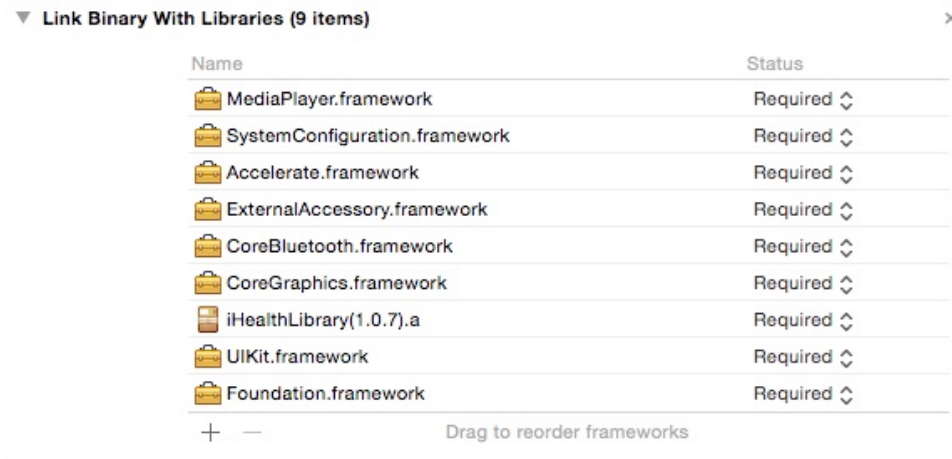


HS SDK 接口文档

1、 需要引入的文件，Frameworks 及相关设置

(1) 需要引入开发工具包 HSSDK，工具包包含文件如下：HSHeader.h、HSMacroFile.h、HS3.h、HS3Controller.h、HS4.h、HS4Controller.h、HS5.h、HS5Controller.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

Supported external accessory protocols,并添加支持协议：com.ihealth.sc221

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

| Custom iOS Target Properties | | | |
|--|--------|--|--|
| Key | Type | Value | |
| Bundle versions string, short | String | 1.0 | |
| Bundle identifier | String | com.zhang.xxx.\${PRODUCT_NAME:rfc1034identifie | |
| InfoDictionary version | String | 6.0 | |
| Main storyboard file base name | String | Main | |
| Bundle version | String | 1.0 | |
| Required background modes | Array | (2 items) | |
| Item 0 | String | App communicates using CoreBluetooth | |
| Item 1 | String | App communicates with an accessory | |
| Supported external accessory protocols | Array | (6 items) | |
| Item 0 | String | com.jiuan.P930 | |
| Item 1 | String | com.jiuan.BPV21 | |
| Item 2 | String | com.jiuan.BPV20 | |
| Item 3 | String | com.ihealth.sc221 | |
| Item 4 | String | com.jiuan.BGV30 | |
| Item 5 | String | com.jiuan.BGV31 | |

2、 使用流程

(1) HS3使用流程

注册 HS3 设备接入消息: `HS3ConnectNoti`;

初始化 HS3 控制类:

```
HS3Controller *controller = [HS3Controller  
shareIHHs3Controller];
```

接收到 `HS3ConnectNoti`后, 获取控制类实例:

```
NSArray *hsDeviceArray = [controller  
getAllCurrentHS3Instace];
```

```
HS3 *hsInstance = [hsDeviceArray objectAtIndex:i];
```

用`hsInstance`这个对象就可以调用HS3相关通讯方法。

(2) HS4使用流程

注册 HS4 设备接入消息: `HS4ConnectNoti`;

初始化 HS4 控制类:

```
HS4Controller *controller = [HS4Controller  
shareIHHs4Controller];
```

接收到 `HS4ConnectNoti` 后, 获取控制类实例:

```
NSArray *hsDeviceArray = [controller  
getAllCurrentHS4Instace];
```

```
HS4 *hsInstance = [hsDeviceArray objectAtIndex:i];
```

用`hsInstance`这个对象就可以调用HS4相关通讯方法。

(3) HS5使用流程

注册 HS5 设备接入消息: `HS5ConnectNoti`;

初始化 HS5 控制类:

```
HS5Controller *controller = [HS5Controller  
shareIHHs5Controller];
```

接收到 `HS5ConnectNoti` 后, 获取控制类实例:

```
NSArray *hsDeviceArray = [controller  
getAllCurrentHS5Instace];
```

```
HS5 *hsInstance = [hsDeviceArray objectAtIndex:i];
```

用`hsInstance`这个对象就可以调用HS5相关通讯方法。

3、HS3 接口方法说明

(1) 创建记忆和测量连接

`-(void)commandInitWithUser:(User *)tempUser`

`Authentication:(BlockUserAuthentication)disposeAuthenticationBlock`

`TransferMemorryData:(StartHS3Transmission)startTransmission`

`UploadDataNum:(DisposeHS3UploadDataNum)hs3UploadDataNum`

`DisposeProgress:(DisposeHS3Progress)progress`

MemoryData: (HS3MemoryData)memoryData
FinishTransmission: (FinishHS3Transmission)finishTransmission
StableWeight: (StableHS3Weight)stableWeight
DisposeErrorBlock: (DisposeHS3ErrorBlock)disposeHS3ErrorBlock;传入参数:

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

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。

height, 用户身高, 单位: cm。

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、height、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 4 种情况可以继续使用 SDK 的测量功能, 后 4 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。另外, 若使用 SDK 时无网络连接, 可以有 10 天的试用期 (从首次使用 SDK 开始记时), 10 天之后如果仍没有接入网络认证则无法继续使用。

startTransmission: 开始记忆传输, 成功 Yes, 失败 No。

hs3UploadDataNum: 记忆总条数, 0~200。

progress: 记忆传输进度, 0.0~1.0。

memoryData: 记忆内容, 包括体重 (kg)、测量时间, 相应 key: weight、date。

finishTransmission: 记忆传输结束。

stableWeight: 稳定体重, 单位 Kg。

disposeHS3ErrorBlock: 测量过程中的错误代码。

错误代码解释:

见第 6 部分 HS3 错误说明。

(2) 关闭蓝牙回连

该方法仅适用于 hsInstance.firmwareVersion>=1.0.2 的 HS3。

-(void)commandTurnOffBTConnectAutoResult:(DisposeResult)dispo

seResult
DisposeErrorBlock:(DisposeHS3ErrorBlock)disposeHS3ErrorBlock;

返回参数:

disposeResult: 成功返回 YES, 失败返回 NO.

disposeHS3ErrorBlock: 见第 6 部分 HS3 错误说明。

(3) 打开蓝牙回连

该方法仅适用于 hsInstance.firmwareVersion>=1.0.2 的 HS3.

-(void)commandTurnOnBTConnectAutoResult:(DisposeResult)disposeResult
DisposeErrorBlock:(DisposeHS3ErrorBlock)disposeHS3ErrorBlock;

返回参数:

disposeResult: 成功返回 YES, 失败返回 NO.

disposeHS3ErrorBlock: 见第 6 部分 HS3 错误说明。

4、HS4 接口方法说明

(1) 创建测量连接

sdk 首次使用时, 需要先调用测量方法, 完成用户验证。

-(void)commandMeasureWithUnit:(HSUnit)tempUnit
andUser:(User *)tempUser
Authentication:(BlockUserAuthentication)disposeAuthenticationBlock
Weight:(UnStableWeight)unStableWeight
StableWeight:(StableWeight)stableWeight
DisposeErrorBlock:(DisposeHS4ErrorBlock)disposeErrorBlock;

传入参数:

tempUnit, HS4 显示单位, HSUnit_Kg、HSUnit_LB、HSUnit_ST。

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

userID、clientID、clientSecret、disposeAuthenticationBlock,

见 HS3 说明。

unStableWeight: 时时体重, 单位 kg。

stableWeight: 稳定体重, 单位 kg。

disposeErrorBlock: 测量过程中的错误代码。

错误代码解释:

见第 6 部分 HS4 错误说明。

(2) 上传记忆数据

若用户未通过验证, 调用此方法会返回: HS4UserInvalidate, 用户信息无效。

-(void)commandTransferMemoryWithUser:(User *)tempUser
memoryData:(StartHS4Transmission)startTransmission
DisposeProgress:(DisposeProgress)progress
MemoryData:(MemoryData)memoryData
FinishTransmission:(FinishHS4Transmission)finishTransmission

`DisposeErrorBlock:(DisposeHS4ErrorBlock)disposeErrorBlock;`

传入参数:

`tempUser`, 需包含属性: `userID`, 见 HS3 说明。

`startTransmission`: 开始记忆传输, 成功 Yes, 失败 No.

`progress`: 记忆传输进度, 0.0~1.0.

`memoryData`: 记忆内容, 包括体重 (kg)、测量时间, 相应 key: `weight`、`date`。

`finishTransmission`: 记忆传输结束。

`disposeErrorBlock`: 记忆上传过程中的错误代码。

错误代码解释:

见第 6 部分 HS4 错误说明。

5、HS5 接口方法说明

(1) 创建用户管理连接

`-(void)commandCreateUserManageConnectWithUser:(User`

`*)tempUser`

`Authentication:(BlockUserAuthentication)disposeAuthenticationB`

`lock currentUserSerialNub:(CurrentSerialNub)serialNub`

`deviceUserList:(MemoryUserListHS5Data) MemoryUserListHS5Data`

`Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock;`

传入参数:

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

`userID`、`clientID`、`clientSecret`、`disposeAuthenticationBlock`,

见 HS3 说明。

返回参数:

`serialNub`: 用户的唯一编号, 第三方应用需要保留, 提供给其他 Api 使用。

`MemoryUserListHS5Data`: HS5 中已有用户的信息, 包括用户的

`serialNub`、`Position` (位置)。对应的 key: `serialNumber`、`position`。

`disposeErrorBlock`: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(2) 创建新用户

若当前用户的 `SerialNub` 不在 HS5 的用户列表中, 且 HS5 的用户数量小于 20 时使用。

`-(void)commandCreateUser:(User *)tempUser`

`position:(uint8_t)tempPosition`

`DisposeHS5Result:(DisposeHS5Result)disposeHS5Result`

`Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock;`

传入参数:

`tempUser`, 需包含属性: `serialNub`、`birthday`、`height`、`isAthlete`、`sex`。

`serialNub`: 用户唯一编号。

`birthday`: 用户生日, `NSDate`。

`height`: 用户身高, 单位 cm。

`isAthlete`: 用户是否是运动员, `UserIsAthelete_No` 不是,

UserIsAthelete_Yes 是。

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

tempPosition: 用户在 HS5 中的位置, 范围: 0~19, 从未被其他用户使用的空余位置中选取。

返回参数:

disposeHS5Result: Yes 成功, No 失败。

disposeErrorBlock: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(3) 修改用户信息

若当前用户的 SerialNub 已经在 HS5 的用户列表中, 用户信息发生改变时使用。

-(void)commandModifyUser:(User *)tempUser

DisposeHS5Result:(DisposeHS5Result)disposeHS5Result

Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock;

传入参数:

tempUser, 需包含属性: serialNub、birthday、height、isAthlete、sex, 说明见创建新用户 Api。

返回参数:

disposeHS5Result: Yes 成功, No 失败。

disposeErrorBlock: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(4) 删除指定用户

若需要从 HS5 中删除用户时使用。

-(void)commandDelteUser:(User *)tempUser

DisposeHS5Result:(DisposeHS5Result)disposeHS5Result

Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock
;

传入参数:

tempUser, 需包含属性: serialNub, 说明见创建新用户 Api。

返回参数:

disposeHS5Result: Yes 成功, No 失败。

disposeErrorBlock: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(5) 创建测量连接

-(void)commandCreateMeasureWithUser:(User *)tempUser

unStableWeight:(UnStableHS5Weight)unStableHS5Weight

MeasureWeight:(StableHS5Weight)stableHS5Weight

ImpedanceType:(ImpedanceWeight)impedanceWeight

BodyCompositionMeasurements:(BodyCompositionMeasurements)bodyCompositionMeasurements

Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock;

传入参数:

tempUser, 需包含属性: serialNub、height, 说明见创建新用户 Api。

返回参数:

unStableHS5Weight: 时时体重, 单位 kg。

stableHS5Weight: 稳定体重, 单位 kg。

impedanceWeight: 阻抗体重, 单位 kg。

bodyCompositionMeasurements: 人体成分信息, 包括体重(kg)、

脂肪含量(%）、水分含量(%）、肌肉含量(kg)、骨骼含量(kg)、内脏脂肪等级、DCI(Kcal)。对应的 key: **weight**、**weightFatValue**、**waterValue**、**muscleValue**、**skeletonValue**、**VFatLevelValue**、**DCIValue**。

disposeHS5Result: Yes 成功, No 失败。

disposeErrorBlock: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(6) 创建记忆上传连接

```
-(void)commandCreateMemoryWithUser:(User *)tempUser
uploadConnect: (DisposeHS5Result)disposeHS5Result
TransferMemoryData:(StartHS5Transmission)startTransmission
DisposeProgress:(DisposeHS5Progress)progress
MemoryData:(MemoryHS5Data)memoryData
FinishTransmission:(FinishHS5Transmission)finishTransmission
Disposehs5ErrorBlock:(DisposeHS5ErrorBlock)disposeErrorBlock;
```

传入参数:

tempUser, 需包含属性: **serialNub**, 说明见创建新用户 Api。

返回参数:

disposeHS5Result: Yes 创建记忆连接成功, No 失败。

startTransmission: 开始记忆传输。

progress: 记忆传输进度, 0.0~1.0。

memoryData: 记忆内容, 详细信息及 key 见测量 Api。另外增加测量时间属性, 对应 key: **date**。

finishTransmission: 记忆传输结束。

disposeErrorBlock: 通讯过程中的错误吗, 见第 6 部分 HS5 错误说明。

(7) 从 cloud 获取 HS5 数据

HS5 在连接网络时, 可以将测量结果直接上传到 cloud 保存, 通过下面的 Api 可以获取到 cloud 的数据。

```
[[HS5Controller shareIHHS5Controller]
getDownloadDataFromCloud:];
-(void)getDownloadDataFromCloud:(BlockDataFromCloud )dataBlock;
```

返回参数:

dataBlock: 从 cloud 获取的测量结果集合, 具体 key 见测量 Api。

5. 相关参数补充说明

设备连接消息: **HS3ConnectNoti**、**HS4ConnectNoti**、**HS5ConnectNoti**

设备断开消息: **HS3DisconnectNoti**、**HS4DisconnectNoti**、**HS5DisconnectNoti**

连接多个设备时, 通过 **deviceID** 属性区别多个设备。

6. 错误代码说明

(1)HS3:

```
typedef enum{
```

```

        HS3DeviceLowPower = 1, //Low battery
        HS3DeviceEr2, //Weight capacity is exceeded
        HS3DeviceEr4, //The Scale calibration error
        HS3DeviceEr7, //Movement while measuring
            HS3DataZeor, //No memory
        HS3DeviceDisconnect, //Device disconnect
        HS3DeviceSendTimeout, //Communication error
        HS3UserInvalidate = 111//User verify error
    }HS3DeviceError;

```

(2)HS4:

```

        typedef enum{
        HS4DeviceLowPower = 1, // Battery level is low
        HS4DeviceEr0, // The Scale failed to initialize
        HS4DeviceEr1, // Maximum weight has been exceeded
        HS4DeviceEr2, // The Scale can't capture a steady reading
        HS4DeviceEr4, // Bluetooth connection error
        HS4DeviceEr7, // Movement while measuring
            HS4DeviceEr8, //Invalidate
        HS4DeviceEr9, // Scale memory access error
            HS4DataZeor, // No memory
        HS4DeviceDisconnect, //Device disconnect
        HS4DeviceSendTimeout, // Communication error
            HS4DeviceRecWeightError, //
        HS4UserInvalidate = 111//User verify error
    }HS4DeviceError;

```

(3)HS5:

```

        typedef enum{
        IHSCOverTimeError = 0, // Communication error
        IHSCUserInScale=5, // Communication Error
        IHSCLowPower=6, // Make sure batteries are installed correctly, if the problem persists,
            replace with a new set of batteries.
        IHSCScaleEr0=7, // Remove the batteries, wait 1 minute and then replace with a new set
            of batteries.
        IHSCScaleEr1=8, // The current weight may be beyond the measurement range of 330
            lbs/150 kg.
        IHSCScaleEr2=9, // Stand still on all four electrodes with bare feet.
        IHSCScaleEr7=10, // Communication Error
        IHSCScaleEr8=11, // Communication Error
        IHSCScaleEr9=12, // Communication Error
            IHScaleBusy=13, // Scale is busy
            HS5DataZeor=14, // No memory
            IHSCScaleCreateUserTestError,
        HS5Disconnect, //Device disconnect

```



```
HS5UserInvalidate = 111//User verify error  
}HS5DeviceError;
```

7. 测试 demo