**Urovo Interface Description Document**

This material and all content contained in it are the property of Shenzhen UROVO TECHNOLOGY CO, LTD. and are protected by Chinese laws and applicable international conventions related to copyright. Without the written authorization of UROVO, no one is allowed to copy, disseminate, distribute, alter or otherwise use part or all of this material in any form. Violators will be held accountable according to law.

目录

[Revision history III](#_Toc6378)

[Directory V](#_Toc14595)

[1. API description 1](#_Toc25417)

[1.1. ISystemInfo 1](#_Toc1210)

[1.1.1. getDeviceId（UTMS&UMS） 1](#_Toc16335)

[1.1.2. getBatterySerialNo（UTMS&UMS） 1](#_Toc28611)

[1.2. INavigation 1](#_Toc15865)

[1.2.1. enableHomeKey（UTMS&UMS） 1](#_Toc29714)

[1.2.2. setLeftKeyEnabled（UTMS&UMS） 1](#_Toc32459)

[1.2.3. getLeftKeyEnabled() 2](#_Toc27146)

[1.2.4. getHomeKeyEnabled（UTMS&UMS） 2](#_Toc3773)

[1.2.5. setRightKeyEnabled（UTMS&UMS） 2](#_Toc4928)

[1.2.6. getRightKeyEnabled 2](#_Toc31758)

[1.3. IDateAndTime 2](#_Toc13330)

[1.3.1. setCurrentTime 3](#_Toc29810)

[1.4. IStatusBar 3](#_Toc31487)

[1.4.1. enableStatusBar 3](#_Toc19838)

[1.5. IWifi 3](#_Toc11258)

[1.5.1. enableWifi 3](#_Toc13642)

[1.5.2. connectWifi 3](#_Toc30467)

[1.5.3. disconnectCurrentWifi 4](#_Toc29963)

[1.5.4. removeSavedWifiNetwork 4](#_Toc23753)

[1.6. IApn 4](#_Toc15811)

[1.6.1. setAPN（UTMS&UMS） 4](#_Toc22821)

[1.6.2. queryAPN（UTMS&UMS） 5](#_Toc30524)

[1.6.3. deleteAPN（UTMS&UMS） 5](#_Toc2736)

[1.7. IDevice 5](#_Toc3034)

[1.7.1. shutdown 5](#_Toc1984)

[1.7.2. wipeData 6](#_Toc16398)

[1.7.3. enableMobileData 6](#_Toc21647)

[1.7.4. enableBluetooth 6](#_Toc23163)

[1.7.5. enableAirplaneMode 6](#_Toc19291)

[1.7.6. enableNFC 7](#_Toc18943)

[1.7.7. startLocalOsUpdate 7](#_Toc3771)

[1.7.8. setBrightness 7](#_Toc8509)

[1.7.9. setDefaultSimData 7](#_Toc8072)

[1.7.10. controlDefaultSimData 7](#_Toc15206)

[1.8. IAppInstall 8](#_Toc15103)

[1.8.1. installApplication 8](#_Toc15436)

[1.8.2. installApplication（retrun result）（UTMS&UMS） 8](#_Toc2827)

[1.8.3. installUpdate 8](#_Toc28165)

[1.8.4. uninstallApplication 9](#_Toc16864)

[1.8.5. uninstallApplication（retrun result）（UTMS&UMS） 9](#_Toc20646)

[1.9. IHotspot 9](#_Toc17215)

[1.9.1. controlHotspot 9](#_Toc5345)

[1.9.2. setHotspotConfig 10](#_Toc20773)

[1.10. IPermission 10](#_Toc16523)

[1.10.1. setGrantAllRequestedPermissionsApps 10](#_Toc7862)

[1.10.2. getGrantAllRequestedPermissionsApps 11](#_Toc29778)

[1.11. IBattery 11](#_Toc919)

[1.11.1. getSerialNumber 11](#_Toc17454)

[1.11.2. getBatteryManufactureDate 11](#_Toc6944)

[1.11.3. getRatedCapacity 11](#_Toc24662)

[1.11.4. getCurrentCapacity 11](#_Toc13830)

[1.11.5. getCurrentCharge 12](#_Toc31409)

[1.11.6. getBatteryHealth 12](#_Toc31342)

[1.11.7. getCycleCount 12](#_Toc29994)

[1.11.8. getTemp 12](#_Toc11660)

[1.11.9. isSmartBattery 12](#_Toc1872)

[1.11.10. getPN 13](#_Toc15159)

[1.11.11. getBatteryFirstUseDate 13](#_Toc15900)

Revision history

**\*A － add M － revise D －** **delete**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | date | A  M  D | Title or brief description |
| V1.0.0 | 2023.09.11 | A | Original version |
| V1.0.1 | 2023.09.18 | A |  |
| V1.0.3 | 2025.03.17 | A | Add Battery Info |
| V1.0.4 | 2025.04.15 |  | Add iWIFI,and bluetooth,nfc,mobiledata,wifi,airplanemode open/close ,os update,screenbrightness setting |
| V1.0.5 | 2025.04.27 |  | Add sim data control |
| V1.0.6 | 2025.07.29 | A | Add 1.8.3 installUpdate  Add 1.7.5 pairBluetooth  Add 1.7.12 launchIntent(Intent intent) |
|  |  |  |  |

Directory

[Revision history III](#_Toc30304)

[Directory IV](#_Toc24595)

[1. API description 1](#_Toc17927)

[1.1. ISystemInfo 1](#_Toc4596)

[1.1.1. getDeviceId（UTMS&UMS） 1](#_Toc16385)

[1.1.2. getBatterySerialNo（UTMS&UMS） 1](#_Toc7884)

[1.2. INavigation 1](#_Toc11335)

[1.2.1. enableHomeKey（UTMS&UMS） 1](#_Toc6195)

[1.2.2. setLeftKeyEnabled（UTMS&UMS） 1](#_Toc8585)

[1.2.3. getLeftKeyEnabled() 2](#_Toc16298)

[1.2.4. getHomeKeyEnabled（UTMS&UMS） 2](#_Toc4749)

[1.2.5. setRightKeyEnabled（UTMS&UMS） 2](#_Toc16129)

[1.2.6. getRightKeyEnabled 2](#_Toc894)

[1.3. IDateAndTime 2](#_Toc27881)

[1.3.1. setCurrentTime 3](#_Toc3764)

[1.4. IStatusBar 3](#_Toc18203)

[1.4.1. enableStatusBar 3](#_Toc23743)

[1.5. IWifi 3](#_Toc22204)

[1.5.1. enableWifi 3](#_Toc4533)

[1.5.2. connectWifi 3](#_Toc6379)

[1.5.3. disconnectCurrentWifi 4](#_Toc536)

[1.5.4. removeSavedWifiNetwork 4](#_Toc5266)

[1.6. IApn 4](#_Toc21459)

[1.6.1. setAPN（UTMS&UMS） 4](#_Toc20468)

[1.6.2. queryAPN（UTMS&UMS） 5](#_Toc5140)

[1.6.3. deleteAPN（UTMS&UMS） 5](#_Toc11160)

[1.7. IDevice 5](#_Toc20757)

[1.7.1. shutdown 5](#_Toc5489)

[1.7.2. wipeData 6](#_Toc32530)

[1.7.3. enableMobileData 6](#_Toc24191)

[1.7.4. enableBluetooth 6](#_Toc27129)

[1.7.5. pairBluetooth 6](#_Toc3641)

[1.7.6. enableAirplaneMode 7](#_Toc16795)

[1.7.7. enableNFC 7](#_Toc10879)

[1.7.8. startLocalOsUpdate 7](#_Toc31847)

[1.7.9. setBrightness 8](#_Toc24550)

[1.7.10. setDefaultSimData 8](#_Toc27891)

[1.7.11. controlDefaultSimData 8](#_Toc5532)

[1.7.12. launchIntent(Intent intent) 8](#_Toc2276)

[1.8. IAppInstall 8](#_Toc4406)

[1.8.1. installApplication 9](#_Toc5102)

[1.8.2. installApplication（retrun result）（UTMS&UMS） 9](#_Toc12840)

[1.8.3.installUpdate 9](#_Toc14344)

[1.8.4.uninstallApplication 10](#_Toc18866)

[1.8.5.uninstallApplication（retrun result）（UTMS&UMS） 10](#_Toc7173)

[1.9. IHotspot 10](#_Toc31625)

[1.9.1. controlHotspot 10](#_Toc21554)

[1.9.2. setHotspotConfig 11](#_Toc13453)

[1.10. IPermission 11](#_Toc9385)

[1.10.1. setGrantAllRequestedPermissionsApps 11](#_Toc28732)

[1.10.2. getGrantAllRequestedPermissionsApps 11](#_Toc2938)

[1.11. IBattery 12](#_Toc2766)

[1.11.1. getSerialNumber 12](#_Toc30936)

[1.11.2. getBatteryManufactureDate 12](#_Toc3354)

[1.11.3. getRatedCapacity 12](#_Toc3559)

[1.11.4. getCurrentCapacity 12](#_Toc2226)

[1.11.5. getCurrentCharge 13](#_Toc16109)

[1.11.6. getBatteryHealth 13](#_Toc3546)

[1.11.7. getCycleCount 13](#_Toc23523)

[1.11.8. getTemp 13](#_Toc21067)

[1.11.9. isSmartBattery 13](#_Toc23228)

[1.11.10. getPN 14](#_Toc5848)

[1.11.11. getBatteryFirstUseDate 14](#_Toc22763)

# API description

## ISystemInfo

### getDeviceId（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getDeviceId() | | |
| Description | Get Device ID。 | | |
| parameter | name | type | tips |
|  |  |  |
| return | String | | |

### getBatterySerialNo（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getBatterySerialNo () | | |
| Description | Get Battery Serial Number。 | | |
| parameter | name | type | tips |
|  |  |  |
| return | String | | |

## INavigation

### enableHomeKey（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableHomeKey(boolean enable) | | |
| Description | enable/disable the Home key。 | | |
| parameter | name | type | tips |
| enable | boolean | true：enable  false：disable |
| return |  | | |

### setLeftKeyEnabled（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void setLeftKeyEnabled(boolean enabled) | | |
| Description | Controls whether the virtual button on the left is available。 | | |
| parameter | name | type | tips |
| enabled | boolean | true：enable  false：disable |
| return |  | | |

### getLeftKeyEnabled()

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean getLeftKeyEnabled() | | |
| Description | Get if the virtual button on the left is available。 | | |
| parameter | name | type | tips |
|  |  |  |
| return | boolean  true：enable  false：disable | | |

### getHomeKeyEnabled（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean getHomeKeyEnabled() | | |
| Description | Obtain whether the Home key is enabled。 | | |
| parameter | name | type | tips |
|  |  |  |
| return | boolean  true：enable  false：disable | | |

### setRightKeyEnabled（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void setRightKeyEnabled(boolean enabled) | | |
| Description | Get if the virtual button on the right is available。 | | |
| parameter | name | type | tips |
| enabled | boolean | true：enable  false：disable |
| return |  | | |

### getRightKeyEnabled

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean getRightKeyEnabled() | | |
| Description | Get whether the right virtual key is available. | | |
| parameter | name | type | tips |
|  |  |  |
| return | boolean  true：enable  false：disable | | |

## IDateAndTime

### setCurrentTime

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean setCurrentTime(long when) | | |
| Description | Set the system date and time。 | | |
| parameter | name | type | tips |
| when | long | Timestamp to be set (in milliseconds) |
| return | boolean  true：success  false：fail | | |

## IStatusBar

### enableStatusBar

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableStatusBar (boolean enable) | | |
| Description | Enable/disable status bar。 | | |
| parameter | name | type | tips |
| enable | boolean | true：enable  false：disable |
| return |  | | |

## IWifi

### enableWifi

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableWifi(boolean isOpen) | | |
| Description | Open/close Wifi。 | | |
| parameter | name | type | tips |
| isOpen | boolean | true：open  false：close |
| return |  | | |

### connectWifi

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean connectWifi(String ssid, String password, int type) | | |
| Description | Connect to the specified wifi. Supports NONE, WEP, WPA/WPA2-Personal, and WPA3-Persional encryption types, but does not support Enhaced Open type; For enterprise wifi, please use 3.3.12 connectEapWifi | | |
| parameter | name | type | tips |
| ssid | String | wifi name |
| password | String | password |
| type | int | Encryption type0:none password Empty  1:WEP  2：WPA/WPA2-个人常用类型  3:WPA3-Persional  4:FT\_PSK (SQ53Z/SQ83 802.11r wifi must be selected) |
| return | boolean  true：The save was successful  false：Save failed | | |

### disconnectCurrentWifi

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void disconnectCurrentWifi() | | |
| Description | disconnect connected wifi | | |
| parameter | name | type | tips |
| return |  | | |

### removeSavedWifiNetwork

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void removeSavedWifiNetwork(String ssid) | | |
| Description | Remove saved wifi | | |
| parameter | name | type | tips |
| ssid | String | wifi name |
| return |  | | |

## IApn

### setAPN（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean setAPN(String name, String apn, String proxy, int port, String user, String server, String password, String mmsc, String mcc, String mnc, String mmsproxy, int mmsport, int authtype, String type, String protocol, int bearer, String roamingprotocol, boolean current) | | |
| Description | To add the detailed parameters that need to be provided for Appen, then , Appern, Mike, and Mike are required fields.。 | | |
| parameter | name | type | tips |
| name | String | Apn name(Required) |
| apn | String | Apn(Required) |
| proxy | String | Proxy address |
| port | int | Proxy port |
| user | String | User name |
| server | String | server |
| password | String | password |
| mmsc | String | MMS Center |
| mcc | String | MCC of the SIM card (required), carrier country code |
| mnc | String | The mnc of the SIM card (required), the carrier network code |
| mmsproxy | String | MMS Proxy |
| mmsport | int | MMS port |
| authtype | int | The authentication type can be 0 None, 1 PAP, 2 CHAP, and 3 PAP&CHAP |
| type | String | APN Type |
| protocol | String | APN protocol, which can be taken as follows: IP, IPV6, IPV4V6 |
| bearer | int | Load-bearing system |
| roamingprotocol | String | APN Roaming Agreement |
| current | boolean | Fill in true |
| return | boolean  true：success  false：fail | | |

### queryAPN（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String queryAPN(String selection, String[] selectionArgs) | | |
| Description | Query the APNs that meet the criteria。 | | |
| parameter | name | type | tips |
| selection | String | Conditional statements |
| selectionArgs | String[] | Used to replace the selection? The value of the variable that is represented |
| return | String  Returns APNs that meet the conditions | | |

### deleteAPN（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public int deleteAPN(String where, String[] whereArgs) | | |
| Description | Deletes the specified APN。 | | |
| parameter | name | type | tips |
| where | String | To query the APN, you need to delete the where statement |
| whereArgs | String[] | Used to replace where? The value of the variable that is represented |
| return | int  Returns the number of APNs that have been successfully deleted | | |

## IDevice

### shutdown

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void shutdown(boolean reboot) | | |
| Description | Shut down/restart。 | | |
| parameter | name | type | tips |
| reboot | boolean | true：shut down  false：reboot |
| return |  | | |

### wipeData

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void wipeData(int flags) | | |
| Description | Clear data。 | | |
| parameter | name | type | tips |
| flag | int | 0：Does not erase the external SD card  1：External SD card |
| return |  | | |

### enableMobileData

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableMobileData(boolean isOpen) | | |
| Description | Open/close MobileData。 | | |
| parameter | name | type | tips |
| isOpen | boolean | True-open  False-close |
| return |  | | |

### enableBluetooth

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableBluetooth(boolean isOpen) | | |
| Description | Open/close Bluetooth。 | | |
| parameter | name | type | tips |
| isOpen | boolean | True-open  False-close |
| return |  | | |

### pairBluetooth

|  |  |  |  |
| --- | --- | --- | --- |
| 定义 | public boolean pairBluetooth(String bluetoothMacAddress, String bluetoothPIN, boolean silentlyPair ) | | |
| 说明 | Use the Bluetooth MAC address and PIN code to perform pairing. If silentlyPair is true, a silent pairing (no popup, automatic background pairing) is performed; otherwise, a normal pairing (with popup) is used.  Note:  If the Bluetooth MAC address or PIN code is empty, an exception will be thrown.  Pairing with a PIN code only works on devices with older Bluetooth stacks (e.g. Bluetooth 2.0 or 2.1). Newer Bluetooth versions may require vendor-specific solutions.  On older-stack devices:  silentlyPair = true ⇒ pairing happens silently in the background.  silentlyPair = false ⇒ a pairing dialog appears, but the pairing still proceeds automatically.  On newer-stack devices, since many do not use PIN-code entry, you can only control whether the pairing dialog is shown; whether the connection actually succeeds depends on whether the device accepts the pairing. | | |
| 参数 | 名称 | 类型 | 备注 |
| bluetoothMacAddress | String | MAC |
| bluetoothPIN | String | PIN |
| silentlyPair | boolean | Whether to perform silent pairing: true means silent pairing, false means non-silent pairing |
| 返回 | boolean  Whether the pairing was successful | | |

### enableAirplaneMode

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableAirplaneMode(boolean isOpen) | | |
| Description | Open/close AirplaneMode。 | | |
| parameter | name | type | tips |
| isOpen | boolean | True-open  False-close |
| return |  | | |

### enableNFC

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void enableNFC(boolean isOpen) | | |
| Description | Open/close NFC。 | | |
| parameter | name | type | tips |
| isOpen | boolean | True-open  False-close |
| return |  | | |

### startLocalOsUpdate

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void startLocalOsUpdate(Context context,String path) | | |
| Description | Use the OS update package to update the local OS | | |
| parameter | name | type | tips |
| path | String | OTA upgrade package path |
| return |  | | |

### setBrightness

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void setBrightness(int light) | | |
| Description | Set the screen brightness | | |
| parameter | name | type | tips |
| light | Int | 1-255 |
| return |  | | |

### setDefaultSimData

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void setDefaultSimData(int simData) | | |
| Description | Set the preferred data SIM card | | |
| parameter | name | type | tips |
| simData | Int | 1. sim1 2. sim2 |
| return |  | | |

### controlDefaultSimData

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void controlDefaultSimData(int simData,boolean isEnable) | | |
| Description | control the data of SIM card | | |
| parameter | name | type | tips |
| simData | Int | 1. sim1 2. sim2 |
| isEnable | boolean | True-enable  False-disable |
| return |  | | |

### launchIntent(Intent intent)

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void launchIntent(Intent intent) | | |
| Description | Provide functionality to launch an Activity from a background thread and clear the background execution restrictions on that Intent. | | |
| parameter | name | type | tips |
| intent | Intent | The Intent used to launch an Activity. |
| return |  | | |

## IAppInstall

### installApplication

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean installApplication(String apkFilePath) | | |
| Description | Install apps silently。 | | |
| parameter | name | type | tips |
| apkFilePath | String | Apk file path |
| return | boolean  true：success  false：fail | | |

### installApplication（retrun result）（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean installApplication(String apkFilePath, boolean installOnSDCard, IPackageInstallObserver observer) { | | |
| Description | Install apps silently。 | | |
| parameter | name | type | tips |
| apkFilePath | String | Apk file path |
| installOnSDCard | boolean | true: installed on the SD card  false: is not installed on the SD card  This parameter is deprecated |
| observer | IPackageInstallObserver | Listen for the return status of the app installation |
| return | boolean  true：success  false：fail | | |

### 1.8.3.installUpdate

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void installUpdate(String apkPath, Intent successIntent, IPackageInstallObserver errorCallback) | | |
| Description | Download the APK at the path specified by apkPath. If the installation succeeds, a broadcast with successIntent will be sent; if it fails, errorCallback will be invoked. Note: If apkPath is empty, an exception will be thrown immediately. | | |
| parameter | name | type | tips |
| apkPath | String | Apk file path |
| successIntent | Intent | Send a broadcast intent after the APK has been successfully installed.” |
| errorCallback | IPackageInstallObserver | 1. Listen for the return status of the app installation |
| return |  | | |

### 1.8.4.uninstallApplication

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean uninstallApplication(String packageName) | | |
| Description | Silently uninstall apps。 | | |
| parameter | name | type | tips |
| packageName | String | The name of the app package |
| return | boolean  true：success  false：fail | | |

### 1.8.5.uninstallApplication（retrun result）（UTMS&UMS）

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean uninstallApplication(String packageName, boolean keepDataAndCache, IPackageDeleteObserver observer) { | | |
| Description | Silently uninstall apps | | |
| parameter | name | type | tips |
| packageName | String | The name of the app package |
| keepDataAndCache | boolean | Whether to retain app data and cache |
| observer | IPackageDeleteObserver | Listen for the return status of the uninstall app |
| return | boolean  true：success  false：fail | | |

## IHotspot

### controlHotspot

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public boolean controlHotspot(boolean enable) | | |
| Description | Turn on/off hotspots | | |
| parameter | name | type | tips |
| enable | boolean | true：turn on  false：turn off |
| return | Boolean  true：success  false：fail | | |

### setHotspotConfig

|  |  |  |  |
| --- | --- | --- | --- |
| definition | Public void setHotspotConfig(String name,String password,int securityType , int apBands,boolean autoOpen) | | |
| Description | Set the hotspot | | |
| parameter | name | type | tips |
| name | String | Hotspot name |
| password | String | Hotspot password, at least 8 digits |
| securityType | Int | 0:None 1:WPA2-Persional 2:WPA2/WPA3-Persional 3:WPA3-Persional |
| apBands | Int | 0:2.4GHz 1:首选5.0GHz |
| autoOpen | Boolean | true open false false |
| return | Boolean  true：success  false：fail | | |

## IPermission

### setGrantAllRequestedPermissionsApps

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public void setGrantAllRequestedPermissionsApps(String pkgName, int mode, int action) | | |
| Description | Set a list of apps that grant all application permissions Note: Permissions will be given when the API is called, and clearing or deleting will not revoke permissions. Apps in the list will be given permissions when they are installed or updated; Some permissions require a signature or privileged application to grant; | | |
| parameter | name | type | tips |
| pkgName | String | Package name； |
| mode | int | The default is 0 |
| action | int | 0: Clear 1: Add 2: Delete |
| return |  | | |

### getGrantAllRequestedPermissionsApps

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public List<String> getGrantAllRequestedPermissionsApps(int mode) | | |
| Description | Get a list of apps that grant full application permissions | | |
| parameter | name | type | tips |
| mode | int | The default is 0 |
| return | List<String>  List of lists | | |

## IBattery

### getSerialNumber

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getSerialNumber() | | |
| Description | Get the battery SN | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getBatteryManufactureDate

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getBatteryManufactureDate() | | |
| Description | Get the battery production date | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getRatedCapacity

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getRatedCapacity() | | |
| Description | Obtain the battery rating capacity of mA | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getCurrentCapacity

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getCurrentCapacity() | | |
| Description | Obtain the battery available capacity of mA | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getCurrentCharge

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getCurrentCharge() | | |
| Description | Obtain the remaining available capacity mA | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getBatteryHealth

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getBatteryHealth() | | |
| Description | Obtain the battery health degree | | |
| parameter | name | type | tips |
|  |  |  |
| return | 1-100 | | |

### getCycleCount

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getCycleCount() | | |
| Description | Get the number of charges | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getTemp

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getTemp() | | |
| Description | Get the battery temperature | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### isSmartBattery

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String isSmartBattery() | | |
| Description | Get whether it is a smart battery | | |
| parameter | name | type | tips |
|  |  |  |
| return | True-yes  False-NO | | |

### getPN

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getPN() | | |
| Description | Get the battery PN | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |

### getBatteryFirstUseDate

|  |  |  |  |
| --- | --- | --- | --- |
| definition | public String getBatteryFirstUseDate() | | |
| Description | Get the battery's first use time | | |
| parameter | name | type | tips |
|  |  |  |
| return |  | | |