge cycle count.
Parameters	
Return Value	Return int: Charge cycle count.
Notes	

3.4.30 Get battery health percentage

Prototype	<code>int getBatteryHealth()</code>
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	<code>Map<String, String> readEepromBatteryField()</code>
Function	Reads EEPROM battery data fields.
Parameters	
Return Value	Retrun Map<data key, data value>

	key - EEPROM battery key information value - EEPROM battery value
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

<code>int stopLocation()</code>	Stop location services
<code>int unregisterLocationListener()</code>	Remove the location listener
<code>int addGeoFence(double longitude, double latitude, float radius, String customId)</code>	Create a circular geofence

3.5.1 Enables the location module and initializes related resources

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

3.5.2 Enables the location module and initializes related resources by key

Prototype	<code>int open(String key)</code>
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 <code>LocationError</code> for specific error codes.
Notes	If the system-signed key fails (location fails with error code <code>ERROR_CODE_FAILURE_AUTH</code>), Please contact Kozen FAE to generate a special API key. Then call <code>onDestroy</code> first, and use this API to reinitialize the location module by passing the API key.

3.5.3 Set location parameters

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

3.5.5 Registers a location callback listener

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

3.5.6 Removes the location listener

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

3.5.7 Stop the location service

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

3.5.8 Creates a circular geofence

Prototype	<code>int addGeoFence(double longitude, double latitude, float radius, String customId)</code>
-----------	--

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 <code>LocationError</code> for specific error codes.
Notes	Must be used after <code>open</code> is successful.

3.5.9 Registers a listener for geofence creation results

Prototype	<code>int registerGeoFenceCreateListener(IGeoFenceCreateListener listener)</code>
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 <code>LocationError</code> for specific error codes.
Notes	Must be used after <code>open</code> is successful.

3.5.10 Set the action for receiving geofence status broadcasts

Prototype	<code>int setGeoFenceResultAction(String action, String pkgName)</code>
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 <code>LocationError</code> for specific error codes.
Notes	Multiple settings will use the last one. Must be used after <code>open</code> is successful.

3.5.11 Removes all geofences

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

3.5.12 Add an APK to a block list

Prototype	<code>int addToBlockOpenAppList(String pkgName)</code>
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 <code>LocationError</code> for specific error codes.
Notes	

3.5.13 Remove an APK from the blocked list

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

3.5.14 Check if an APK is in block

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

-- Location changed listener - `ILocationChangeListener`--

<code>void onLocationChanged(MapLocation mapLocation, int errorCode, String errorDetail)</code>	Location changed callback
---	---------------------------

3.5.19 Location changed callback

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

-- GeoFence create listener - `GeoFenceCreateListener` --

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

3.5.20 Callback for geofence creation completion

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

➤ 3.6 Network module

-- Get network module - `getNetworkManager`--

<code>int addApn(ApnConfiguration config)</code>	Add an APN.
<code>int enableApn(String name)</code>	Enable an APN.
<code>void addWifiNetwork(String ssid, String password, WifiSecurityType security)</code>	Add a Wi-Fi network.
<code>void connectWifi(String ssid)</code>	Connects to a saved Wi-Fi network.
<code>void disconnectWifi()</code>	Disconnect Wi-Fi.
<code>void forgetWifi(String ssid)</code>	Forget a Wi-Fi network.
<code>List<DataUsageStats> getWifiDataUsageStats(Date fromTimestamp, Date toTimestamp)</code>	Get Wi-Fi data usage statistics.
<code>WifiNetwork[] getWifiNetworks()</code>	Get saved Wi-Fi networks.
<code>void toggleWifi(boolean enabled)</code>	Enable or disable Wi-Fi.
<code>boolean hasBluetooth()</code>	Check Bluetooth support.
<code>boolean isBluetoothActive()</code>	Get Bluetooth state.
<code>void toggleBluetooth(boolean enabled)</code>	Enable or disable Bluetooth.
<code>int setMobileDataEnabled(boolean enabled)</code>	Set the mobile-data switch.
<code>int isMobileDataEnabled()</code>	Get the mobile-data switch state.
<code>void setDataSaverAllowed(boolean allowed)</code>	Set data saver mode.
<code>boolean isDataSaverAllowed()</code>	Get data saver mode state.
<code>AppDataUsageInfo getAppDataUsage(String packageName, long startTime, long endTime)</code>	Get application data usage.
<code>int isSimCardPresent(int slotIndex)</code>	Check SIM-card presence.
<code>int deleteApn(String name)</code>	Delete an APN.

List<ApnConfiguration> getApnList()	Get the APN list.
int disableApn()	Disable the active APN.

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	<code>void disconnectWifi()</code>
Function	Disconnect Wi-Fi.
Parameters	
Return Value	
Notes	

3.6.6 Forget a Wi-Fi network

Prototype	<code>void forgetWifi(String ssid)</code>
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	<code>List<DataUsageStats> getWifiDataUsageStats(Date fromTimestamp, Date toTimestamp)</code>
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	<code>WifiNetwork[] getWifiNetworks()</code>
Function	Get saved Wi-Fi networks.
Parameters	
Return Value	Return: Saved Wi-Fi networks.
Notes	

3.6.9 Enable or disable Wi-Fi

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

3.6.15 Set data saver mode

Prototype	<code>void setDataSaverAllowed(boolean allowed)</code>
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	<code>boolean isDataSaverAllowed()</code>
Function	Get data saver mode state.
Parameters	
Return Value	Return: true: enabled; false: disabled.
Notes	

3.6.17 Get application data usage

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

3.6.18 Check SIM-card presence

Prototype	<code>int isSimCardPresent(int slotIndex)</code>
Function	Check SIM-card presence.
Parameters	Parameters: slotIndex – SIM Card slot
Return Value	Return: 1: Present; 0: Not present <0: Error (Refer to DeviceError).
Notes	

3.6.19 Delete an APN

Prototype	<code>int deleteApn(String name)</code>
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	<code>List<ApnConfiguration> getApnList()</code>
Function	Get the APN list.
Parameters	
Return Value	Return: APN configuration list.
Notes	

3.6.21 Disable the active APN

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

➤ 3.7 Resource module

-- Get resource module - getResourcesManager--

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

3.7.1 Installs or updates an app

Prototype	<code>int installOrUpdate(String path)</code>
Function	Installs or updates an app.
Parameters	<code>path</code> - The path of the app installation package.
Return value	Return: 0: Success Others: Failure (refer to <code>ResourceError</code>).
Notes	

3.7.2 Uninstalls an app

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

3.7.3 Updates the device system version or MCU firmware version

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

3.7.4 Updates the resource package

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

3.7.5 Install or update an app with listner

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

3.7.6 Update the resource package with listener

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

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

Prototype	<code>int updateOTAWithListener(String path, OnUpdateOTAListener listener)</code>
Function	Update the device system version / MCU firmware version with listener
Parameters	Parameters: <code>path</code> - 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	

-- Upgrade listener - OnAppUpgradeListener--

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

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()
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 - getDeviceLogsManager--

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.

<code>int clearFilterTag()</code>	Clear all tag filters.
<code>int clearWhiteList()</code>	Clear the package whitelist.
<code>int confirmLogFilter(boolean logFilter)</code>	Commit or disable log filtering.
<code>int createZipByDateTime(long startDateTime, long endDateTime, OnCompressListener listener)</code>	Compress logs by date and time.
<code>int enableLog(boolean enable)</code>	Enable or disable background log collection.
<code>ArrayList<String> getBlackList()</code>	Get the package blacklist.
<code>String getDeviceLogsPath()</code>	Get the device log directory path.
<code>String getFilterLevel()</code>	Get the log level filter.
<code>ArrayList<String> getFilterTag()</code>	Get the tag filter list.
<code>PackageFilterMode getPackageFilterMode()</code>	Get the package filtering mode.
<code>ArrayList<String> getWhiteList()</code>	Get the package whitelist.
<code>int initLogConfig(int bufferSize, int totalSize, int fileSize, String logDir)</code>	Initialize the log configuration.
<code>int setFilterLevel(String level)</code>	Set the log level filter.
<code>int submitPackageFilter(PackageFilterMode mode)</code>	Set the package filtering mode .
<code>boolean isLogEnabled()</code>	Check the log collection state.
<code>int reportUploadResult(boolean result)</code>	Business services obtain compressed logs and report upload results.

3.8.1 Add package names to the blacklist

Prototype	<code>int addBlackList(String packageNames)</code>
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 <code>confirmLogFilter(true)</code> .

3.8.2 Add a tag filter

Prototype	<code>int addFilterTag(String tag)</code>
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	<code>int confirmLogFilter(boolean logFilter)</code>
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	<code>int createZipByDateTime(long startDateTime, long endDateTime, OnCompressListener listener)</code>
Function	Compress logs by date and time.
Parameters	Parameters: startDateTime – Start timestamp. endDateTime – End timestamp. 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	<code>int enableLog(boolean enable)</code>
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	<code>ArrayList<String> getBlackList()</code>
Function	Get the package blacklist.
Parameters	
Return Value	Return: <code>ArrayList<String></code> – 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	<code>String getDeviceLogsPath()</code>
Function	Get the device log directory path.
Parameters	
Return Value	Return: <code>String</code> – 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	<code>String getFilterLevel()</code>
Function	Get the log level filter.
Parameters	
Return Value	Return: <code>String</code> — 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	<code>ArrayList<String> getFilterTag()</code>
Function	Get the tag filter list.
Parameters	
Return Value	Return: <code>ArrayList<String></code> – 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	<code>PackageFilterMode getPackageFilterMode()</code>
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 Certification module - getCertificationManager--

android.os.ParcelFileDescriptor	Get perception data in file stream format
---------------------------------	---

<code>collectPerceptionData()</code>	
<code>ArrayList<String> getBatteryCurrentMaxCapacity()</code>	Get the current max capacity list of the main battery
<code>ArrayList<String> getBatteryCycleCount()</code>	Get the cycle count list of the main battery
<code>ArrayList<String> getBatteryDesignCapacity()</code>	Get the factory design capacity list of the main battery
<code>ArrayList<String> getBatteryHealthPercent()</code>	Get the battery health percentage list
<code>ArrayList<String> getBatteryHealthStatus()</code>	Get the battery health status list
<code>ArrayList<String> getPrintDistance()</code>	Get the print distance list
<code>ArrayList<String> getSmallBatteryVoltage()</code>	Get the small battery voltage list

3.9.1 Get perception data as a file stream

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

3.9.4 Get current max capacity of large battery

Prototype	<code>ArrayList<String> getBatteryCurrentMaxCapacity()</code>
Function	Get current max capacity of large battery
Parameters	
Return Value	Return:

	List in the format: "yyyyMMdd,capacity" (e.g., "20250630,2750")
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")
Notes	Voltage data for the small/internal battery.

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	-10000

	initialization exception	
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_NOENOUGHSAATELLITES	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 and the WiFi switch is off. Please turn off airplane mode or turn on the WiFi switch.	-31018
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

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

● 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.Termi

		nalManagerService"
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
High_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
PREFER_2G	ENUM	Prefer 2G.
PREFER_3G	ENUM	Prefer 3G.
PREFER_4G	ENUM	Prefer 4G.
PREFER_5G	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.

● 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