|  |
| --- |
|  |
| **M800 Android SDK Description** |
| **V1.0** |
|  |
| **Author: WangYong** |
| **2023/11/15** |

# Change log

|  |  |  |
| --- | --- | --- |
| **Date** | **Log** | **Author** |
| 2023-11-15 | Create | WangYong |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |
| --- |
|  |

# Catalog

[Change log 1](#_Toc150962655)

[Catalog 2](#_Toc150962656)

[Android SDK Intro 6](#_Toc150962657)

[1. Function and Purpose 6](#_Toc150962658)

[Integration 6](#_Toc150962659)

[1 .SDK framework 6](#_Toc150962660)

[2 .Integration 6](#_Toc150962661)

[Eclipse Config 6](#_Toc150962662)

[API 8](#_Toc150962663)

[1.API initialization 8](#_Toc150962664)

[Description 8](#_Toc150962665)

[Parameters 8](#_Toc150962666)

[2. Login to the server 8](#_Toc150962667)

[Description 8](#_Toc150962668)

[Parameters 8](#_Toc150962669)

[3. Query the device online status 8](#_Toc150962670)

[Description 9](#_Toc150962671)

[Parameters 9](#_Toc150962672)

[4. Query sleep status 9](#_Toc150962673)

[Description 9](#_Toc150962674)

[Parameters 9](#_Toc150962675)

[5. Query environment parameters 9](#_Toc150962676)

[Description 9](#_Toc150962677)

[Parameters 10](#_Toc150962678)

[6. Turn on real-time data 10](#_Toc150962679)

[Description 10](#_Toc150962680)

[Parameters 10](#_Toc150962681)

[7. Turn off real-time data 10](#_Toc150962682)

[Description 10](#_Toc150962683)

[Parameters 10](#_Toc150962684)

[8. End monitoring manually 11](#_Toc150962685)

[Description 11](#_Toc150962686)

[Parameters 11](#_Toc150962687)

[9. Device firmware upgrade 11](#_Toc150962688)

[Description 11](#_Toc150962689)

[Parameters 11](#_Toc150962690)

[10. Register device online status change listener 12](#_Toc150962691)

[Description 12](#_Toc150962692)

[Parameters 12](#_Toc150962693)

[11. Unregister device online status change listener 12](#_Toc150962694)

[Description 12](#_Toc150962695)

[Parameters 12](#_Toc150962696)

[12. Register sleep report upload status listener 12](#_Toc150962697)

[Description 12](#_Toc150962698)

[Parameters 13](#_Toc150962699)

[13. Unregister sleep report upload status listener 13](#_Toc150962700)

[Description 13](#_Toc150962701)

[Parameters 13](#_Toc150962702)

[14. Register real-time data listener 13](#_Toc150962703)

[Description 13](#_Toc150962704)

[Parameters 13](#_Toc150962705)

[15. unregister real-time data listener 14](#_Toc150962706)

[Description 14](#_Toc150962707)

[Parameters 14](#_Toc150962708)

[16. Register for real-time sleep state monitoring 14](#_Toc150962709)

[Description 14](#_Toc150962710)

[Parameters 14](#_Toc150962711)

[17. Unregister for real-time sleep state monitoring 14](#_Toc150962712)

[Description 15](#_Toc150962713)

[Parameters 15](#_Toc150962714)

[18.Query the device collection status 15](#_Toc150962715)

[Interface Description 15](#_Toc150962716)

[Parameters Description 15](#_Toc150962717)

[Object Description 15](#_Toc150962718)

[StatusCode 15](#_Toc150962719)

[Description 15](#_Toc150962720)

[Fields 15](#_Toc150962721)

[IResultCallback<T> 16](#_Toc150962722)

[Description 16](#_Toc150962723)

[Function 16](#_Toc150962724)

[CallbackData<T> 16](#_Toc150962725)

[Description 16](#_Toc150962726)

[Fields 16](#_Toc150962727)

[SleepState 17](#_Toc150962728)

[Description 17](#_Toc150962729)

[Fields 17](#_Toc150962730)

[CollectState 17](#_Toc150962731)

[Description 17](#_Toc150962732)

[Fields 17](#_Toc150962733)

[RealTimeData 17](#_Toc150962734)

[Description 17](#_Toc150962735)

[Fields 18](#_Toc150962736)

[HistoryData 18](#_Toc150962737)

[Description 18](#_Toc150962738)

[Fields 18](#_Toc150962739)

[Summary 18](#_Toc150962740)

[Description 18](#_Toc150962741)

[Fields 19](#_Toc150962742)

[Detail 19](#_Toc150962743)

[Description 19](#_Toc150962744)

[Fields 19](#_Toc150962745)

[Analysis 20](#_Toc150962746)

[Description 20](#_Toc150962747)

[Fields 20](#_Toc150962748)

[SleepStatusType 23](#_Toc150962749)

[Description 23](#_Toc150962750)

[Fields 23](#_Toc150962751)

[EnvironmentData 23](#_Toc150962752)

[Description 23](#_Toc150962753)

[Fields 23](#_Toc150962754)

[OnlineStateListener 23](#_Toc150962755)

[Interface Introduction 23](#_Toc150962756)

[Method description 24](#_Toc150962757)

[SleepReportUploadStateListener 24](#_Toc150962758)

[Interface Introduction 24](#_Toc150962759)

[Method description 24](#_Toc150962760)

[RealtimeDataListener 24](#_Toc150962761)

[Interface Introduction 24](#_Toc150962762)

[Method description 25](#_Toc150962763)

[RealtimeSleepStateListener 25](#_Toc150962764)

[Interface Introduction 25](#_Toc150962765)

[Method description 25](#_Toc150962766)

# Android SDK Intro

## Function and Purpose

RestOn SDK,launched by Sleepace, is a software development kit for fast Internet APP development on android platform.

The SDK encapsulates the communication process between APP and hardware, and provides functions such as device configuration, device control and data query. Using the SDK, users do not need to care about complex communication protocols and the underlying implementation, only need to focus on the APP interaction and business level.

# Integration

## 1 .SDK framework

|  |  |
| --- | --- |
| **Framework** | **Description** |
| sdkcore.jar | SDK base core |
| wifidevicecommonsdk.jar | General HTTP interface package for WiFi devices |
| m800sdk.jar | M800 SDK |
| gson-2.8.0.jar | JSON parsing Toolkit (version 2.8.0 and below, there may be exceptions when compiling in the eclipse environment) |

## 2 .Integration

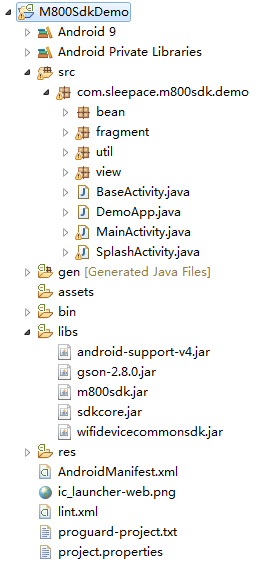
There are many Android development tools, and here we introduce the engineering configuration method of Sleepace SDK with Eclipse.

## Eclipse Config

**Step 1**：

In the project to create a "libs" folder, copy sdkcore.jar, wifidevicecommonsdk.jar, m800sdk.jar, gson-2.8.0.jar to "libs" folder.

Like this:



**Setp 2:**

Config the “AndroidManifest.xml”

<uses-permission android:name=*"android.permission.INTERNET"* />

<uses-permission android:name=*"android.permission.WRITE\_EXTERNAL\_STORAGE"*/>

<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"* />

# API

## 1.API initialization

M800Helper.getInstance(Context mContext);

### Description

M800Helper Initialization

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| mContext | Context | Context object. It is recommended to pass ApplicationContext |

## Login to the server

**public** **void** login(**final** String ip, **final** **int** port, **final** String sid, **final** String deviceId, **final** IResultCallback<Void> cb)

### Description

It is used to connect and log in to the server. IP, port and Sid are obtained through the authentication interface. Please refer to demo for specific usage

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| ip | String | Server IP |
| port | int | Server port |
| sid | String | Sid information |
| deviceId | String | Device ciphertext ID |
| cb | IResultCallback<Void> | Callback interface |

## Query the device online status

**public** **void** queryDeviceOnlineState(**final** **short** deviceType, **final** String deviceId, **final** IResultCallback<Byte> cb)

### Description

It is used to query the online status of equipment

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| cb | IResultCallback<Byte> | Callback interface, 0: not online 1: online 0xff: unknown state |

## Query sleep status

**public** **void** querySleepState(**final** **short** deviceType, **final** String deviceId, **final** **int** num, **final** IResultCallback<[SleepState](#_SleepState)> cb)

### Description

Used to query sleep status

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| num | int | Left and right side numbering, 0 left side (single person version), 1 right side |
| cb | IResultCallback< [SleepState](#_SleepState)> | Callback interface to return sleep state information. Please refer to SleepState for details |

## Query environment parameters

**public** **void** queryEnvironmentData(**final** String deviceId, **final** IResultCallback<[EnvironmentData](#_EnvironmentData_1)> cb)

### Description

Used to query environmental parameters, such as temperature and humidity

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceId | String | Device ciphertext ID |
| cb | IResultCallback< [EnvironmentData](#_EnvironmentData_1)> | Callback interface, return environment parameters, detailed reference: EnvironmentData |

## Turn on real-time data

**public** **void** startRealTimeData(**final** **short** deviceType, **final** String deviceId, **final** **int** num, **final** IResultCallback<[RealTimeData](#_RealTimeData_1)> cb)

### Description

Used to view real-time heart rate breathing

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| num | int | Left and right side numbering, 0 left side (single person version), 1 right side |
|  |  |  |
| cb | IResultCallback< [RealTimeData](#_RealTimeData_1)> | Callback interface. After successful operation, the callback interface will report heart rate and respiratory data in real time. For details, please refer to: RealtimeData |

## Turn off real-time data

**public** **void** stopRealTimeData(**final** **short** deviceType, **final** String deviceId, **final** **int** num, **final** IResultCallback<Void> cb)

### Description

It is used to turn off the real-time data. After shutdown, the device will not report the heart rate and respiratory data in real time, but it is still under monitoring.

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| num | int | Left and right side numbering, 0 left side (single person version), 1 right side |
| cb | IResultCallback<Void> | Callback interface |

## End monitoring manually

**public** **void** stopCollection(**final** **int** userId, **final** **short** deviceType, **final** String deviceId, **final** **int** num, **final** IResultCallback<Void> cb)

### Description

Used to end monitoring

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| userId | int | User ID |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| num | int | Left and right side numbering, 0 left side (single person version), 1 right side |
| cb | IResultCallback<Void> | Callback interface |

## Device firmware upgrade

**public** **void** upgradeDevice(**final** **float** deviceVersion, **final** **short** deviceType, **final** String deviceId, **final** IResultCallback<Integer> cb)

### Description

For device firmware upgrade

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceVersion | float | Firmware target version number |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| cb | IResultCallback<Integer> | Callback the interface. After the upgrade is successful, the interface returns the upgrade progress |

## Register device online status change listener

**public** **void** registOnlineStateListener([OnlineStateListener](#_OnlineStateListener) onlineStateListener)

### Description

It is used to monitor the online status of equipment

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [OnlineStateListener](#_OnlineStateListener) | Custom interface | Device online status monitoring, for details, refer to: OnlinestateListener |

## Unregister device online status change listener

**public** **void** unregistOnlineStateListener([OnlineStateListener](#_OnlineStateListener) onlineStateListener)

### Description

Used to log off device online status monitoring

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [OnlineStateListener](#_OnlineStateListener) | Custom interface | Device online status monitoring, for details, refer to: OnlinestateListener |

## Register sleep report upload status listener

**public** **void** registSleepReportUploadStateListener([SleepReportUploadStateListener](#_SleepReportUploadStateListener) sleepReportUploadState)

### Description

Used to monitor sleep report upload status

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [SleepReportUploadStateListener](#_SleepReportUploadState) | Custom interface | Sleep report upload state listener, for details, refer to: [SleepReportUploadStateListener](#_SleepReportUploadState) |

## Unregister sleep report upload status listener

**public** **void** unregistSleepReportUploadStateListener(SleepReportUploadStateListener onlineStateListener)

### Description

Used to log off sleep report upload status monitoring

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [SleepReportUploadStateListener](#_SleepReportUploadState) | Custom interface | Sleep report upload state listener, for details, refer to: [SleepReportUploadStateListener](#_SleepReportUploadState) |

## Register real-time data listener

**public** **void** registRealtimeDataListener([RealtimeDataListener](#_RealtimeDataListener) realtimeDataListener)

### Description

Used to monitor real-time data

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [RealtimeDataListener](#_RealtimeDataListener) | Custom interface | Real-time data monitoring, refer to: [RealtimeDataListener](#_RealtimeDataListener) |

## unregister real-time data listener

**public** **void** unregistRealtimeDataListener([RealtimeDataListener](#_RealtimeDataListener) realtimeDataListener)

### Description

Used to log off real-time data monitoring

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [RealtimeDataListener](#_RealtimeDataListener) | Custom interface | Real-time data monitoring, refer to: [RealtimeDataListener](#_RealtimeDataListener) |

## Register for real-time sleep state monitoring

**public** **void** registRealtimeSleepStateListener([RealtimeSleepStateListener](#_RealtimeSleepStateListener) realtimeSleepStateListener)

### Description

Used to monitor real-time sleep status

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [RealtimeSleepStateListener](#_RealtimeSleepStateListener) | Custom interface | Real-time data monitoring, refer to: [RealtimeSleepStateListener](#_RealtimeSleepStateListener) |

## Unregister for real-time sleep state monitoring

**public** **void** unregistRealtimeSleepStateListener([RealtimeSleepStateListener](#_RealtimeSleepStateListener) realtimeSleepStateListener)

### Description

Used to log off real-time sleep status monitoring

### Parameters

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| [RealtimeSleepStateListener](#_RealtimeSleepStateListener) | Custom interface | Real-time data monitoring, refer to: [RealtimeSleepStateListener](#_RealtimeSleepStateListener) |

## 18.Query the device collection status

**public** **void** queryCollectState(**final** **short** deviceType, **final** String deviceId, **final** **int** num, **final** IResultCallback<[CollectState](#_RealTimeData)> cb)

### Interface Description

It is used to query the equipment collection status. If the equipment collection is in progress, the interface will return the time stamp when the equipment started collection, which is used to calculate the equipment collection duration.

### Parameters Description

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| deviceType | short | Equipment type |
| deviceId | String | Device ciphertext ID |
| num | int | Left and right side numbering, 0 left side (single person version), 1 right side |
| cb | IResultCallback<[CollectState](#_CollectState)> | Callback interface |

# Object Description

## StatusCode

### Description

Status of execution

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value** | **Description** |
| STATUS\_SUCCESS | int | 0x00 | success |
| STATUS\_FAILED | int | -1 | failed |
| STATUS\_TIMEOUT | int | -2 | timeout |
| STATUS\_DISCONNECT | int | -3 | Bluetooth is disconnected |
| STATUS\_BLUETOOTH\_NOT\_OPEN | int | -4 | Bluetooth is not open |
| STATUS\_PARAMETER\_ERROR | int | -5 | Parameter error |

## IResultCallback<T>

### Description

Callback interface

### Function

**void** onResultCallback(CallbackData<T> cd)

callback function

## CallbackData<T>

### Description

Callback object

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| status | short | [Status](#_StatusCode) of execution |
| callbackType | short | Interface Type, used to distinguish between operating interface |
| result | T | The result of execution |

## SleepState

### Description

Sleep state

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| asleepFlag | byte | Falling asleep sign 0x00: Not falling asleep 0x01: Falling asleep Other: Invalid |
| wakeupFlag | byte | Awake sign 0x00: No wake-up 0x01: Wake-up Others: Invalid |
| outOfBedFlag | byte | Out of bed sign 0x00: Not out of bed 0x01: Out of bed Other: Invalid |
| sleepState | byte | Sleep depth 0x00: invalid (indicating that the function is not supported yet) 0x01: awake 0x02: light sleep 0x03: mid sleep 0x04: deep sleep |
| situpFlag | byte | Sitting up 0x00: lying down 0x01: sitting up |

## CollectState

### Description

Device collection status

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| state | byte | Device collection status 0x00: Not collected 0x01: Collecting |
| startTime | int | Time stamp at the beginning of acquisition, in seconds |

## RealTimeData

### Description

Real-time data

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| heartRate | short | Heart rate |
| breathRate | short | Respiration rate |
| status | byte | Status identification, see SleepStatusType for detailed description |
| statusValue | short | The value of status. Unit(second) |
| temp | byte | Temperature, valid only for equipment with temperature and humidity |
| humidity | byte | Humidity, valid only for equipment with temperature and humidity |
| sleepFlag | byte | Asleep or not  1: asleep  0: not |
| wakeFlag | byte | Awake critical logo :  1: yes  0: no |

## HistoryData

### Description

Historical data

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| summary | Summary | History data [Summary](#_Summary) |
| detail | Detail | History data [Detail](#_Detail) |
| analy | Analysis | [Analysis](#_Analysis) |

## Summary

### Description

Summary of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| recordCount | int | The length of Collecting.  Unit(minute)  Eg:  300. It means you collect for 300 minutes |
| startTime | int | Start time(timestamp). Unit(second) |
| stopMode | int | How to stop collecting：  0: Call the method “Stop Collecting”  1: stop automatically if you leave the bed for an hour  2: Error(a、Collect more than 24 hours，b、reston shutdown c、upgrade)  3: restart |
| timeStep | int | Record interval (default 60s, ie: 1 minute a time a point) |
| timezone | int | Timezone |

## Detail

### Description

Detail of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| breathRate | int[] | Breath rate |
| heartRate | int[] | Heart rate |
| status | int[] | Status |
| statusValue | int[] | The value of status |
| eHumidity | int[] | Humidity(required equipment support) |
| eTemp | int[] | Temperature(required equipment support) |

## Analysis

### Description

Analysis of sleep report

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| averageBreathRate | int | Average breath rate(n counts per min) |
| averageHeartBeatRate | int | Average heart rate(n counts per min) |
| fallAlseepAllTime | int | Fall asleep time(Unit:min) |
| wakeAndLeaveBedBeforeAllTime | int | Duration of awake before getting up(Unit:min) |
| leaveBedTimes | int | Counts of leaving bed |
| trunOverTimes | int | Counts of turning over |
| bodyMovementTimes | int | Counts of body movement |
| ~~heartBeatPauseTimes~~ | ~~int~~ | ~~Counts of Heat beat Pause(Abandoned)~~ |
| breathPauseTimes | int | Counts of apnea |
| deepSleepPerc | int | Deep sleep percentage |
| inSleepPerc | int | Mid sleep percentage |
| lightSleepPerc | int | Light sleep percentage |
| wakeSleepPerc | int | Awake percentage |
| duration | int | Sleep duration(Unit:min) |
| wakeTimes | int | Awake times |
| lightSleepAllTime | int | Duration of Light sleep(Unit:min) |
| inSleepAllTime | int | Duration of Mid sleep(Unit:min) |
| deepSleepAllTime | int | Duration of Deep sleep)(Unit:min) |
| wakeAllTime | int | Duration of Awake)(Unit:min) |
| breathPauseAllTime | int | Duration of Apnea(Unit:seconds) |
| ~~heartBeatPauseAllTime~~ | ~~int~~ | ~~Duration of heart beat pause)(Unit:seconds)(~~ ~~Abandoned)~~ |
| leaveBedAllTime | int | Duration of leaving bed(Unit:min) |
| maxHeartBeatRate | int | Maximum heart rate(n counts per min) |
| maxBreathRate | int | Maximum breath rate(n counts per min) |
| minHeartBeatRate | int | Minimum heart rate)(n counts per min) |
| minBreathRate | int | Minimum breath rate(n counts per min) |
| heartBeatRateFastAllTime | int | Duration of tachycardia(Unit:seconds) |
| heartBeatRateSlowAllTime | int | Duration of bradycardia(Unit:seconds) |
| breathRateFastAllTime | int | Duration of tachypnea(Unit:seconds |
| breathRateSlowAllTime | int | Duration of bradypnea(Unit:seconds) |
| sleepScore | int | Score  90>=score<=100 Bravo!  80>=score<90 Good!  60>=score<80, average!  score <60 Bad |
| sleepCurveArray | float[] | Example:  [0.212,1.231,2.111,0.212,1.231,2.111,....]  0: awake 0 ~ 1: light sleep 1 ~ 2: moderate sleep 2 ~ 3: deep sleep  Drawing sleep curve (Unit:min) |
| sleepCurveStatusArray | ~~short[]~~ | ~~Sleep Event Flag (Unit:min)~~ |
| breathRateStatusAry | int[] | Apnea, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of Apnea in this minute(Unit:seconds) |
| heartRateStatusAry | int[] | Heart beat pause, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of Heat beat Pause in this minute(Unit:seconds) |
| leftBedStatusAry | int[] | Leave bed, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: Duration of leaving bed in this minute(Unit:seconds) |
| turnOverStatusAry | int[] | Turn over, It used to draw the graph  Example:  [0,0,1,0,2]  0: nothing  Other: the times of turning over |
| sleepState | byte[] | Analytical Sleep Status  0:awake, 1:light Sleep, 2:middle sleep, 3:deep sleep |
| algorithmVer | String | Algorithm version |
| fallsleepTimeStamp | int | The time you fall asleep(timestamp) |
| wakeupTimeStamp | int | The time you wake up(timestamp) |
| reportFlag | int | 1. Long report(>3h) 2. Short report(>10m && <3h) |
| md\_body\_move\_decrease\_scale | short | Score Deduction:Score Deduction due to body movement |
| md\_leave\_bed\_decrease\_scale |  | Score Deduction:Score Deduction due to the times of leaving bed |
| md\_wake\_cnt\_decrease\_scale | short | Score Deduction:Score Deduction due to the wake count |
| md\_start\_time\_decrease\_scale | short | Score Deduction:Score Deduction due to sleeping time (too late) |
| md\_fall\_asleep\_time\_decrease\_scale | short | Score Deduction:Score Deduction due to long falling sleep time |
| md\_perc\_deep\_decrease\_scale | short | Score Deduction:Score Deduction due to the deep sleep |
| md\_sleep\_time\_decrease\_scale | short | Score Deduction due to sleeping time too short |
| md\_sleep\_time\_increase\_scale | short | Score Deduction due to sleeping time too long |
| md\_breath\_stop\_decrease\_scale | short | Score Deduction:Score Deduction due to breathing stop |
| ~~md\_heart\_stop\_decrease\_scale~~ | ~~short~~ | ~~Score Deduction:Score Deduction due to Heart beat stop(Abandoned)~~ |
| md\_heart\_low\_decrease\_scale | short | Score Deduction:Score Deduction due to slow heart beat |
| md\_heart\_high\_decrease\_scale | short | Score Deduction:Score Deduction due to Rapid heart beat |
| md\_breath\_low\_decrease\_scale | short | Score Deduction:Score Deduction due to slow breathing |
| md\_breath\_high\_decrease\_scale | short | Score Deduction:Score Deduction due to rapid breathing |
| md\_perc\_effective\_sleep\_decrease\_scale | short | Score Deduction:Score Deduction due to good sleeping (ratio of middle sleep/deep sleep) |

## SleepStatusType

### Description

Status value of monitoring

### Fields

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Value** | **Description** |
| SLEEP\_OK | byte | 0x00 | normal |
| SLEEP\_INIT | byte | 0x01 | init |
| SLEEP\_B\_STOP | byte | 0x02 | apnea |
| ~~SLEEP\_H\_STOP~~ | ~~byte~~ | ~~0x03~~ | ~~Heartbeat pause(Abandoned)~~ |
| SLEEP\_BODYMOVE | byte | 0x04 | Body movement |
| SLEEP\_LEAVE | byte | 0x05 | Leaving bed |
| SLEEP\_TURN\_OVER | byte | 0x06 | Turning over |
| SLEEP\_BODYMOVE\_TEMP | byte | 0x07 | Amplitude of body motion |
| SLEEP\_INVALID | byte | -1 | invalid |

## EnvironmentData

### Description

The result of getting environment data

### Fields

|  |  |  |
| --- | --- | --- |
| **Field** | **Type** | **Description** |
| temperature | int | Temperature,Unit(Celsius) |
| humidity | int | humidity |

## OnlineStateListener

### Interface Introduction

Device online status monitoring

### Method description

**void** onlineStateChanged(**byte** onlineState);

onlineState：0: offline 1: online 0xFF: unknown state

## SleepReportUploadStateListener

### Interface Introduction

Sleep report upload status monitoring

### Method description

**void** onStateChanged(**byte** state);

state：

0: No data (no need to wait any longer)

1: ready

2: syncing

3: uploading

4: Upload failed (no need to wait any longer)

5: Successful upload

6: End of upload

note:

0, subsequent content is invalid

1, indicating that the device has started the upload process, the subsequent content is invalid

2. It indicates that the device starts to synchronously monitor the device data process, and the subsequent content is invalid

Process description: The entire upload task starts in the preparation state (1) and ends in the end state (6)

## RealtimeDataListener

### Interface Introduction

Real-time data monitoring

### Method description

**void** onReceive([RealTimeData](#_RealTimeData) realTimeData);

realTimeData：Real-time data reported by the device, including heart rate and breath rate, etc. refer to：[RealTimeData](#_RealTimeData_1)

## RealtimeSleepStateListener

### Interface Introduction

Real-time sleep state monitoring

### Method description

**void** onReceive([SleepState](#_SleepState) sleepState);

sleepState：The real-time sleep status reported by the device refer to：[SleepState](#_SleepState)