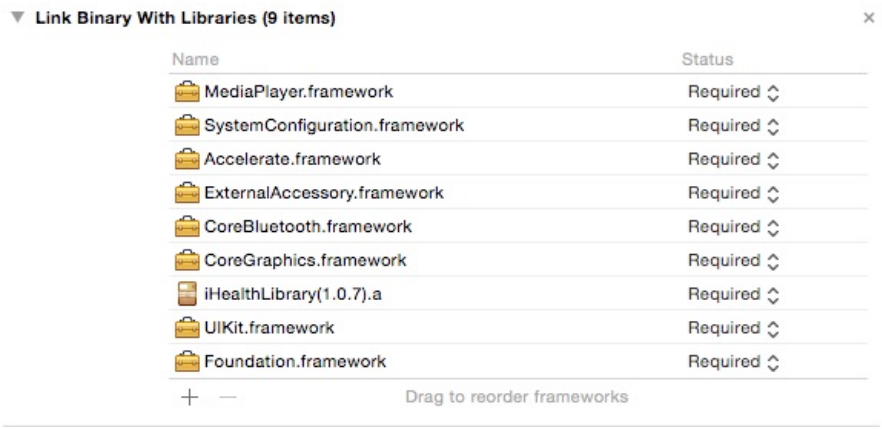
# AM SDK 接口文档

## 1、 需要引入的文件，Frameworks 及相关设置

（1） 需要引入开发工具 AMSDK，工具包包含文件如下：AMHeader.h、AMMacroFile.h、AM3.h、AM3Controller.h、AM3S.h、AM3SController.h、AM4.h、AM4Controller.h、User.h 、iHealthLibrary(x.x.x).a，支持 IOS6.0 及其以上版本。

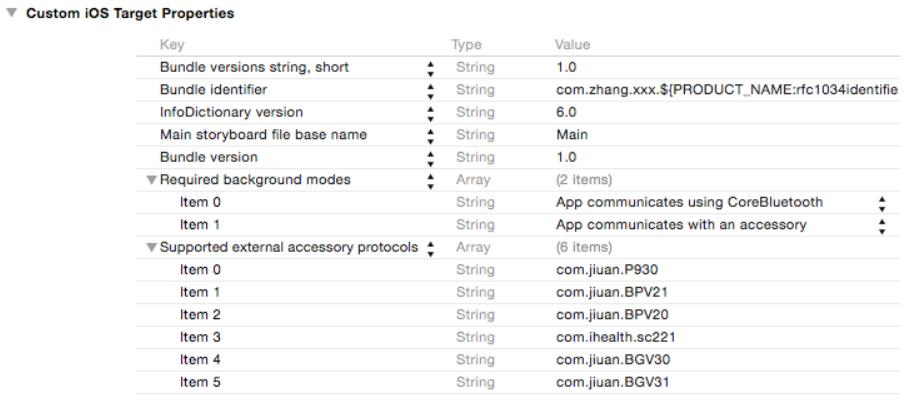
（2） Frameworks




（3） 工程设置

在 Info 中增加项

Required background modes，支持后台附件通讯模式： App communicates with an accessory 、App communicates using CoreBluetooth




## 2、 使用流程

### (1)AM3使用流程

注册 AM3 设备接入消息：AM3ConnectNoti；

初始化 AM3 控制类：

```
AM3Controller *controller = [AM3Controller shareIHAM3Controller];
```

接收到 AM3ConnectNoti后，获取控制类实例：

```
NSArray *amDeviceArray = [controller getAllCurrentAM3Instace];
AM3 *amInstance = [amDeviceArray objectAtIndex:i];
```

用amInstance这个对象就可以调用AM3相关通讯方法。

注：

若已经连接上自己的AM设备，可以停止连接其他AM设备，使用Api:

[controller commandCanConnectAM:],传入True可以连接， False禁止连接。

## (2) AM3S使用流程

注册 AM3S 设备接入消息：AM3SConnectNoti；

初始化 AM3S 控制类：

```
AM3SController *controller = [AM3SController shareIHAM3SController];
```

接收到 AM3SConnectNoti后，获取控制类实例：

```
NSArray *amDeviceArray = [controller getAllCurrentAM3SInstace];
AM3S *amInstance = [amDeviceArray objectAtIndex:i];
```

用amInstance这个对象就可以调用AM3S相关通讯方法。

注：

a. 可以指定连接特定设备，提高连接效率，使用Api:

[controller commandSetYourDeviceID:],传入指定AM设备的唯一ID（mac地址）即可。

b. 若需要连接其他所有AM设备，使用Api:

[controller commandCanConnectOtherDevice:], 传入True可以连接，False禁止连接。

## (3) AM4使用流程

注册 AM4 设备接入消息：AM4ConnectNoti；

初始化 AM4 控制类：

```
AM4Controller *controller = [AM4Controller shareIHAM4Controller];
```

接收到 AM4ConnectNoti后，获取控制类实例：

```
NSArray *amDeviceArray = [controller getAllCurrentAM4Instace];
AM4 *amInstance = [amDeviceArray objectAtIndex:i];
```

用amInstance这个对象就可以调用AM4相关通讯方法。

注：

c. 可以指定连接特定设备，提高连接效率，使用Api:

[controller commandSetYourDeviceID:],传入指定AM设备的唯一ID（mac地址）即可。

d. 若需要连接其他所有AM设备，使用Api:
[controller commandCanConnectOtherDevice:], 传入True可以连接，False禁止连接。

# 3、AM3 接口方法说明

## （1）创建用户管理连接

此接口验证通过后，才可以使用其他 Api。

```
-(void)commandCreateUserManageConnectWithUser:(User
*)tempUser
Authentication:(BlockUserAuthentication)disposeAuthenticationB
lock currentUserSerialNub:(CurrentSerialNub)serialNub
amUser:(DisposeAM3AskUserID)disposeAskUserID
binedAMSerialNub:(DisposeBinedAMSerialNub)binedSerialnub
currentSerialNub:(DisposeCurrentSerialNub)currentSerialNub
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

传入参数：

tempUser，需包含属性：clientID，clientSecret，userID。

userID，用户的唯一标示，格式为邮箱或手机号，目前支持邮箱格式。

clientID 和 clientSecret，为 sdk 应用唯一标示，通过注册 iHealth SDK 应用获取。申请注册 SDK 应用请联系：lvjincan@ihealthlabs.com.cn.

返回参数：

disposeAuthenticationBlock，对 userid、clientID、clientSecret 进行验证后的返回结果。

验证结果解释：

UserAuthen_RegisterSuccess,新用户注册成功

UserAuthen_LoginSuccess,用户登录成功

UserAuthen_CombinedSuccess,用户为 iHealth 用户，增加了 SDK 测量功能，通过 SDK 测量产生的测量数据也属于此用户。

UserAuthen_TrySuccess,网络异常仅测试使用

UserAuthen_InvalidateUserInfo, userid 或 clientID 或 clientSecret 验证失败

UserAuthen_SDKInvalidateRight,应用无此权限

UserAuthen_UserInvalidateRight,用户无此权限

UserAuthen_InternetError，网络异常导致验证失败

前 3 种情况可以继续使用 SDK 的测量功能，后 5 种情况会终止测量，需要解决反馈的问题后再调用此接口功能。

注意：首次使用 SDK 进行新用户注册时，会自动弹出”iHealth 免责声明”，需要用

户同意后才能继续使用。且首次使用需要保证网络通畅。

serialNub:用户唯一标示，SDK 调用者需要记录，该标示会写入到 AM 设备中，做为设备被用户绑定的唯一标示。

disposeAskUserID：AM 设备中存储的用户唯一标示，0 标示该设备未被用户绑定.

binedSerialnub：用户绑定的 AM 设备的唯一标示（mac 地址）。

currentSerialNub: 当前连接设备的唯一标示（mac 地址）.

disposeErrorBlock:通讯过程中的错误代码，见第 6 部分 AM3 错误说明。

### （2）绑定用户

查询到 AM 设备中存储的用户唯一标示为 0 时使用.

注：绑定用户时需要网络连接通畅，否则无法绑定成功

```
-(void)commandsetAM3UserID:(NSNumber*)userID
DisposeBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

输入参数：

userID，用户唯一标示。

返回参数：

disposeBlock ：绑定成功返回 YES,失败返回 NO.

disposeErrorBlock: 见第 6 部分 AM3 错误说明。

### （3）初始化 AM

首次使用必须调用，保证 AM 中存储的用户信息完整，设置 AM 用户信息、运动目标、时间，查询电池电量等。

```
-(void)commandSyncUserInfoWithUser:(User *)tempUser
andGoal:(NSNumber*)goalNumber
DisposeStateInfo:(DisposeAM3StateInfo)disposeStateInfo
DisposeBattery:(DisposeAM3Battery)disposeBattery
DisposeBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

输入参数：

tempUser，用户信息，需要包含属性：birthday、height、weight、bmr、sex、lengthUnit。

birthday，用户生日，NSDate.

height，用户身高，单位（cm）。

weight，用户体重，单位（kg）。

bmr，用户基础代谢率。

sex，用户性别，UserSex_Female 女，UserSex_Male 男。

lengthUnit，AM 显示总距离单位，LengthUnit_Mile 英里，LengthUnit_Kilometer 千米。

goalNumber，用户目标步数，默认为 10000.

返回参数：

disposeStateInfo：AM 状态，State_wrist 手腕，State_waist 腰。

disposeBattery：AM 电池电量，百分比，范围 0～100.

disposeBlock：初始化完成，True 成功， False 失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明。

（4）恢复出厂设置

-(void)commandResetDeviceDisposeResultBlock:(DisposeAM3Block) disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

返回参数:

disposeBlock: True 成功， False 失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

（5）查询 AM 闹钟

-(void)commandQueryAlarmInfo:(DisposeAM3TotoalAlarmData)disposeTotoalAlarmData
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

返回参数:

disposeTotoalAlarmData ：闹钟数组，最多包含 3 个闹钟，每个闹钟包含以下属性：AlarmId、Time、IsRepeat、Switch、（Sun、Mon、Tue、Wed、Thu、Fri、Sat。

AlarmId：闹钟 id，范围：1、2、3.

Time：闹钟时间，格式 HH:mm。

IsRepeat：是否重复， True 重复， False 不重复。

Switch：闹钟开关，True 打开， False 关闭。

Sun、Mon、Tue、Wed、Thu、Fri、Sat：需要重复的置为 True.

disposeErrorBlock: 见第 6 部分 AM3 错误说明

（6）设置闹钟

-(void)commandSetAlarmWithAlarmDictionary:(NSDictionary *)alarmDic DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock

输入参数:

alarmDic：闹钟信息，包含属性：AlarmId、Time、IsRepeat、Switch、（Sun、Mon、Tue、Wed、Thu、Fri、Sat，参数说明见查询闹钟 Api.

返回参数:

disposeBlock: True 设置成功，False 设置失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

（7）删除闹钟

-(void)commandDeleteAlarmViaID:(NSNumber *)alarmID
DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;

输入参数:

alarmID：闹钟 id，范围：1、2、3.

返回参数：

disposeBlock: True 删除成功，False 删除失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

### （8）查询提醒

```
-(void)commandQueryReminder:(RemindAM3Info)remindInfo
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

返回参数：

remindInfo：提醒数组，包含属性：Time、Switch。

Time：提醒时长，格式 HH:mm，即提醒间隔为(HH*60+mm)分钟。

Switch：提醒开关，True 打开， False 关闭。

disposeErrorBlock: 见第 6 部分 AM3 错误说

### （9）设置提醒

```
-(void)commandSetReminderwithReminderDictionary:(NSDictionary
*)reminderDic DisposeResultBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

输入参数：

reminderDic：提醒信息，包含属性：Time、Switch。

返回参数：

disposeBlock: True 设置成功，False 设置失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

### （10）上传 AM 数据

数据类型：5分钟运动数据、5分钟睡眠数据、当天总步数和总卡路里。其中，5分钟运动数据中的步数、卡路里为当前时间总步数和总卡路里，如果需要计算每5分钟的运动数据，需要用两个记录做差计算获取。

```
-(void)commandSyncAllAMData:(StartAM3Transmission)startAM3Transm
issio  DisposeProgress:(DisposeAM3ProgressData)disposeAM3ProgressData
historyData:(AM3HistoryData)AM3historyData
FinishTransmission:(FinishAM3Transmission)finishAM3Transmission
startsleepdata:(StartSleepTransmission) startSleepTransmission
DisposeSleepProgress:(DisposeSleepProgressData)disposeSleepProgressD
ata sleephistoryData:(SleepHistoryData)sleepHistoryData
FinishSleepTransmission:(FinishSleepTransmission)finishSleepTransmis
sion CurrentActiveInfo:(DisposeAM3QueryCurrentActiveInfo)
disposeQueryCurrentActiveInfo
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock
AM3IsOnTransmission:(AM3IsOnTransmission)am3isOnTransmission
SleepIsOnTransmission:(SleepIsOnTransmission)sleepisOnTransmission;
```

返回参数：

startAM3Transmissio:开始上传运动数据，包含属性：StartActiveHistoryDate、StepSize、StartActiveHistoryTotoalNum。

StartActiveHistoryDate：运动开始时间，yyyy-MM-dd.

StepSize：步长，单位 cm。

StartActiveHistoryTotoalNum：记录总条数。

disposeAM3ProgressData：上传运动数据百分比，范围 0.0～1.0.

AM3historyData：运动记录，包含属性：AMDate、AMCalorie、AMStepNum。

AMDate：运动时间， NSDate.

AMCalorie:当前时间总卡路里。

AMStepNum：当前时间总步数。

finishAM3Transmission：运动上传完成。

startSleepTransmission：开始上传睡眠，包含属性：SleepActiveHistoryDate、StartActiveHistoryTotoalNum。

SleepActiveHistoryDate：睡眠开始时间，yyyy-MM-dd HH:mm:ss.

StartActiveHistoryTotoalNum: 记录总条数.

disposeSleepProgressData: 上传睡眠数据百分比，范围 0.0～1.0。

sleepHistoryData：睡眠记录，包含属性：AMDate、SleepData。

AMDate：睡眠时间， NSDate.

SleepData:睡眠等级，0 清醒， 1 浅睡， 2 深睡。

finishSleepTransmission：睡眠上传完成。

disposeQueryCurrentActiveInfo：当天最新总卡路里和总步数，包含属性：Step、Calories。

Step：当前总步数。

Calories：当前总卡路里。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

am3isOnTransmission： Invalidate.

sleepisOnTransmission: Invalidate.

### （11）设置 AM 状态

```
-(void)commandSetState:(ActiveState)activeState
DisposeBlock:(DisposeAM3Block)disposeBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

输入参数:

activeState：Sleep_State 睡眠，Active_State 运动，Fly_State 飞行， Drive_State 驾驶。

其中，Fly_State、Drive_State 对 App 有效。

返回参数：

disposeBlock: True 设置成功，False 设置失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

### （12）断开 AM 连接

```
-(void)commandDisconnectDisposeBlock:(DisposeAM3Block)disp
oseBlock
DisposeErrorBlock:(DisposeAM3ErrorBlock)disposeErrorBlock;
```

disposeBlock: True 设置成功，False 设置失败。

disposeErrorBlock: 见第 6 部分 AM3 错误说明

# 4、AM3S 接口方法说明

AM3S 的部分 Api 与 AM3 功能一致，本部分仅介绍不一致及新功能 Api.

（1）发送随机数

此 Api 将发送随机数给 AM3S 设备，只有当随机数匹配成功后，用户才可与该设备绑定。

```
-(void)commandSetRandomString:(DisposeRandomNumberSetting)disposeBlock
DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;
```

返回参数：

disposeBlock, True 发送成功，False 发送失败。随机数为 6 位数字，范围 0～999999，AM3S 接收到该随机数之后，会显示在屏幕上，用户可读取后进行匹配。

disposeErrorBlock: 见第 6 部分 AM3S 错误说明。

（2）绑定用户

注：绑定用户时需要网络连接通畅，否则无法绑定成功。

```
-(void)commandSetAM3SUserID:(NSNumber*)userID
withRandom:(NSString *)tempRandom
DisposeBlock:(DisposeAM3SBlock)disposeBlock
DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;
```

输入参数：

userID，用户唯一标示。

tempRandom，AM 设备上显示的 6 位随机数。

返回参数：

disposeBlock ：绑定成功返回 YES,失败返回 NO.

disposeErrorBlock: 见第 6 部分 AM3S 错误说明。

（3）设置 AM3S 状态

```
-(void)commandSetState:(AM3SActiveState)activeState
DisposeBlock:(DisposeAM3SBlock)disposeBlock
DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;
```

输入参数：

activeState，仅支持 AM3SFly_State（飞行模式）。

返回参数：

disposeBlock ：设置成功返回 YES,失败返回 NO.

disposeErrorBlock: 见第 6 部分 AM3S 错误说明。

（4）上传 AM3S 报表数据

```
-(void)commandSetSyncsportCount:(DisposeSyncSportCount)disposeSyncSportCount
DisposeMeasureData:(DisposeMeasureData)disposeMeasureData
disposeFinishMeasure:(DisposeFinishMeasure)disposeFinishMeasure
```

DisposeErrorBlock:(DisposeAM3SErrorBlock)disposeErrorBlock;

返回参数：

disposeSyncSportCount：总的报表条数。

disposeMeasureData：报表数据，包含 ReportStage_Swimming、ReportStage_Work_out、ReportStage_Sleep_summary、ReportStage_Activeminute,目前仅支持 ReportStage_Work_out、ReportStage_Sleep_summar。

Workout 包含属性：ReportState、Work_outMeasureDate、Work_outTimeNumber、Work_outStepNumber、Work_outLengthNumber、Work_outCalories。

ReportState：ReportStage_Work_out。

Work_outMeasureDate：开始时间。

Work_outTimeNumber：运动时长，单位分钟。

Work_outStepNumber：运动总步数。

Work_outLengthNumber：运动距离，单位 km。

Work_outCalories：运动卡路里，单位 kal。

Sleep 包含属性：ReportState、Sleep_summaryMeasureDate、Sleep_summarySleepTime、Sleep_summarysleepEfficiency、Sleep_summarysleepAddMinute。

ReportState：ReportStage_Sleep_summary。

Sleep_summaryMeasureDate：睡眠开始时间.

Sleep_summarySleepTime:睡眠时长，单位分钟。

Sleep_summarysleepEfficiency：睡眠效率，百分比，范围 0.0～100.0.

Sleep_summarysleepAddMinute：数据修正时长，将睡眠开始时间之前的特定时长数据修改为睡眠清醒数据。

# 5、AM4 接口方法说明

AM4 的部分 Api 与 AM3 功能一致，本部分仅介绍不一致及新功能 Api.

（1）发送随机数

此 Api 将发送随机数给 AM4 设备，只有当随机数匹配成功后，用户才可与该设备绑定。

-(void)commandAM4SetRandomBlock:(DisposeAM4SetRandomBlock)disposeSetRandom
disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;

返回参数：

disposeSetRandom,True 发送成功，False 发送失败。随机数为 6 位数字，范围 0～999999，AM4 接收到该随机数之后，会显示在屏幕上，用户可读取后进行匹配。

disposeErrorBlock: 见第 7 部分 AM4 错误说明。

（2）绑定用户

注：绑定用户时需要网络连接通畅，否则无法绑定成功。

```
-(void)commandSetAM4UserID:(NSNumber*)userID
withRandom:(NSString *)tempRandom
DisposeBlock:(DisposeAM4SetUserIDBlock)disposeBlock
DisposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;
```

输入参数：

userID，用户唯一标示。

tempRandom，AM 设备上显示的 6 位随机数。

返回参数：

disposeBlock ：绑定成功返回 YES,失败返回 NO.

disposeErrorBlock: 见第 7 部分 AM4 错误说明。

（3）设置 AM4 状态

```
-(void)commandAM4SetState:(AM4ActiveState)activeState
disposeBlock:(DisposeAM4SetStateBlock)disposeSetState
disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;
```

输入参数：

activeState，仅支持 AM4Active_State（飞行模式）。

返回参数：

disposeSetState：设置成功返回 YES,失败返回 NO.

disposeErrorBlock: 见第 7 部分 AM4 错误说明。

(4)设置游泳

```
-(void)commandAM4SetSwimmingState:(BOOL)swimmingIsOpen
swimmingPoolLength:(NSNumber *)swimmingPoolLength
NOSwimmingTime:(NSDate *)noSwimmingDate
unit:(AM4SwimmingUnit)unit
resultBlock:(DisposeAM4SettingSwimmingBlock)disposeSetSwimmin
g disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;
```

输入参数：

swimmingIsOpen：yes,打开游泳功能，no，关闭游泳功能，默认关闭

swimmingPoolLength：泳池长度

noSwimmingDate：自动切出游泳时间

unit：单位

返回参数：

disposeSetSwimming：设置成功返回YES,失败返回 NO.

disposeErrorBlock：见第7部分AM4错误说明。

(5)上传 AM4 报表数据

```
-(void)commandAM4SetSyncsportCount:(DisposeAM4SyncSportCountB
lock)disposeSyncSportCount
disposeMeasureData:(DisposeAM4MeasureDataBlock)disposeMeasureData
disposeFinishMeasure:(DisposeAM4WorkoutFinishBlock)disposeFinishM
easure disposeErrorBlock:(DisposeAM4ErrorBlock)disposeErrorBlock;
```

返回参数：

disposeSyncSportCount：总的报表条数。

disposeMeasureData：报表数据，包含 ReportStage_Swimming、ReportStage_Work_out、ReportStage_Sleep_summary、ReportStage_Activeminute,目前仅支持 ReportStage_Work_out、ReportStage_Sleep_summar,ReportStage_Swimming。

Workout 包含属性：AM4ReportState、AM4Work_outMeasureDate、AM4Work_outTimeNumber、AM4Work_outStepNumber、AM4Work_outLengthNumber、AM4Work_outCalories。

AM4ReportState：AM4ReportStage_Work_out。

AM4Work_outMeasureDate：开始时间。

AM4Work_outTimeNumber：运动时长，单位分钟。

AM4Work_outStepNumber：运动总步数。

AM4Work_outLengthNumber：运动距离，单位 km。

AM4Work_outCalories：运动卡路里，单位 kal。

Sleep 包含属性：AM4ReportState、AM4Sleep_summaryMeasureDate、AM4Sleep_summarySleepTime、AM4Sleep_summarysleepEfficiency、AM4Sleep_summarysleepAddMinute。

AM4ReportState：AM4ReportStage_Sleep_summary。

AM4Sleep_summaryMeasureDate：睡眠开始时间.

AM4Sleep_summarySleepTime:睡眠时长，单位分钟。

AM4Sleep_summarysleepEfficiency：睡眠效率，百分比，范围 0.0～100.0.

AM4Sleep_summarysleepAddMinute：数据修正时长，将睡眠开始时间之前的特定时长数据修改为睡眠清醒数据。

Swimming 包含属性：AM4ReportState、AM4SwimmingMeasureDate、AM4SwimmingTimeNumber、AM4SwimmingTimes、AM4Swimmingcalories、AM4SwimmingAct、AM4SwimmingCircleCount、AM4SwimmingPoollength。

AM4ReportState：AM4ReportStage_Swimming

AM4SwimmingMeasureDate：结束时间

AM4SwimmingTimeNumber：游泳耗时

AM4SwimmingTimes：划水次数

AM4Swimmingcalories：卡路里

AM4SwimmingAct：泳姿

AM4SwimmingCircleCount：圈数

AM4SwimmingPoollength：泳池长度

## 6．相关参数补充说明

设备连接消息：AM3ConnectNoti、AM3SConnectNoti、AM4ConnectNoti

设备断开消息：AM3DisConnectNoti、AM3SDisConnectNoti、AM4DisConnectNoti

连接多个设备时，通过 serialNumber 属性区别多个设备。

## 7．错误代码说明

(1)AM3:

```
typedef enum{
AMErrorOverTime = 0,
AM_Reset_Device_Faild,
AMErrorDisconnect,
AMErrorUserInvalidate
}AMErrorID;
```

(2)AM3S:

```
typedef enum{
AM3SErrorOverTime = 0,
AM3S_Reset_Device_Faild,
AM3SErrorDisconnect,
AM3SErrorUserInvalidate
}AM3SErrorID;
```

(3)AM4:

```
typedef enum{
AM4ErrorOverTime = 0,
AM4_Reset_Device_Faild,
AM4ErrorDisconnect,
AM4ErrorUserInvalidate
}AM4ErrorID;
```

## 7. 测试 demo