裂变智能穿戴产品
SDK 使用说明（iOS)

前言：SDK 希望你理解 BLE (Bluetooth Low Energy) 基本概念，熟练使用苹果的蓝牙框架 CoreBluetooth。

➢ 目录

➢ 更新记录

版本号	更新内容	发布日期	备注
V1.0.0	✧ 定义 SDK 基础架构，以及功能 api 接口，具体分为四类: ✧ 1.蓝牙管理器（FBBluetoothManager） ✧ 2.AT 协议指令集（FBAtCommand） ✧ 3.BG 协议指令集（FBBgCommand） ✧ 4.OTA 管理器（FBBluetoothOTA）	20201231	首次发版
V2.0.0	✧ 1.优化已知问题 ✧ 2.AT 指令回调数据类型由 NSDictionary 字典转换为使用对象模型，详见各回调 block ✧ 3.增加自定义表盘压缩算法协议	20220414	
V3.0.0	✧ 1.优化已知问题 ✧ 2.重构 blcok 回调方式，修复 block 回调异常问题	20220715	
V3.0.1	✧ 1.广播信息解析：适配号兼容 ✧ 2.新增印地语、孟加拉语、乌尔都语、波斯语 ✧ 3.新增获取精神压力记录协议、血压协议暂不可用	20221119	
V3.0.2	✧ 1.优化 OTA 通知问题 ✧ 2.GPS 运动互联控制增加 确认/取消 指令	20221214	
V3.0.3	✧ 1.新增定时心率、定时血氧、定时精神压力 检测开关设置协议 ✧ 2.FBMessageModel 新增消息推送类型 ✧ 3.FB_MOTIONMODE 新增运动类型	20221230	
V3.0.4	✧ 1.新增 get、set 通话音频开关协议 ✧ 2.新增 get、set 多媒体音频开关协议 ✧ 3.EM_FUNC_SWITCH 新增类型	20230105	
V3.0.5	✧ 1.修正获取血压记录协议 ✧ 2.新增获取运动高频心率记录(1 秒 1 次) ✧ 3.新增获取精神压力协议 ✧ 4.兼容获取手动测量记录协议 ✧ 5.FBFirmwareVersionObject 新增配置 ✧ 6.自动同步系统时间(同步 UTC 时间 + 设置时区)、获取 运动统计报告+运动详情纪录	20230130	

	✧ 7.FB_RECORDTYPE 新增类型 ✧ 8.FB_MULTIPLERECORDREPORTS 新增类型		
V3.0.6	✧ 1.修正 set 心率异常提醒参数合法性判断	20230202	
V3.0.7	✧ 1.优化设备搜索性能 ✧ 2.优化数据发送间隔 ✧ 3.新增"确认手机被找到"协议（FBAtCommand）fbUpPhoneConfirmedFoundDataWithBlock:	20230209	
V3.0.8	✧ 1.绑定请求超时时长由 30s 延长至 60s ✧ 2.新增获取设备 log 数据协议	20230211	
V3.0.9	✧ 1.FBFirmwareVersionObject 新增配置: ✧ 是否支持一次性推送多种运动模式 ✧ 支持一次性推送多种运动模式的个数，0 不支持 ✧ 2.新增一次性推送多种运动模式协议 ✧ 3.FB_OTANOTIFICATION 新增 OTA 通知类型: ✧ FB_OTANotification_Multi_Sport(9) ✧ 4.FBCustomDataTools 新增「多个运动类型 Bin 文件压缩合并成一个 Bin 文件」，配合「一次性推送多种运动模式」使用 ✧ 5.FBBluetoothOTA 新增进度模型 FBProgressModel，兼容一个 bin 文件包含多个包时的升级进度问题 ✧ 6.修正部分地区使用冬/夏令时，时区无法设置导致时间错误问题 ✧ 7.绑定设备请求可传入 Mac 地址，但是建议传 nil，SDK 内部会为你管理绑定密钥 ✧ 8.GPS 运动控制增加错误码 FB_GPS_MOTION_STATE_NONE 本地无此运动信息 ✧ 9.优化搜索设备，使用数据模型 FBPeripheralModel ✧ 10.FB_MOTIONMODE 新增运动类型: ✧ 法国式拳击(139)	20230301	
V3.1.0	✧ 1.FB_OTANOTIFICATION 新增 OTA 通知类型: ✧ FB_OTANotification_Multi_Dial_Built_in(200)	20230324	

	✧ FB_OTANotification_Multi_Sport_Built_in(201) ✧ 2.FBCustomDataTools「多个运动类型 Bin 文件压缩合并成一个 Bin 文件」压缩算法 API 调整（2048 或 4096） ✧ 3.新增"获取设备运动类型列表"协议（FBBgCommand）fbGetListOfDeviceMotionTypesWithBlock: ✧ 4.原厂 OTA SDK 更新：RTKOTASDK.framework		
V3.1.1	✧ 1.优化内部压缩算法 ✧ 2.修复"获取设备硬件信息"结构体版本错误问题 ✧ 2.新增"获取设备绑定状态"协议（FBAtCommand）fbGetBindingStatusRequestWithBlock: ✧ 3.新增"获取当前运动状态"协议（FBAtCommand）fbGetCurrentExerciseStateStatusWithBlock:	20230329	
V3.1.2	✧ 1.新增图片资源，自定义表盘，不同分辨率使用不同大小的切图	20230403	
V3.1.3	✧ 1.新增"设备确认被找到"协议（FBAtCommand）fbUpDeviceConfirmedFoundDataWithBlock: ✧ 2.新增"获取系统功能开关信息"协议（FBBgCommand）fbGetSystemFunctionSwitchInformationWithBlock: ✧ 3.新增"设置系统功能开关信息"协议（FBBgCommand）fbSetSystemFunctionSwitchInformation: withBlock: ✧ 4.FBFirmwareVersionObject 新增配置： ✧ 是否支持日常心率检测开关控制 ✧ 是否支持日常血氧检测开关控制 ✧ 是否支持日常血压检测开关控制 ✧ 是否支持日常精神压力检测开关控制 ✧ 5.新增二进制数据埋点 log 完整解析 ✧ 6.OTA 新增错误状态	20230412	

	FB_OTANotification_ERROR_Busy_Sport：设备处于运动中，请结束运动后重试... ✧ 7.修正"界面跳转测试"协议错误（FBAtCommand）fbUpInterfaceJumpTestCode:		
V3.1.4	✧ 1.优化已知问题 ✧ 2.新增"自定义表盘支持抗锯齿"处理 ✧ 3.FBFirmwareVersionObject 新增配置: ✧ 是否支持系统功能开关的设定和获取大数据指令（0252H / 0352H） ✧ 是否支持零星小睡 ✧ 是否支持自定义表盘抗锯齿 ✧ 4.新增尼泊尔语 ✧ 5.新增乌克兰语 ✧ 6.报告/记录 按时间戳由小到大排序 ✧ 7.广播信息中设配号兼容	20230518	
V3.1.5	✧ 1.修复"设置/获取 个人用户信息"协议已知错误问题 ✧ 2.新增"读取片外 flash 空间数据"协议，用于获取设备意外重启信息，供固件分析问题（FBBgCommand）fbReadOffChipFlashWithAddress: withLength: withBlock: ✧ 3.FBFirmwareVersionObject 新增: 适配号，长整形（部分手表支持） Hardfault 信息空间地址 Hardfault 信息空间尺寸 系统参数空间地址 系统参数空间尺寸 是否支持带适配号验证的 OTA 通知指令 是否支持hardfault信息和系統参数读取 是否支持表盘 CRC 校验 ✧ 4.EM_FUNC_SWITCH 新增类型: FS_TIMING_BP_WARN(33) FS_DEVICE_EXCEPTION_WARN(34) ✧ 5.新增表盘数据校验（UTC 或 CRC） ✧ 6.原厂 OTA SDK 更新至 v1.4.9 版本（RTKOTASDK.framework、RTKLEFoundation.framework）修改多包 OTA 文件顺序 ✧ 7.FBTypeRecordModel 记录生成周期	20230718	

	参数名由原先 createTimes 改为 recordingCycle，避免歧义，并且单位统一为秒；新增参数 ✧ 8.FBRecordDetailsModel 运动详情记录新增参数 一公里用时（一公里配速，单位秒）KilometerPace，一英里用时（一英里配速，单位秒）MilePace，仅部分设备支持，具体根据参数 记录格式定义（recordDefinition）而定 ✧ 9.自定义表盘抗锯齿切图更新 ✧ 10.优化记录/报告排序及其他已知问题		
V3.1.6	✧ 1.EM_FUNC_SWITCH 新增类型： FS_AGPS_LOCATION_REQUEST(35) FS_AGPS_DATA_REQUEST(36) ✧ 2.FB_OTANOTIFICATION 新增 OTA 通知类型： FB_OTANotification_AGPS_Package(30) ✧ 3.新增"推送 AGPS 位置基础信息(经纬度 UTC)"协议（FBBgCommand）fbPushAGPSLocationInformation: withBlock: ✧ 4.新增"同步 AGPS 定位数据"协议（FBBgCommand）fbSynchronizeAGPSPositioningData: withBlock: ✧ 5.FB_MOTIONMODE 新增运动类型： 沙滩排球(140) ✧ 6.FB_LANGUAGES 新增语言类型： FB_SDK_ms(29) FB_SDK_sk(30) FB_SDK_my(31) FB_SDK_da(32) ✧ 6.SDK 同时支持 真机、模拟器 编译运行（注意：模拟器无法使用蓝牙）	20230824	
V3.1.7	✧ 1.FBFirmwareVersionObject 新增标志位： 是否支持静息心率 是否支持 AGPS 定位 ✧ 2.修复"获取运动定位记录"协议已知错误问题（FBBgCommand）fbGetMotionLocationRecordDataStartTime: forEndTime: withBlock:	20231010	
V3.1.8	✧ 1.FBMacro.h 调整枚举声明方式	20231116	

	✧ 2.广播信息优化 ✧ 3.OTA 进度回调优化 ✧ 4.时间格式转换优化 ✧ 5.断开连接同时是否清除连接历史记录 @see disconnectPeripheralAndClearHistory: ✧ 6.其他已知问题的优化		
V3.1.9	✧ 1.优化绑定密钥缓存逻辑	20231128	
V3.2.0	✧ 1.定位记录详情增加 公/英里 里程点 ✧ 2.修改翻译 ✧ 3.优化已知问题	20231215	
V3.2.1	✧ 1.新增生成 AGPS 星历 bin 文件数据 (FBCustomDataTools) fbGenerateAGPSEphemerisBinFileDataWithModel: ✧ 2.适配兼容新协议版本的数据格式解析 ✧ 3.FBFirmwareVersionObject 新增标志位: 芯片厂商类型 (瑞昱、海思) ✧ 4.不同芯片类型使用不同的 OTA 方法 (SDK 内部已做区分, FBBluetoothOTA 传入对应的 otaType 即可) ✧ 5.qz 压缩算法增加 crc 校验和, 以及其他优化 ✧ 6.优化已知问题	20240122	
V3.2.2	✧ 1. Fission_Sdk_iOS.framework 最低系统版本要求由 iOS10.0+ 提高至 iOS12.1+ ✧ 2. 不再支持 x86_64(模拟器)。新增 Framework 库依赖: SCompressLib.framework MagicTool.framework Starscream.framework ffmpegKit-kit @link https://github.com/arthenica/ffmpeg-kit (参考 pod 'ffmpeg-kit-ios-full', '~> 6.0' 最低系统版本要求 12.1) ✧ 3.新增运动类型: 141-151 ✧ 4.FBFirmwareVersionObject 新增标志位:	20240822	

	是否支持 PATCH 版本字段 是否支持日程功能 是否支持紧急联系人（SOS） 是否支持今日天气显示城市名称 是否支持 JS 应用 支持常用联系人设置的个数 支持紧急联系人设置的个数 是否支持今日天气显示城市名称 是否支持 JS 应用 是否支持多媒体空间 是否支持视频表盘 ✧ 5.新增"获取未使用的 日程信息 ID"协议（FBAtCommand）fbGetUnusedScheduleIDWithBlock: ✧ 6.新增"获取日程信息"协议（FBBgCommand）fbGetScheduleInforWithBlock: ✧ 7.新增"设置日程信息"协议（FBBgCommand）fbSetSchedulenforWithScheduleModel:withRemoved:withBlock: ✧ 8.新增"获取紧急联系人信息"协议（FBBgCommand）fbGetEmergencyContactListWithBlock: ✧ 9.新增"设置紧急联系人信息"协议（FBBgCommand）fbSetEmergencyContactListWithModel:withBlock: ✧ 10.优化运动记录数据排序问题 ✧ 11.优化海思芯片 OTA 方案 ✧ 12.新增"OTA 文件增加文件信息"（FBCustomDataTools）createFileName:withFileData:withOTAType:方法，当前此方法仅用于海思芯片方案 ✧ 13.自定义表盘兼容海思芯片方案，支持自定义视频表盘（FBCustomDataTools）fbGenerateCustomDialBinFileDataWithDialModel: ✧ 14.支持百度相关：语音识别、文字翻译、文心一言、文字生成语音、百度导航（文字导航）。详见 FBBaiduCloudKit 类 ✧ 15.支持系统麦克风的调用封装。详见		

	FBAudioRecorder 类。注意需要在 Info.plist 中添加 NSMicrophoneUsageDescription 权限 ✧ 16.新增"获取系统空间使用信息"协议 (FBBgCommand) fbGetSystemSpaceUsageInforWithBlock: ✧ 17.新增"获取表盘列表文件信息"协议 (FBBgCommand) fbGetDialListFileInforWithBlock: ✧ 18.新增"获取 JS 应用列表文件信息"协议 (FBBgCommand) fbGetJsAppListFileInforWithBlock: ✧ 19.新增"删除表盘列表文件信息"协议 (FBBgCommand) fbDeleteDialListFileInfor:withBlock: ✧ 20.新增"删除 JS 应用列表文件信息"协议 (FBBgCommand) fbDeleteJsAppListFileInfor:withBlock: ✧ 21.支持【支付宝】【乘车码】功能，SDK 内部处理，外部无需任何调用 ✧ 22.新增支持瑞昱 8773 芯片 ✧ 23.优化自定义表盘图片抗锯齿 ✧ 24.优化已知问题		

➢ 环境要求

iOS 12.1 及以上操作系统，不支持模拟器

依赖

- CoreBluetooth.framework

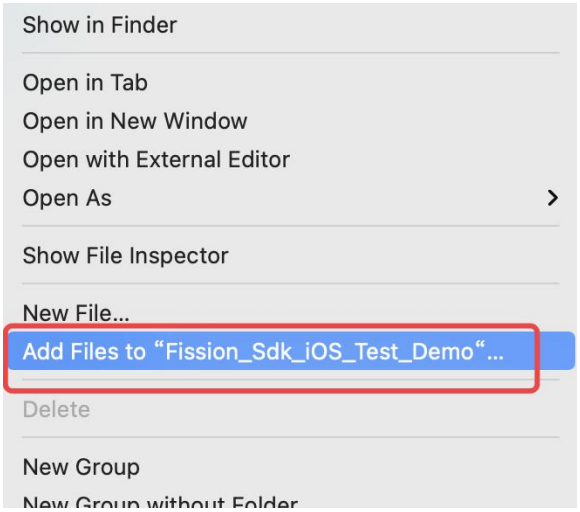
➢ 导入 SDK

方式一：通过 CocoaPods 安装. 在 Podfile 中添加以下内容

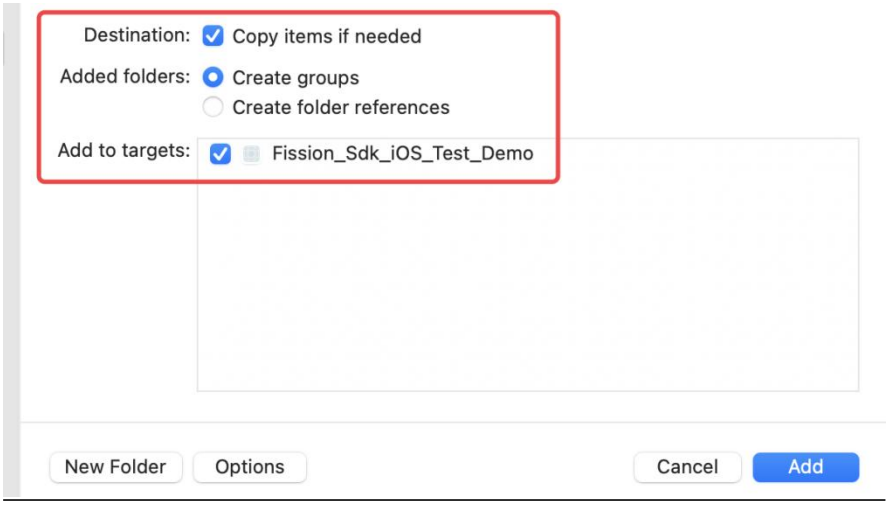
pod 'Fission_Sdk_iOS', git: 'https://github.com/linwear/Fission_Sdk_iOS.git'

方式二：手动导入 SDK

1、将 Fission_Sdk_iOS.framework、RTKOTASDK.framework、RTKLEFoundation.framework、SCompressLib.framework、MagicTool.framework、Starscream.framework 文件 Add File 导入工程:

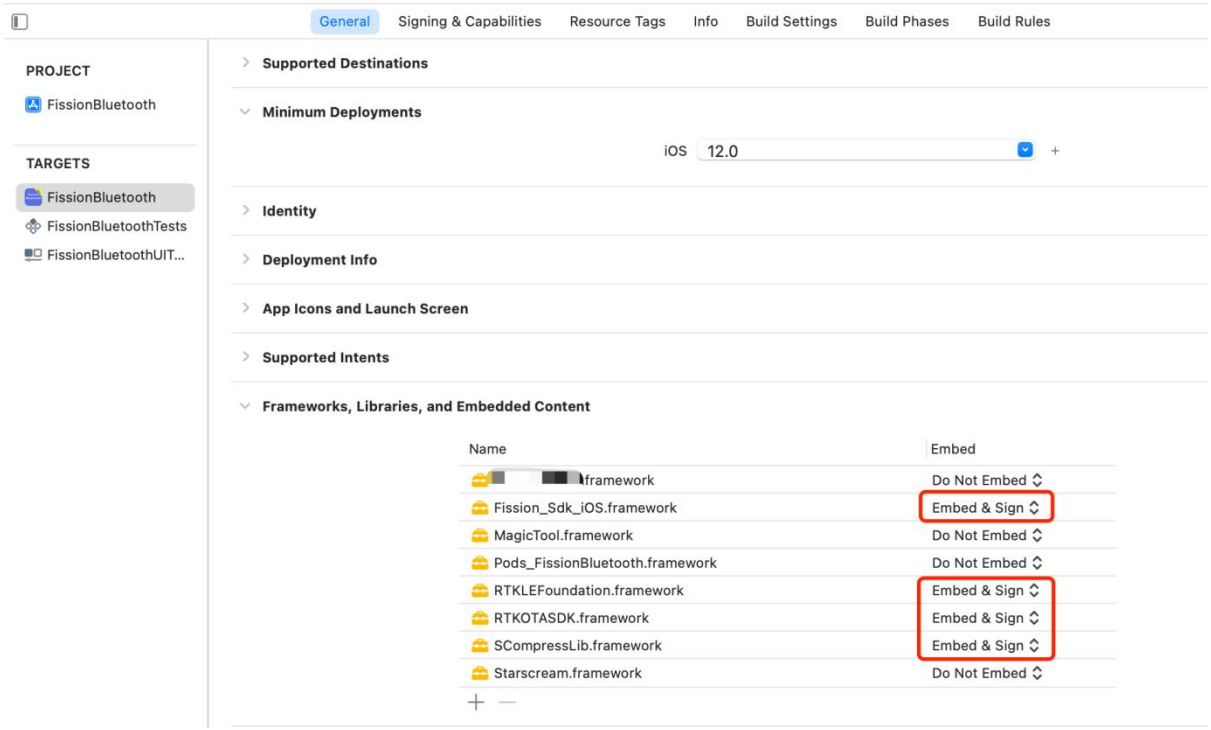


此时应注意勾选选项：

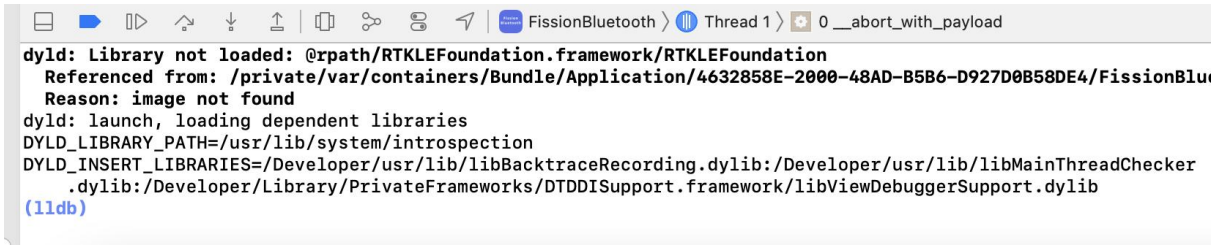


2、集成依赖 FFmpeg（参考 https://github.com/arthenica/ffmpeg-kit.git）

3、在 TARGETS - General 中修改 Fission_Sdk_iOS.framework、RTKOTASDK.framework、RTKLEFoundation.framework、SCompressLib.framework 的嵌入方式为 Embed&Sign：



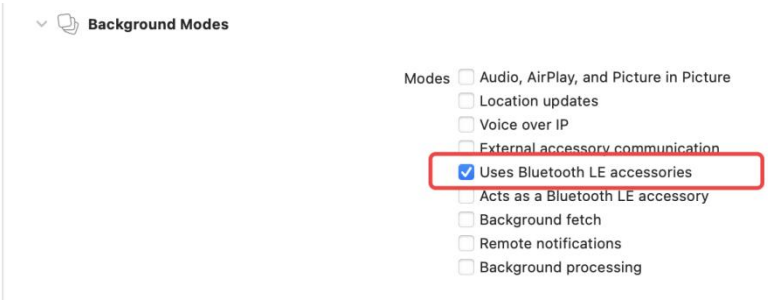
Note：为了使 app 能够执行 SDK 的代码，framework 需要包含在 app bundle 中。如果无法执行，console 中打印 image not found，表示这一步没有完成。



4、在 TARGETS - Build Settings - Other Linker Flags 中添加 -ObjC

➢ 设置蓝牙后台模式

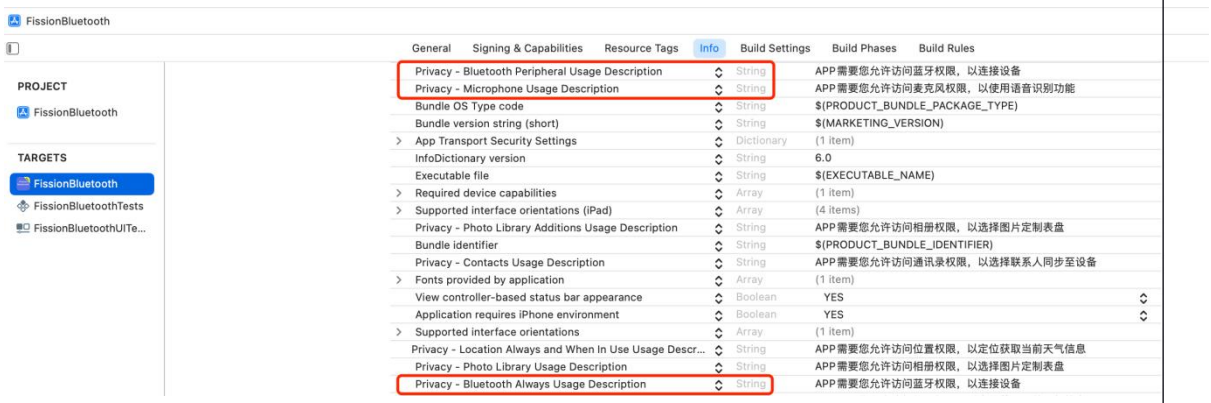
后台运行模式，开启以下 Background mode：



➢ 设置隐私权限

隐私数据使用说明，iOS 需要 app 对使用的隐私数据提供可视化的文字说明描述，否则 app 无法启动，需要在 info plist 中添加以下使用描述：

- Privacy - Bluetooth Peripheral Usage Description
- Privacy - Bluetooth Always Usage Description
- Privacy - Microphone Usage Description



➢ 开始使用，导入头文件

由于 SDK 使用 Objective-C 与 Swift 混合开发。如果你的项目没有自动生成桥接头文件（Bridging Header），请手动创建一个。这里不对 Bridging Header 做过多叙述。

在使用 SDK 接口的源代码中，使用`#import <Fission_Sdk_iOS/Fission_Sdk_iOS.h>`引入头文件即可使用相关 API

①初始化蓝牙管理器 **FBBluetoothManager.sharedInstance** 设置回调监听方法：

```
/// 蓝牙状态改变回调
```

```
-(void)fbOnCentralManagerDidUpdateStateWithBlock:(FBOnCentralManagerDid
UpdateStateBlock)fbBlock;

/// 搜索蓝牙设备回调
-(void)fbDiscoverPeripheralsWithBlock:(FBDiscoverPeripheralsBlock)fbBlo
ck;

/// 设备连接 成功/失败 回调
-(void)fbOnConnectedAtChannelWithBlock:(FBOnConnectedAtChannelBlock)fbB
lock;

/// 设备断开连接回调
-(void)fbOnDisconnectAtChannelWithBlock:(FBOnDisconnectAtChannelBlock)f
bBlock;

/// 蓝牙系统错误回调
-(void)fbBluetoothSystemErrorWithBlock:(FBBluetoothSystemErrorBlock)fbB
lock;

/// 开始扫描设备
- (void)scanForPeripherals;

/// 停止扫描设备
- (void)cancelScan;

/// 连接设备
- (void)connectToPeripheral:(CBPeripheral * _Nonnull)peripheral;

/// 断开设备
- (void)disconnectPeripheral;
```

更多 api 接口方法使用参考 **FBBluetoothManager** 类;

代码示例 1，设置监听扫描到设备的回调:

```
// 扫描到设备回调方法
[FBBluetoothManager.sharedInstance fbDiscoverPeripheralsWithBlock:^(BOOL isPair, NSString * _Nonnull device_Name, NSString *
    _Nonnull mac_Address, NSString * _Nonnull adapt_Number, CBPeripheral * _Nonnull peripheral, NSDictionary * _Nonnull
    advertisementData, NSNumber * _Nonnull RSSI) {

    // do something...
}];
```

代码示例 2，开始扫描设备:

```objc
// 开始扫描设备
[FBBluetoothManager.sharedInstance scanForPeripherals];
```

②AT 指令集 **FBAtCommand.sharedInstance**，更多 api 接口方法使用参考 **FBAtCommand** 类；

代码示例，请求绑定设备：

```objc
// 绑定设备请求
[FBAtCommand.sharedInstance fbBindDeviceRequestWithBlock:^(NSInteger responseObject, NSError * _Nullable error) {
    if (error) {
        // 失败
    } else {
        // 根据自身业务处理结果
    }
}];
```

③BG 指令集 **FBBgCommand.sharedInstance**，具体 api 接口方法使用参考 **FBBgCommand** 类；

代码示例，请求设备硬件信息：

```objc
// 获取设备硬件信息请求
[FBBgCommand.sharedInstance fbGetHardwareInformationDataWithBlock:^(FB_RET_CMD status, float progress, FBDeviceInfoModel *
    _Nullable responseObject, NSError * _Nullable error) {
    if (error) {
        // 失败
    } else if (status == FB_DATATRANSMISSIONDONE) {
        // 成功
    }
}];
```

④OTA 管理器 **FBBluetoothOTA.sharedInstance**，固件升级、在线表盘、自定义表盘、运动推送・・・通过传入 `FB_OTANOTIFICATION` 参数，区分 OTA 类型，具体 api 接口方法使用参考 **FBBluetoothOTA** 类；

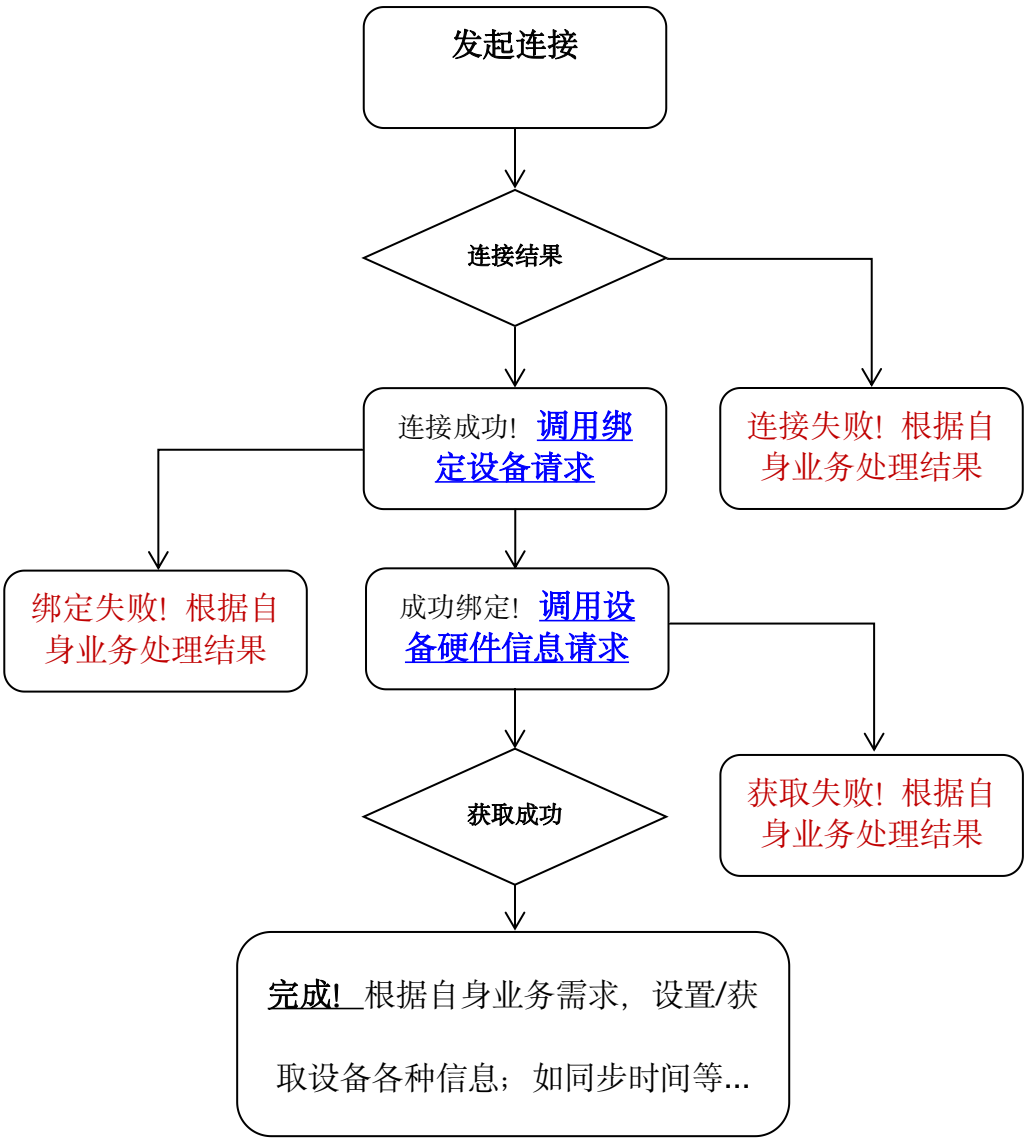
代码示例，固件升级：

```objc
// 升级固件
[FBBluetoothOTA.sharedInstance fbStartCheckingOTAWithBinFileData:binFile withOTAType:FB_OTANotification_Firmware
    withBlock:^(FB_RET_CMD status, float progress, FBOTADoneModel * _Nonnull responseObject, NSError * _Nonnull error) {
    if (error) {
        // 失败
    } else if (status==FB_DATATRANSMISSIONDONE){
        // 成功
    } else if (status==FB_INDATATRANSMISSION){
        // progress进度
    }
}];
```

自定义表盘 bin 文件支持本地生成，请参考：

```objc
// 生成自定义表盘文件
NSData *file = [FBCustomDataTools.sharedInstance fbGenerateCustomDialBinFileDataWithDialModel:model];
```

➢ 固定流程说明



Note：SDK 底层部分功能处理依赖【设备硬件信息】，硬件功能相关信息

FBAllConfigObject.firmwareConfig 详见 **FBFirmwareVersionObject** 类；

①发起连接
②连接成功，请求绑定设备
③绑定设备成功，请求获取设备硬件信息
④设备硬件信息请求成功，开始根据自身业务需求，调用其他协议接口...

➢ 基础控制与查询 API（FBAtCommand）

1、获取设备电量信息

```
- (void)fbReqBatteryStatusDataWithBlock:(FBReqBatteryStatusBlock _Nonnull)fbBlock;
/*
 手表设备电量信息 | Watch device battery info
 */
@interface FBBatteryInfoModel : NSObject
/**
 电池电量状态 | Battery state
*/
@property (nonatomic, assign) FB_BATTERYLEVEL batteryState;
/**
 电池电量 | Battery level
*/
@property (nonatomic, assign) NSInteger batteryLevel;
@end

typedef enum {
    BATT_NORMAL    = 0,  //正常 | Normal
    BATT_LOW_POWER = 1,  //低压 | Low power
    BATT_CHARGING  = 2,  //充电中 | Charging
    BATT_FULL      = 3,  //电池满 | Full power
}FB_BATTERYLEVEL;
```

2、获取设备版本信息

```
- (void)fbReqDeviceVersionDataWithBlock:(FBReqDeviceVersionBlock _Nonnull)fbBlock;
/*
 设备版本信息 | Device version information
 */
@interface FBDeviceVersionModel : NSObject
/**
 硬件版本号 | Hardware version number
*/
@property (nonatomic, copy) NSString *hardwareVersion;
/**
 软件版本号 | Software version number
*/
```

```
@property (nonatomic, copy) NSString *softwareVersion;
@end
```

3、获取协议版本信息

```
- (void)fbReqProtocolVersionDataWithBlock:(FBReqProtocolVersionBlock
_Nonnull)fbBlock;
/**
 协议版本号 | Protocol version number
*/
NSString *responseObject;
```

4、获取 UTC 时间

```
- (void)fbReqUTCTimeDataWithBlock:(FBGet_AT_ResultCallBackBlock
_Nonnull)fbBlock;
/**
 UTC 时间（秒） | UTC time (seconds)
*/
NSInteger responseObject;
```

5、获取时区

```
- (void)fbReqTimezoneDataWithBlock:(FBGet_AT_ResultCallBackBlock
_Nonnull)fbBlock;
/**
 时区（分钟） | Time zone (minutes)
*/
NSInteger responseObject;
```

6、~~同步 UTC 时间~~（建议使用 fbAutomaticallySynchronizeSystemTimeWithBlock：替换）

```
~~- (void)fbSynchronizeUTCTimeWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;~~
```

7、~~设置时区~~（建议使用 fbAutomaticallySynchronizeSystemTimeWithBlock：替换）

```
~~- (void)fbUpTimezoneMinuteData:(NSInteger)timeZoneMinute
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;~~
/**
 时区（分钟） | Time zone (minutes)
*/
```

```
NSInteger timeZoneMinute;
```

8、设置系统时间（同步 UTC 时间+设置时区）

```
-
(void)fbAutomaticallySynchronizeSystemTimeWithBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

9、设置时间显示模式

```
- (void)fbUpTimeModeData:(FB_TIMEDISPLAYMODE)hoursMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 时间显示模式 | Time display mode
*/
TIMEDISPLAYMODE hoursMode;

typedef enum {
    FB_TimeDisplayMode12Hours = 12,  //12 小时制 | 12 hour system
    FB_TimeDisplayMode24Hours = 24,  //24 小时制 | 24 hour system
}FB_TIMEDISPLAYMODE;
```

10、设置语言

```
- (void)fbUpLanguageData:(FB_LANGUAGES)language
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 语言 | language
*/
LANGUAGES language;

typedef enum {
    FB_SDK_zh_Hans = 0,   //中文简体 | Simplified Chinese
    FB_SDK_en      = 1,   //英文 | English
    FB_SDK_ja      = 2,   //日语 | Japanese
    FB_SDK_fr      = 3,   //法语 | French
    FB_SDK_de      = 4,   //德语 | German
    FB_SDK_es      = 5,   //西班牙语 | Spanish
    FB_SDK_it      = 6,   //意大利语 | Italian
    FB_SDK_pt      = 7,   //葡萄牙语 | Portuguese
    FB_SDK_ru      = 8,   //俄语 | Russian
    FB_SDK_cs      = 9,   //捷克语 | Czech
    FB_SDK_pl      = 10,  //波兰语 | Polish
```

```
    FB_SDK_zh_Hant = 11,  //中文繁体 | Chinese traditional
    FB_SDK_ar      = 12,  //阿拉伯语 | Arabic
    FB_SDK_tr      = 13,  //土耳其语 | Turkish
    FB_SDK_vi      = 14,  //越南语 | Vietnamese
    FB_SDK_ko      = 15,  //韩语 | Korean
    FB_SDK_he      = 16,  //希伯来语 | Hebrew
    FB_SDK_th      = 17,  //泰语 | Thai
    FB_SDK_id      = 18,  //印尼语 | Indonesian
    FB_SDK_nl      = 19,  //荷兰语 | Dutch
    FB_SDK_el      = 20,  //希腊语 | Greek
    FB_SDK_sv      = 21,  //瑞典语 | Swedish
    FB_SDK_ro      = 22,  //罗马尼亚语 | Romanian
    FB_SDK_hi      = 23,  //印地语 | Hindi
    FB_SDK_bn      = 24,  //孟加拉语 | Bangla
    FB_SDK_ur      = 25,  //乌尔都语 | Urdu
    FB_SDK_fa      = 26,  //波斯语 | Persian
    FB_SDK_ne      = 27,  //尼泊尔语 | Nepali
    FB_SDK_uk      = 28,  //乌克兰语 | Ukrainian
}FB_LANGUAGES;
// 更多类型请查看 SDK 内 FBMacro.h 文件
```

11、设置距离单位

```
- (void)fbUpDistanceUnitData:(FB_DISTANCEUNIT)units
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 单位 | Unit
*/
FB_DISTANCEUNIT units;

typedef enum {
    FB_EnglishUnits = 0,  //英制单位 | English units
    FB_MetricUnit   = 1,  //公制单位 | Metric unit
}FB_DISTANCEUNIT;
```

12、设置震动提醒开关

```
- (void)fbUpShakeAlterSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

13、设置抬腕亮屏开关

```
- (void)fbUpWristSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

14、进入/退出拍照模

```
- (void)fbUpTakePhotoStatusData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 进入 NO 退出 | YES Enter NO Exit */
BOOL switchMode;
```

15、开启/关闭数据流

```
- (void)fbUpDataStreamData:(NSInteger)second
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 数据返回时间间隔（单位秒） 0 表示关闭 | Data return time interval (unit: second) 0 means closed
*/
NSInteger second;
```

16、设置心率开关

```
- (void)fbUpHeartRateData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

17、收到设备即使拍照的回调

```
- (void)fbUpTakePhotoClickDataWithBlock:(FBUpTakePhotoClickBlock
_Nonnull)fbBlock;
// 设备端操作了摇一摇拍照，会主动通过此回调通知 app 端
```

18、手机查找设备

```
- (void)fbUpFindDeviceDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

19、收到设备查找手机的回调

```
- (void)fbUpFindPhoneDataWithBlock:(FBUpFindPhoneBlock _Nonnull)fbBlock;
// 设备端操作了查找手机，会主动通过此回调通知 app 端
```

20、手机确认被找到

```
- (void)fbUpPhoneConfirmedFoundDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
// 当设备成功查找到手机时，APP 调用该方法可停止设备查找手机
```

21、收到设备取消查找手机的回调

```
- (void)fbAbandonFindingPhoneWithBlock:(FBAbandonFindingPhoneBlock
_Nonnull)fbBlock;
// 设备端取消了查找手机，会主动通过此回调通知 app 端
```

22、收到蓝牙配对成功的回调

```
- (void)fbUpPairingCompleteDataWithBlock:(FBUpPairingCompleteBlock
_Nonnull)fbBlock;
// 设备与手机蓝牙配对成功，会主动通过此回调通知 app 端
```

23、重启设备

```
- (void)fbUpRebootDeviceDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

24、恢复出厂设置

```
- (void)fbUpResetDeviceDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

25、软关机

```
- (void)fbUpSoftDownDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

26、启动 OTA 升级模式

```
- (void)fbUpOpenOTADataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

27、安全确认

```
- (void)fbUpSafetyConfirmDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

28、启动/退出自检模式

```
- (void)fbUpSelfTestData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 启动 NO 退出 | YES Start NO Exit */
BOOL switchMode;
```

29、清除用户信息

```
- (void)fbUpClearUserInfoDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

30、清除运动数据

```
- (void)fbUpClearSportDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

31、设置设备主动断开连接

```
- (void)fbUpDisConnectDataWithBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

32、界面跳转测试

```
- (void)fbUpInterfaceJumpTestCode:(NSInteger)interfaceCode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 指定界面代号 | Specify interface code
*/
NSInteger interfaceCode;
```

33、女性生理状态设定

```
-(void)fbUpFemalePhysiologicalStateData:(FB_FEMALEPHYSIOLOGICALSTATE)stateCode withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 女性生理状态 | Female's physiological state
*/
FB_FEMALEPHYSIOLOGICALSTATE stateCode;

typedef enum {
    FB_FPS_NotUsed             = 0,  //未启用 | Not used
    FB_FPS_Pregnancy           = 1,  //怀孕期 | Pregnancy
    FB_FPS_Menstruation        = 2,  //月经期 | Menstruation
    FB_FPS_Safety              = 3,  //安全期 | Safety period
    FB_FPS_Ovulation           = 4,  //排卵期 | During ovulation
    FB_FPS_OvulationDay        = 5,  //排卵日 | Ovulation day
    FB_FPS_PregnancyPreparation = 6,  //备孕期 | Pregnancy preparation period
}FB_FEMALEPHYSIOLOGICALSTATE;
```

34、获取未使用的 记事体醒/闹钟信息 ID

```
- (void)fbGetUnusedClockIDWithBlock:(FBGet_AT_ResultCallBackBlock _Nonnull)fbBlock;

// 在 app 端新建记事体醒/闹钟信息时，需先调用该接口查询可用的 ID

/**
 可用 ID | Available IDs
*/
NSInteger responseObject;
```

35、开启/退出短跑模式

```
- (void)fbUpSprintMode:(FB_SPRINTMODE)mode withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 开启后采集速度由一分钟一笔运动详情，改为一秒一次
/**
 短跑模式开关 | Sprint mode switch
*/
FB_SPRINTMODE mode;
```

```
typedef enum {
    FB_SPRINTMODE_OFF = 0,  //关闭 | Close
    FB_SPRINTMODE_ON  = 1,  //开启 | Open
}FB_SPRINTMODE;
```

36、监听设备的定位请求

```
- (void)fbUpPositioningSwitchWithBlock:(FBUpPositioningSwitchBlock _Nonnull)fbBlock;
// app 端接收到该回调时，需要主动上报一次定位信息给设备端
```

```
@interface FBBgCommand : NSObject
#pragma mark - App推送手机定位信息 | App push mobile location information
/**
 App推送手机定位信息 | App push mobile location information
 @param longitude      经度 | Longitude
 @param latitude       纬度 | Latitude
*/
- (void)fbPushMobileLocationInformationWithLongitude:(float)longitude withLatitude:(float)latitude withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

37、OTA 类型通知

```
- (void)fbUpOTANotificationWithType:(FB_OTANOTIFICATION)type withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

38、进入/退出生产模式

```
- (void)fbUpProductionTestModeIsOpen:(BOOL)isOpen withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 进入 NO 退出 | YES Enter NO Exit */
BOOL isOpen;
```

39、监听设备端功能状态变更回调

```
-(void)fbReceiveFunctionSwitchSynchronizationWithBlock:(FBReceiveFunctionSwitchSynchronizationBlock _Nonnull)fbBlock;
/*
 手表设备功能变更通知 | Watch device function change notice
 */
@interface FBWatchFunctionChangeNoticeModel : NSObject
/**
 变更的功能 | Changed functions
*/
@property (nonatomic, assign) EM_FUNC_SWITCH functionMode;
/**
 功能更改值 | Function change value
```

```
 @note 根据变更的功能类型，功能更改值代表的含义不同，具体参考上述枚举【EM_FUNC_SWITCH】 / According to the changed function type, the meaning of the function change value is different. Refer to the above enumeration 【EM_FUNC_SWITCH】 for details
 */
@property (nonatomic, assign) NSInteger functionChangeValue;
@end
typedef enum {
    FS_NULL                  = 0,   //无 | Nothing
    FS_SENSOR_GATHER          = 1,   //体征数据采集总开关状态，0 关 1 开 | Sign data acquisition master switch status, 0 off and 1 on
    FS_MOTOR_ENABLE           = 2,   //振动开关状态，0 关 1 开 | Vibration switch status, 0 off, 1 on
    FS_DONT_DISTURB_WARN       = 3,   //勿扰开关状态，0 关 1 开 | Do not disturb switch status, 0 off and 1 on
    FS_CLOCK_1_WARN           = 4,   //闹钟 1 的开关状态，0 关 1 开 | Alarm 1 switch status, 0 off and 1 on
    FS_CLOCK_2_WARN           = 5,   //闹钟 2 的开关状态，0 关 1 开 | Alarm 2 switch status, 0 off and 1 on
    FS_CLOCK_3_WARN           = 6,   //闹钟 3 的开关状态，0 关 1 开 | Alarm 3 switch status, 0 off and 1 on
    FS_CLOCK_4_WARN           = 7,   //闹钟 4 的开关状态，0 关 1 开 | Alarm 4 switch status, 0 off and 1 on
    FS_CLOCK_5_WARN           = 8,   //闹钟 5 的开关状态，0 关 1 开 | Alarm 5 switch status, 0 off and 1 on
    FS_LOWBATTERY_WARN        = 9,   //低压提醒功能开关状态，0 关 1 开 | Low voltage reminder function switch status, 0 off and 1 on
    FS_TARGET_DAY_WARN        = 10,  //日目标提醒检测总开关状态，0 关 1 开 | Daily target alert detection master switch status, 0 off and 1 on
    FS_TARGET_WEEK_WARN       = 11,  //周目标提醒检测总开关状态，0 关 1 开 | Weekly target alert detection master switch status, 0 off and 1 on
    FS_TARGET_SELF_WARN       = 12,  //自我鼓励目标提醒检测总开关状态，0 关 1 开 | Self encouragement target alert detection master switch status, 0 off and 1 on
    FS_HEARTRATE_LEVEL_WARN    = 13,  //心率超标提醒开关状态，为 0 关 非 0 开 | The heart rate exceeds the limit reminder switch status, which is 0 off and not 0 on
    FS_WEARING_STATE_WARN      = 14,  //佩戴状态，0 未佩戴 1 佩戴 | Wearing status, 0 not wearing, 1 wearing
    FS_TAKEPHOTOS_WARN        = 15,  //拍照模式开关状态，0 关 1 开 | Photo mode switch status, 0 off and 1 on
    FS_STATEOFCHARGE_WARN      = 16,  //设备充电状态更新，0 放电状态、1 低压状态、2 充电状态、3 充满状态 | The charging state of the equipment is updated, including 0 discharge state, 1 low voltage state, 2 charging state and 3 full state
    FS_MUSICINTERFACESTATUS    = 17,  //进入音乐界面状态 | Music interface status
    FS_BRIGHTSCREENTIMECHANGES = 18,  //亮屏时长改变 | The duration of bright screen changes
    FS_WRISTLIFT_WARN         = 19,  //抬腕开关状态，0 关 1 开 | Wrist lifting switch status, 0 off and 1 on
    FS_PERCENTAGE_BATTERY      = 20,  //当前电池电量百分比 | Current battery power percentage
    FS_WATER_DRINKING_WARN     = 21,  //喝水提醒开关状态，0 关 1 开 | Water drinking reminder switch status, 0 off and 1 on
    FS_SEDENTARY_WARN         = 22,  //久坐提醒开关状态，0 关 1 开 | Sedentary reminder switch status, 0 off and 1 on
    FS_OTA_PERCENTAGE         = 23,  //OTA 百分比 | OTA percentage
    FS_MUTE_SWITCH            = 24,  //静音开关同步（安卓专用） | Mute switch synchronization (Android only)
    FS_OTA_INTERFACE_STATUS    = 25,  //手表 OTA 升级界面状态，1 进入 OTA 界面，0 退出 OTA 界面 | Watch OTA upgrade interface status, 1 enters the OTA interface, 0 exits the OTA interface
```

```
    FS_ALARMCLOCK_CHANGENOTICE = 26,  //手表记事提醒/闹钟信息变更通知事件 | Watch reminder / alarm clock information change notification event

    FS_TIMING_HR_WARN          = 28, //定时心率检测开关状态,0 关 1 开 | Timing heart rate detection switch status, 0 off 1 on
    FS_TIMING_SPO2_WARN        = 29, //定时血氧检测开关状态，0 关 1 开 | Timing blood oxygen detection switch status, 0 off 1 on
    FS_TIMING_STRESS_WARN      = 30, //定时精神压力检测开关状态，0 关 1 开 | Timing mental stress detection switch status, 0 off 1 on
    FS_CALLAUDIO_WARN          = 31, //通话音频开关状态，0 关 1 开 | Call audio switch status, 0 off 1 on
    FS_MULTIMEDIAAUDIO_WARN    = 32, //多媒体音频开关状态，0 关 1 开 | Multimedia audio switch status, 0 off 1 on
    FS_TIMING_BP_WARN          = 33, //定时血压检测开关状态，0 关 1 开 | Timing blood pressure detection switch status, 0 off 1 on
    FS_DEVICE_EXCEPTION_WARN   = 34, //设备异常信息读取请求 | Device exception information read request
    FS_AGPS_LOCATION_REQUEST   = 35, //AGPS 位置基础信息(经纬度 UTC)请求 | AGPS location basic information (longitude and latitude UTC) request
    FS_AGPS_DATA_REQUEST       = 36, //AGPS 定位数据请求 | AGPS positioning data request

    FS_OTHER_EXPAND            = 255  //更多功能待拓展 | More functions to be expanded
}EM_FUNC_SWITCH;
// 更多类型请查看 SDK 内 FBMacro.h 文件
```

40、设置温度单位

```
- (void)fbUpTemperatureUnitWithUnit:(FB_TEMPERATUREUNIT)unit withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
温度单位 | Temperature unit
*/
FB_TEMPERATUREUNIT  unit;

typedef enum {
    FB_Centigrade      = 0,  //摄氏度 C | Centigrade(C)
    FB_FahrenheitDegree = 1,  //华氏度 F | Fahrenheit degree(F)
}FB_TEMPERATUREUNIT;
```

41、获取亮屏时长

```
-(void)fbGetTheDurationOfBrightScreenWithBlock:(FBGet_AT_ResultCallBackBlock _Nonnull)fbBlock;
/**
亮屏时长（秒） | Duration of screen lighting (seconds)
*/
```

```
NSInteger responseObject;
```

42、设置亮屏时长

```
- (void)fbSetTheDurationOfBrightScreenWithDuration:(int)duration
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 亮屏时长（秒） | Duration of screen lighting (seconds)
*/
int duration;
```

43、切换至指定表盘

```
- (void)fbTogglesTheSpecifiedDialWithIndex:(int)index
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/**
 表盘索引 | Dial index
*/
int index;
```

44、设置震动反馈

```
- (void)fbVibrationFeedbackSwitchWithMode:(BOOL)mode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL mode;
```

45、请求绑定设备

```
/** macAddress              手表 Mac 地址，可不传，为 nil 时 SDK 内部处理，建议传 nil */
- (void)fbBindDeviceRequest:(NSString * _Nullable)macAddress
withBlock:(FBGet_AT_ResultCallBackBlock _Nonnull)fbBlock;
/**
 绑定结果 | Binding Results
 0 拒绝绑定
 1 同意绑定
 2 已被绑定
 3 确认超时
 4 递交秘钥错误
 5 递交秘钥正确
 6 无需绑定
*/
NSInteger responseObject;
```

46、请求解绑设备

```
/** macAddress            手表 Mac 地址，可不传，为 nil 时 SDK 内部处理，建议传 nil */
- (void)fbUnbindDeviceRequest:(NSString * _Nullable)macAddress withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

47、获取当天静息心率

```
-(void)fbGetRestingHeartRateOfTheDayWithBlock:(FBGet_AT_ResultCallBackBlock _Nonnull)fbBlock;
/**
 当天静息心率 | The resting heart rate of the day
*/
NSInteger responseObject;
```

48、获取指定提示功能

```
- (void)fbGetPromptFunctionWithMode:(FB_PROMPTFUNCTION)mode withBlock:(FBGet_AT_ResultCallBackBlock _Nonnull)fbBlock;
/**
 提示阀值，等于 0 则代表关闭 | Prompt threshold value, equal to 0 means closed
*/
NSInteger responseObject;
/**
 提示功能 | Prompt function
*/
FB_PROMPTFUNCTION mode;

typedef enum {
    FB_ExerciseHeartRate = 1,  //运动心率超高提示 | Exercise heart rate ultra-high prompt
    // 更多... 待拓展 | More... To be expanded
}FB_PROMPTFUNCTION;
```

49、设置指定提示功能

```
- (void)fbSetPromptFunctionWithMode:(FB_PROMPTFUNCTION)mode withThreshold:(NSInteger)threshold withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

50、app 端同步 GPS 运动状态到设备端

```
- (void)fbSynchronizationGPS_MotionWithModel:(FBGPSMotionActionModel
*)model withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/*
 GPS 运动状态信息 | GPS motion status information
 */
@interface FBGPSMotionActionModel : NSObject
/**
 运动模式 | Movement mode
*/
@property (nonatomic, assign) FB_MOTIONMODE MotionMode;
/**
 GPS 运动状态 | GPS Motion status
*/
@property (nonatomic, assign) FB_GPS_MOTION_STATE MotionState;
/**
 当前运动总时间，单位秒 | Total current movement time, in seconds
 */
@property (nonatomic, assign) NSInteger totalTime;
@end
```

更多运动模式参考 FB_MOTIONMODE

```
typedef enum {
    FB_SettingStopMotion  = 0,  //停止运动 | Stop motion
    FB_SettingStartMotion = 1,  //开始运动 | Start motion
    FB_SettingPauseMotion = 2,  //暂停运动 | Pause motion
    FB_SettingKeepMotion  = 3,  //继续运动 | Keep motion
}FB_GPS_MOTION_STATE;
```

51、监听设备端 GPS 运动状态变更回调

```
-(void)fbGPS_MotionWatchStatusChangeCallbackWithBlock:(FBGPSMotionWatch
StatusChangeBlock _Nonnull)fbBlock;
// 参考上述参数
```

52、定时心率检测开关设置

```
- (void)fbTimingHeartRateDetectionSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

53、定时血氧检测开关设置

```
- (void)fbTimingBloodOxygenDetectionSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

54、定时精神压力检测开关设置

```
- (void)fbTimingStressDetectionSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

55、获取通话音频开关状态

```
- (void)fbGetCallAudioSwitchWithBlock:(FBGet_AT_ResultCallBackBlock
_Nonnull)fbBlock;
```

56、设置通话音频开关状态

```
- (void)fbSetCallAudioSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

57、获取多媒体音频开关状态

```
-
(void)fbGetMultimediaAudioSwitchWithBlock:(FBGet_AT_ResultCallBackBlock
_Nonnull)fbBlock;
```

58、设置多媒体音频开关状态

```
- (void)fbSetMultimediaAudioSwitchData:(BOOL)switchMode
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/** YES 开 NO 关 | YES ON NO OFF */
BOOL switchMode;
```

59、获取未使用的 日程信息 ID

```
- (void)fbGetUnusedScheduleIDWithBlock:(FBGet_AT_ResultCallBackBlock
_Nonnull)fbBlock;
```

➢ 记录报告同步 API（FBBgCommand）

1、获取设备硬件信息

```
-(void)fbGetHardwareInformationDataWithBlock:(FBGetHardwareInformationBlock _Nonnull)fbBlock;
/*
 设备硬件信息 | Device hardware information
*/
@interface FBDeviceInfoModel : NSObject
/**
 结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
 硬件标志 | Hardware logo
*/
@property (nonatomic, copy) NSString *hardwareIdentifier;
/**
 mac 地址 | MAC address
*/
@property (nonatomic, copy) NSString *mac;
/**
 硬件版本 | Hardware version
*/
@property (nonatomic, copy) NSString *hardWareVersion;
/**
 固件版本 | Firmware version
*/
@property (nonatomic, copy) NSString *firmwareVersion;
/**
 UI 版本 | UI version
*/
@property (nonatomic, copy) NSString *UI_Version;
/**
 协议版本 | Protocol version
*/
@property (nonatomic, copy) NSString *protocolVeriosn;
/**
 设备名称 | Equipment name
*/
@property (nonatomic, copy) NSString *deviceName;
```

```
/**
 设备 ID | Device ID
*/
@property (nonatomic, assign) NSInteger deviceID;
/**
 设备 SN 号 | Equipment Sn number
*/
@property (nonatomic, copy) NSString *deviceSN;
/**
 固件更新日期 | Firmware update date
*/
@property (nonatomic, copy) NSString *firmwareUpdateTime;
/**
 适配号 | Matching number
*/
@property (nonatomic, copy) NSString *fitNumber;
/**
 二维码信息 | QR code information
*/
@property (nonatomic, assign) NSInteger QR_code;
/**
 MAC 二维码版本 | Mac QR code version
*/
@property (nonatomic, assign) NSInteger Mac_QR_code_version;
/**
 显示屏型号 | Display model
*/
@property (nonatomic, assign) NSInteger display_model;
/**
 TP 型号 | TP model
*/
@property (nonatomic, assign) NSInteger TP_model;
/**
 手表表盘形状 | Watch dial shape
 @note  shape            手表表盘形状，0:长方形、1:圆形、2:正方形 | Watch dial shape, 0: rectangle, 1: circle, 2: Square
*/
@property (nonatomic, assign) NSInteger shape;
/**
 手表显示分辨率宽高 | Watch display resolution width and height
 */
@property (nonatomic, assign) CGSize watchDisplaySize;
/**
 表盘缩略图显示分辨率宽高 | Dial thumbnail display resolution width and height
 */
```

```
@property (nonatomic, assign) CGSize dialThumbnailDisplaySize;
/**
 音频库版本 | Audio library version
 */
@property (nonatomic, copy) NSString *audioTimeVersion;
@end
```

@note 该接口请求成功时，内部会自动更新 **FBFirmwareVersionObject** 本地缓存，更多信息请查看 **FBAllConfigObject.firmwareConfig**

2、获取当日实时测量数据

```
-(void)fbGetCurrentDayActivityDataWithBlock:(FBGetCurrentDayActivityDataBlock _Nonnull)fbBlock;
/*
 当日实时测量数据 | Real time measurement data of the day
*/
@interface FBCurrentDataModel : NSObject
/**
 本次数据产生时间点，时间戳 GMT 秒 | Time point of data generation, time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 当前累计步数 | Current cumulative steps
*/
@property (nonatomic, assign) NSInteger currentStep;
/**
 当前累计消耗卡路里（千卡） | Current cumulative calories consumed (kcal)
*/
@property (nonatomic, assign) NSInteger currentCalories;
/**
 当前累计行程（米） | Current cumulative travel (m)
*/
@property (nonatomic, assign) NSInteger currentDistance;
/**
 当前心率（次/分钟） | Current heart rate (times / min)
*/
@property (nonatomic, assign) NSInteger currentHeartRate;
/**
 当前心率等级 | Current heart rate level
```

```
*/
@property (nonatomic, assign) FB_CURRENTHEARTRANGE HeartRateRange;
/**
 当前血氧（%）| Current blood oxygen (%)
*/
@property (nonatomic, assign) NSInteger currentOxy;
/**
 当前血氧等级 | Current blood oxygen level
*/
@property (nonatomic, assign) FB_CURRENTOXYRANGE OxyRange;
/**
 当前电池电量（%）| Current battery level (%)
*/
@property (nonatomic, assign) NSInteger batteryLevel;
/**
 当前收缩血压（高压，mmHg）| Current systolic blood pressure (high pressure, mmHg)
*/
@property (nonatomic, assign) NSInteger currentShrinkBlood;
/**
 当前舒张血压（低压，mmHg）| Current diastolic blood pressure (low pressure, mmHg)
*/
@property (nonatomic, assign) NSInteger currentDiastoleBlood;
/**
 当前累计运动时间（分钟）| Current cumulative movement time (minutes)
*/
@property (nonatomic, assign) NSInteger currentSportTimes;
/**
 当前累计激烈运动时间（分钟）| Current accumulated intense exercise time (minutes)
*/
@property (nonatomic, assign) NSInteger currentSportFierceTimes;
/**
 当前发生久坐累计时间（分钟）| Current accumulated sitting time (minutes)
*/
@property (nonatomic, assign) NSInteger sittingTime;
/**
 当前久坐期间平均步数、步数/小时 | Average steps during current sedentary period, steps / hour
*/
@property (nonatomic, assign) NSInteger sittingStep;
/**
 当前的经度 | Current longitude
*/
@property (nonatomic) float currentLongitude;
/**
 当前的纬度 | Current latitude
```

```
*/
@property (nonatomic) float currentLatitude;
/**
 当天每小时步数曲线，一小时一笔，固定 24 笔 | Steps per hour curve of the day, one transaction per hour, 24 fixed transactions

 @note 第一笔为 0 时~1 时的步数，第二笔为 1 时~2 时，以此类推... | The first stroke is the number of steps from 0 hour to 1 hour, the second stroke is from 1 hour to 2 hours, and so on
*/
@property (nonatomic, strong) NSArray <NSNumber *> *currentStepCurve;
/**
 当前精神压力值 | Current stress value
*/
@property (nonatomic, assign) NSInteger currentStress;
/**
 当前精神压力等级 | Current stress level
*/
@property (nonatomic, assign) FB_CURRENTSTRESSRANGE StressRange;
@end

typedef enum {
    FB_HR_NORMAL           = 0,  //正常的 | Normal
    FB_HR_MODERATE         = 1,  //缓和的 | Moderate
    FB_HR_VIGOROUS         = 2,  //充沛的 | Vigorous
    FB_HR_MAX_HR           = 3,  //心率过快 | The heart rate is too fast
    FB_HR_TAKE_IT_EASY      = 4,  //别紧张 | Take it easy
    FB_HR_WATCH_YOUR_LIMITS = 5,  //注意你的极限 | Watch your limits
    FB_HR_DONT_OVEREXERT    = 6,  //不要用力过猛 | Don't overdo it
}FB_CURRENTHEARTRANGE;

typedef enum {
    FB_OXY_NORMAL   = 0,  //正常 | Normal
    FB_OXY_MILD     = 1,  //轻度缺氧 | Mild hypoxia
    FB_OXY_MODERATE = 2,  //中度缺氧 | Moderate hypoxia
    FB_OXY_SEVERE   = 3,  //重度缺氧 | Severe hypoxia
}FB_CURRENTOXYRANGE;

typedef enum {
    FB_STRESS_RELAX     = 1,  //1-25 放松 | 1-25 Relax
    FB_STRESS_NORMAL    = 2,  //26-50 正常 | 26-50 normal
    FB_STRESS_SECONDARY = 3,  //51-75 中等 | 51-75 Medium
    FB_STRESS_HIGN      = 4   //76-99 偏高 | 76-99 high
}FB_CURRENTSTRESSRANGE;
```

3、获取当前睡眠实时统计报告

```
-(void)fbGetCurrentSleepStatisticsReportDataWithBlock:(FBGetSleepStatisticsReportBlock _Nonnull)fbBlock;
/*
 睡眠统计报告 | Sleep statistics report
*/
@interface FBSleepCaculateReportModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
 本次开始睡觉时间，GMT 秒 | Time to go to bed this time, GMT seconds
*/
@property (nonatomic, assign) NSInteger startSleepTime;
/**
 本次开始睡觉时间，GMT 转年月日时分秒 | Time to go to bed this time, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *startDateTimerStr;
/**
 本次结束睡觉时间，GMT 秒 | The end of the sleep time, GMT seconds
*/
@property (nonatomic, assign) NSInteger endSleepTime;
/**
 本次结束睡觉时间，GMT 转年月日时分秒 | The end of the sleep time, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *endDateTimerStr;
/**
 本次睡眠持续总时间（分钟） | Total sleep duration (minutes)
*/
@property (nonatomic, assign) NSInteger continueSleepTime;
/**
 本次睡眠清醒累计时间（分钟） | Cumulative time of waking up in this sleep (minutes)
*/
@property (nonatomic, assign) NSInteger awakeTime;
```

```
/**
 本次睡眠浅睡累计时间（分钟）| Cumulative time of light sleep (minutes)
*/
@property (nonatomic, assign) NSInteger lightSleepTime;
/**
 本次睡眠深睡累计时间（分钟）| Cumulative time of deep sleep (minutes)
*/
@property (nonatomic, assign) NSInteger deepSleepTime;
/**
 本次睡眠眼动累计时间（分钟）| Cumulative time of eye movement in this sleep (minutes)
*/
@property (nonatomic, assign) NSInteger eyeMoveTime;
/**
 本次睡眠零星小睡累计时间（分钟）| Cumulative time of this sporadic nap (minutes)
*/
@property (nonatomic, assign) NSInteger sporadicNapTime;
/**
 本次睡眠时最大血氧（%）| Maximum blood oxygen during this sleep (%)
*/
@property (nonatomic, assign) NSInteger maxOxy;
/**
 本次睡眠时最小血氧（%）| Minimum blood oxygen during this sleep (%)
*/
@property (nonatomic, assign) NSInteger minOxy;
/**
 本次睡眠时最大心率（次/分钟）| Maximum heart rate during this sleep (times / min)
*/
@property (nonatomic, assign) NSInteger maxHeartRate;
/**
 本次睡眠时最小心率（次/分钟）| Minimum heart rate during this sleep (times / min)
*/
@property (nonatomic, assign) NSInteger minHeartRate;
/**
 本次睡眠时静息心率 | Resting heart rate during sleep
*/
@property (nonatomic, assign) NSInteger restingHeartRate;
@end
```

4、获取当前睡眠实时状态记录

```
-(void)fbGetCurrentSleepStateRecordingDataWithBlock:(FBGetSleepStateRecordingBlock _Nonnull)fbBlock;
/*
 睡眠状态记录 | Sleep state recording
```

```
*/
@interface FBSleepStatusRecordModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 结构体版本，0:不支持眼动，1:支持眼动 | Structure version, 0: eye movement is not supported, 1: eye movement is supported
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
 是否有零星小睡，YES:有零星小睡，NO:无零星小睡 | Whether there are sporadic naps, yes: sporadic naps, No: no sporadic naps
*/
@property (nonatomic, assign) BOOL isNap;
/**
 零星小睡数据的偏移位置长度 | Offset position length of sporadic nap data
*/
@property (nonatomic, assign) NSInteger napDataOffset;
/**
 开始睡眠时间，GMT 秒 | Sleep start time, GMT seconds
*/
@property (nonatomic, assign) NSInteger startSleepTime;
/**
 开始睡眠时间，GMT 转年月日时分秒 | Sleep start time, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *startDateTimerStr;
/**
 睡眠结束时间，GMT 秒 | Sleep end time, GMT seconds
*/
@property (nonatomic, assign) NSInteger endSleepTime;
/**
 睡眠结束时间，GMT 转年月日时分秒 | Sleep end time, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *endDateTimerStr;
/**
 当天深度睡眠时间（分钟） | Deep sleep time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger deepSleepTime;
/**
 当天浅睡时间（分钟） | Light sleep time of the day (minutes)
```

```
*/
@property (nonatomic, assign) NSInteger lightSleepTime;
/**
 当天眼动时间（分钟）| Eye movement time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger eyeMoveTime;
/**
睡眠状态数组有效长度 | Effective length of sleep state array
*/
@property (nonatomic, assign) NSInteger EffectiveLength;
/**
 夜间睡眠状态数组 | Night sleep status array
*/
@property (nonatomic, strong) NSArray <FBSleepStateModel *> *sleepStateArray;
/**
 零星小睡状态数组 | Sporadic nap status array
*/
@property (nonatomic, strong) NSArray <FBSleepStateModel *> *napStateArray;
@end



/*
 睡眠状态详细 | Sleep status details
*/
@interface FBSleepStateModel : NSObject
/**
 睡眠状态起始时间戳 GMT 秒 | Sleep state start timestamp GMT seconds
*/
@property (nonatomic, assign) NSInteger startStatusGMT;
/**
 起始 GMT 转年月日时分秒 | From GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *startDateTimeStr;
/**
 睡眠状态结束时间戳 GMT 秒 | Sleep state end timestamp GMT seconds
*/
@property (nonatomic, assign) NSInteger endStatusGMT;
/**
 结束 GMT 转年月日时分秒 | End GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *endDateTimeStr;
/**
 睡眠状态 | Sleep state
*/
```

```
@property (nonatomic, assign) FB_SLEEPSTATE SleepStatus;
/**
 持续睡眠时间（分钟）| Duration of sleep (minutes)
*/
@property (nonatomic, assign) NSInteger durationSleepTime;
@end

typedef enum {
    Awake_state   = 0,  //清醒状态 | Awake state
    Shallow_sleep = 1,  //浅层睡眠 | Shallow sleep
    Deep_sleep    = 2,  //深层睡眠 | Deep sleep

    Eye_move      = 3,  //眼动状态（结构体版本不等于 0 时才有此类型）| Eye move (This type is only available when the structure version is not equal to 0)
}FB_SLEEPSTATE;
```

5、获取每日活动统计报告

```
- (void)fbGetDailyActivityDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetDailyActivityDataBlock _Nonnull)fbBlock;
/*
 每日活动统计报告 | Daily activity statistics report
*/
@interface FBDayActivityModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
 当天的累计步数 | Cumulative steps of the day
*/
@property (nonatomic, assign) NSInteger totalSteps;
/**
 当天消耗的卡路里（千卡）| Calories consumed that day (kcal)
*/
```

```objc
@property (nonatomic, assign) NSInteger totalCalories;


/**
 当天的累计行程（米）| Cumulative itinerary of the day (m)
*/
@property (nonatomic, assign) NSInteger totalDistance;
/**
 当天的平均心率（次/分钟）| Average heart rate of the day (times / min)
*/
@property (nonatomic, assign) NSInteger avgHeartRate;
/**
 当天最高心率（次/分钟）| Maximum heart rate of the day (times / min)
*/
@property (nonatomic, assign) NSInteger maxHeartRate;
/**
 当天最低心率（次/分钟）| Lowest heart rate of the day (times / min)
*/
@property (nonatomic, assign) NSInteger minHeartRate;
/**
 当天平均血氧（%）| Average blood oxygen of the day (%)
*/
@property (nonatomic, assign) NSInteger avgOxy;
/**
 当天累计运动时间（分钟）| Accumulated exercise time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger totalSportTime;
/**
 当天激烈运动时间（分钟）| Intense exercise time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger voilentSportTime;
/**
 当天深度睡眠时间（分钟）| Deep sleep time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger deepSleeTime;
/**
 当天浅睡眠时间（分钟）| Light sleep time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger lightSleepTime;
/**
 当天睡眠眼动时间（分钟）| Sleep eye movement time of the day (minutes)
*/
@property (nonatomic, assign) NSInteger eyeMoveTime;
/**
 当天最高血压（mmHg）| Highest blood pressure of the day (mmHg)
```

```
*/
@property (nonatomic, assign) NSInteger maxBlood;
/**
 当天最低血压（mmHg）| Lowest blood pressure of the day (mmHg)
*/
@property (nonatomic, assign) NSInteger minBlood;
/**
 当天最高精神压力值 | Maximum stress value of the day
*/
@property (nonatomic, assign) NSInteger maximumStress;
/**
 当天最高精神压力等级 | Maximum stress level of the day
*/
@property (nonatomic, assign) FB_CURRENTSTRESSRANGE StressRange;
@end
```

6、获取整点活动统计报告

```
- (void)fbGetHourlyActivityDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetHourlyActivityDataBlock_Nonnull)fbBlock;
/*
 整点活动统计报告 | Statistical report of on-time activities
*/
@interface FBHourReportModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
 到此刻为止的累计计步数 | Cumulative steps up to now
*/
@property (nonatomic, assign) NSInteger hourStep;
/**
 到此刻为止的累计行走距离（米） | Accumulated walking distance up to now (m)
*/
```

```
@property (nonatomic, assign) NSInteger hourdDistance;
/**
 到此刻为止的累计消耗卡路里（千卡）| Cumulative calories burned so far (kcal)
*/
@property (nonatomic, assign) NSInteger hourCalories;
@end
```

7、获取睡眠统计报告

```
- (void)fbGetSleepStatisticsReportDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime
withBlock:(FBGetSleepStatisticsReportBlock _Nonnull)fbBlock;
// 参考上述睡眠实时统计报告参数
```

8、获取睡眠状态记录

```
- (void)fbGetSleepStateRecordingDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetSleepStateRecordingBlock
_Nonnull)fbBlock;
// 参考上述睡眠实时状态记录参数
```

9、获取运动记录列表

```
- (void)fbGetMotionRecordListDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetMotionRecordListBlock
_Nonnull)fbBlock;
/*
 运动记录列表 | Sports record list
*/
@interface FBSportRecordModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
```

```
开始运动时间，GMT 秒 | Start time of exercise, GMT seconds
*/
@property (nonatomic, assign) NSInteger startSportTime;
/**
开始运动时间，GMT 转年月日时分秒 | Start time of exercise, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *startDateTimerStr;
/**
结束运动时间，GMT 秒 | End exercise time, GMT seconds
*/
@property (nonatomic, assign) NSInteger endSportTime;
/**
结束运动时间，GMT 转年月日时分秒 | End exercise time, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *endDateTimerStr;
/**
运动模式 | Motion mode
*/
@property (nonatomic, assign) FB_MOTIONMODE MotionMode;
@end
```

10、获取运动统计报告

```
- (void)fbGetSportsDataReportDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetSportsDataReportBlock_Nonnull)fbBlock;
/*
 运动统计报告 | Sports statistics report
*/
@interface FBSportCaculateModel : NSObject
/**
时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
结构体版本 | Structure version
*/
@property (nonatomic, assign) NSInteger structVersion;
/**
```

```
运动开始时间戳，作为每笔运动的唯一识别 id， GMT 秒 | The movement start time stamp is used as the unique identification ID of each movement, GMT seconds.
*/
@property (nonatomic, assign) NSInteger startSportTime;
/**
运动开始时间，GMT 转年月日时分秒 | Start time of exercise, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *startDateTimerStr;
/**
运动结束时间戳，GMT 秒 | Movement end timestamp, GMT seconds
*/
@property (nonatomic, assign) NSInteger endSportTime;
/**
运动结束时间，GMT 转年月日时分秒 | End time of exercise, GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *endDateTimerStr;
/**
运动总时间（秒） | Total exercise time (seconds)
*/
@property (nonatomic, assign) NSInteger totalSportTime;
/**
运动总步数 | Total movement steps
*/
@property (nonatomic, assign) NSInteger totalSteps;
/**
运动总卡路里（千卡） | Total exercise calories (kcal)
*/
@property (nonatomic, assign) NSInteger totalCalories;
/**
运动总距离（单位米，通过计步估算） | Total distance of movement (in meters, estimated by steps)
*/
@property (nonatomic, assign) NSInteger totalDistance;
/**
本次运动轨迹运动距离（单位米，通过 gps 定位 计算） | The movement distance of this movement track (unit: m, calculated by GPS positioning)
*/
@property (nonatomic, assign) NSInteger gpsDistance;
/**
运动模式 | Motion mode
*/
@property (nonatomic, assign) FB_MOTIONMODE MotionMode;
/**
本次运动最大心率（次/分钟） | Maximum heart rate of this exercise (times / min)
*/
```

```objc
@property (nonatomic, assign) NSInteger maxHeartRate;
/**
 本次运动最小心率（次/分钟）| Minimum heart rate of this exercise (times / min)
*/
@property (nonatomic, assign) NSInteger minHeartRate;
/**
 本次运动平均心率，运动结束时计算，心率和/ 记录次数（次/分钟）| Average heart rate of the exercise, calculated at the end of the exercise, heart rate and / or record times (times / minute)
*/
@property (nonatomic, assign) NSInteger avgHeartRate;
/**
 本次运动最大步频（步/分钟）| Maximum stride frequency (step / min)
*/
@property (nonatomic, assign) NSInteger maxStride;
/**
 本次运动平均步频 = 步频和/记录次数（步/分钟）| Average stride frequency = stride frequency and / or recording times (step / minute)
*/
@property (nonatomic, assign) NSInteger avgStride;
/**
 运动次数，中途休息次数 | Number of exercises, number of breaks
*/
@property (nonatomic, assign) NSInteger breakTimes;
/**
 中断 UTC 记录，同时用于统计运动总时间 | The UTC record is interrupted and used to count the total movement time at the same time
*/
@property (nonatomic, strong) NSArray <FBSportPauseModel *> *sportPauseArray;
/**
 本次运动最大速度(单位:米/秒) | Maximum speed of this movement (unit: M / s)
*/
@property (nonatomic, assign) NSInteger maxSpeed;
/**
 本次运动平均速度 = 距离/用时（米/秒） | Average speed of this movement = distance / time (M / s)
*/
@property (nonatomic, assign) NSInteger avgSpeed;
/**
 本次无轨迹运动平均配速（秒/公里） | Average speed of this trackless movement (s / km)
*/
@property (nonatomic, assign) NSInteger noTrackAvgSpeed;
/**
 本次有轨迹运动配速（秒/公里） | This time, there is track movement speed (s / km)
*/
@property (nonatomic, assign) NSInteger trackAvgSpeed;
/**
 重复运动的周期数(来回次数，圈数)(单位：圈) | Number of cycles of repeated motion (number of cycles, number of turns) (unit: turns)
```

```
*/
@property (nonatomic, assign) NSInteger sportRepeatCount;
/**
 摆臂次数，划水次数(单位:次)｜Arm swing times, stroke times (unit: Times)
*/
@property (nonatomic, assign) NSInteger armSwingTimes;
/**
 热身运动时间，单位分钟｜Warm up exercise time in minutes
*/
@property (nonatomic, assign) NSInteger heartRate_level_1;
/**
 燃脂运动时间，单位分钟｜Fat burning exercise time in minutes
*/
@property (nonatomic, assign) NSInteger heartRate_level_2;
/**
 有氧运动时间，单位分钟｜Aerobic exercise time in minutes
*/
@property (nonatomic, assign) NSInteger heartRate_level_3;
/**
 无氧运动时间，单位分钟｜Anaerobic exercise time in minutes
*/
@property (nonatomic, assign) NSInteger heartRate_level_4;
/**
 极限运动时间，单位分钟｜Extreme exercise time in minutes
*/
@property (nonatomic, assign) NSInteger heartRate_level_5;
@end
```

11、获取心率记录

```
- (void)fbGetHeartRateRecordDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetHeartRateRecordBlock _Nonnull)fbBlock;
/*
 类型记录/报告｜Type record / Report
*/
@interface FBTypeRecordModel : NSObject
/**
 第一条记录(结构体)的形成时间戳 GMT 秒｜Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒｜GMT to YYYY-MM-dd HH:mm:ss
*/
```

```objc
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 本次运动的开始时间戳 GMT 秒（仅记录类型为：运动详情记录、运动定位记录 时 有值）| The start time stamp of this movement is GMT seconds (only the record types are: motion detail record and motion positioning record)
*/
@property (nonatomic, assign) NSInteger sportTimeStamp;
/**
 记录生成周期（记录类型为心率/计步/血氧/血压记录时，单位分钟；为运动详情/运动定位记录时，单位秒）| Record generation cycle (record type: heart rate / step / blood oxygen / blood pressure record, unit: minute; record type: movement details / movement positioning record, unit: Second)
*/
@property (nonatomic, assign) NSInteger createTimes;
/**
 有效记录条数 | Number of effective records
*/
@property (nonatomic, assign) NSInteger EffectiveRecord;
/**
 单条记录长度 | Length of single record
*/
@property (nonatomic, assign) NSInteger recordLength;
/**
 记录类型 | Record type
*/
@property (nonatomic, assign) FB_RECORDTYPE RecordType;
/**
 类型记录数组 | Type record array
*/
@property (nonatomic, strong) NSArray <FBRecordDetailsModel *> *recordArray;
@end


/*
 * 记录类型 | Record type
 */
typedef enum {
    FB_HeartRecord     = 0,  //心率记录 | Heart rate recording
    FB_StepRecord      = 1,  //计步记录 | Step count record
    FB_BloodOxyRecord  = 2,  //血氧记录 | Blood oxygen recording
    FB_BloodPreRecord  = 3,  //血压记录 | Blood pressure recording
    FB_SportsRecord    = 4,  //运动详情记录 | Sports detail record
    FB_MotionGpsRecord = 5,  //运动定位记录 | Motion location record
    FB_HFHeartRecord   =6,  //运动高频心率记录(1 秒 1 次) | Sports high-frequency heart rate recording (1 time per second)
    FB_StressRecord    = 7,  //精神压力记录 | Stress Record
}FB_RECORDTYPE;
```

```objc
/*
 类型记录数组详情（具体参考枚举值 FB_RECORDTYPE）| Type record array details (refer to enumeration value FB_RECORDTYPE for details)
*/
@interface FBRecordDetailsModel : NSObject
#pragma mark - 以下值，通用，有值 | The following values, general, have values
/** 记录形成时间戳 GMT 秒 | Record the formation time stamp GMT seconds */
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/** GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss */
@property (nonatomic, copy) NSString *dateTimeStr;

#pragma mark - 当 FB_RECORDTYPE 为 FB_HeartRecord、FB_HFHeartRecord 时（心率记录、运动高频心率记录(1 秒 1 次)），以下有值 | When FB_RECORDTYPE is FB_HeartRecord, FB_HFHeartRecord (heart rate record, exercise high-frequency heart rate record (1 time per second)), the following values
/** 心率值 | Heart rate value */
@property (nonatomic, assign) NSInteger hr;

#pragma mark - 当 FB_RECORDTYPE 为 FB_StepRecord 时（计步记录），以下有值 | When FB_RECORDTYPE is FB_StepRecord (step counting record), the following values
/** 计步数累加值 | Accumulated value of step count */
@property (nonatomic, assign) NSInteger step;

#pragma mark - 当 FB_RECORDTYPE 为 FB_BloodOxyRecord 时（血氧记录），以下有值 | When FB_RECORDTYPE is FB_BloodOxyRecord (blood oxygen record), the following values
/** 血氧值（%） | Blood oxygen value (%) */
@property (nonatomic, assign) NSInteger Sp02;

#pragma mark - 当 FB_RECORDTYPE 为 FB_BloodPreRecord 时（血压记录），以下有值 | When FB_RECORDTYPE is FB_BloodPreRecord (blood pressure record), the following values
/** 收缩压（高压，mmHg） | Systolic blood pressure (high pressure, mmHg) */
@property (nonatomic, assign) NSInteger pb_max;
/** 舒张压（低压，mmHg） | Diastolic blood pressure (low pressure, mmHg) */
@property (nonatomic, assign) NSInteger pb_min;

#pragma mark - 当 FB_RECORDTYPE 为 FB_StressRecord 时（精神压力记录），以下有值 | When FB_RECORDTYPE is FB_StressRecord (mental stress record), the following values
/** 精神压力值 | Mental stress value */
@property (nonatomic, assign) NSInteger stress;
/** 精神压力等级 | Mental stress level */
@property (nonatomic, assign) FB_CURRENTSTRESSRANGE StressRange;

#pragma mark - 当 FB_RECORDTYPE 为 FB_SportsRecord 时（运动详情记录），以下有值 | When FB_RECORDTYPE is FB_SportsRecord (sports details record), the following values
```

```objc
/** 实时配速（秒/千米）| Real time pace (SEC / km) */
@property (nonatomic, assign) NSInteger pace;
/** 一分钟内消耗的卡路里值（千卡）| Calories consumed in one minute (kcal) */
@property (nonatomic, assign) NSInteger calories;
/** 一分钟内的步数（实时步频，步/分钟）| Steps in one minute (real time step frequency, step / minute) */
@property (nonatomic, assign) NSInteger stepFrequency;
/** 运动中的实时距离（米）| Real time distance in motion (m) */
@property (nonatomic, assign) NSInteger distance;
/** 实时心率（次/分钟）| Real time heart rate (times / min) */
@property (nonatomic, assign) NSInteger heartRate;
/** 实时体力，0~100 | Real time physical strength, 0-100 */
@property (nonatomic, assign) NSInteger stamina;
/** 运动状态。NO 正常，YES 暂停 | Motion state. NO normal, YES pause */
@property (nonatomic, assign) BOOL isSuspend;

#pragma mark - 当 FB_RECORDTYPE 为 FB_MotionGpsRecord 时（运动定位记录），以下有值 | When FB_RECORDTYPE is FB_MotionGpsRecord (motion positioning record), the following values
/** 纬度 | Real time distance in motion */
@property (nonatomic) float latitude;
/** 经度 | Real time heart rate */
@property (nonatomic) float longitude;
/** GPS 速度（米/秒）| GPS speed (M / s) */
@property (nonatomic, assign) NSInteger speed;
/** 状态。NO 正常，YES 暂停 | Status. NO normal, YES pause */
@property (nonatomic, assign) BOOL gpsIsSuspend;
@end
```

12、获取计步记录

```objc
- (void)fbGetStepCountRecordDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetStepCountRecordBlock _Nonnull)fbBlock;
// 参考上述记录参数
```

13、获取血氧记录

```objc
- (void)fbGetBloodOxygenRecordDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetBloodOxygenRecordBlock _Nonnull)fbBlock;
// 参考上述记录参数
```

14、获取血压记录

```
- (void)fbGetBloodPressureRecordsDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetBloodPressureRecordsBlock
_Nonnull)fbBlock;
// 参考上述记录参数
```

15、获取精神压力记录

```
- (void)fbGetStressRecordsDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetStressRecordsBlock
_Nonnull)fbBlock;
// 参考上述记录参数
```

16、获取运动详情记录

```
- (void)fbGetExerciseDetailsDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetExerciseDetailsBlock
_Nonnull)fbBlock;
// 参考上述记录参数
```

17、获取 运动统计报告+运动详情纪录

```
-
(void)fbGetSportsStatisticsDetailsReportsWithStartTime:(NSInteger)start
Time forEndTime:(NSInteger)endTime
withBlock:(FBGetSportsStatisticsDetailsRecordBlock _Nonnull)fbBlock;
// 参考上述记录参数
```

18、获取运动定位记录

```
- (void)fbGetMotionLocationRecordDataStartTime:(NSInteger)startTime
forEndTime:(NSInteger)endTime withBlock:(FBGetMotionLocationRecordBlock
_Nonnull)fbBlock;
// 参考上述记录参数
```

19、获取运动高频心率记录(1 秒 1 次)

```
-
(void)fbExerciseHighFrequencyHeartRateRecordsDataStartTime:(NSInteger)s
tartTime forEndTime:(NSInteger)endTime
withBlock:(FBGetExerciseHFHRRecordsBlock _Nonnull)fbBlock;
```

```
// 参考上述记录参数
```

20、获取手动测量数据记录

```
- (void)fbGetManualMeasurementDataStartTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetManualMeasureDataBlock_Nonnull)fbBlock;
/*
 手动测量数据记录 | Manual measurement data record
*/
@interface FBManualMeasureDataModel : NSObject
/**
 时间戳 GMT 秒 | Time stamp GMT seconds
*/
@property (nonatomic, assign) NSInteger GMTtimeInterval;
/**
 GMT 转年月日时分秒 | GMT to YYYY-MM-dd HH:mm:ss
*/
@property (nonatomic, copy) NSString *dateTimeStr;
/**
 心率值 | Heart rate value
*/
@property (nonatomic, assign) NSInteger hr;
/**
 血氧值（%） | Blood oxygen value (%)
*/
@property (nonatomic, assign) NSInteger Sp02;
/**
 收缩压（高压，mmHg） | Systolic blood pressure (high pressure, mmHg)
 */
@property (nonatomic, assign) NSInteger pb_max;
/**
 舒张压（低压，mmHg） | Diastolic blood pressure (low pressure, mmHg)
 */
@property (nonatomic, assign) NSInteger pb_min;
/**
 精神压力值 | Mental stress value
 */
@property (nonatomic, assign) NSInteger stress;

/**
 精神压力等级 | Mental stress level
 */
@property (nonatomic, assign) FB_CURRENTSTRESSRANGE StressRange;
```

```
@end
```

21、获取指定的记录和报告

```
-(void)fbGetSpecialRecordsAndReportsDataWithType:(FB_MULTIPLERECORDREPORTS)recordTypes startTime:(NSInteger)startTime forEndTime:(NSInteger)endTime withBlock:(FBGetSpecialRecordsAndReportsBlock _Nonnull)fbBlock;
typedef enum {
    FB_CurrentDayActivityData          = 1<<0,   //当日实时测量数据 | Real time measurement data of the day
    FB_HeartRateRecording              = 1<<1,   //心率记录 | Heart rate recording
    FB_StepCountRecord                 = 1<<2,   //计步记录 | Step counting record
    FB_BloodOxygenRecording            = 1<<3,   //血氧记录 | Blood oxygen record
    FB_BloodPressureRecording          = 1<<4,   //血压记录 | Blood pressure record
    FB_HFHeartRateRecording            = 1<<5,   //运动高频心率记录(1秒1次) | Sports high-frequency heart rate recording (1 time per second)
    FB_StressRecording                 = 1<<6,   //精神压力记录 | Stress Record
    FB_SportsDetailsRecord             = 1<<7,   //运动详情记录 | Sports detail record
    FB_SportsPositioningRecord         = 1<<8,   //运动定位记录 | Sports positioning record
    FB_DailyActivityReport             = 1<<9,   //每日活动报告 | Daily activity report
    FB_OnHourActivityReport            = 1<<10,  //整点活动报告 | On hour activity report
    FB_SleepStatisticsReport           = 1<<11,  //睡眠统计报告 | Sleep statistics report
    FB_SleepStateRecording             = 1<<12,  //睡眠状态记录 | Sleep state recording
    FB_CurrentSleepStatisticsReport    = 1<<13,  //当前睡眠实时统计报告 | Current sleep real time statistics report
    FB_CurrentSleepStateRecording      = 1<<14,  //当前睡眠实时状态记录 | Current sleep real time status record
    FB_SportsRecordList                = 1<<15,  //运动记录列表 | Sports record list
    FB_SportsStatisticsReport          = 1<<16,  //运动统计报告 | Sports statistics report
    FB_Sports_Statistics_Details_Report = 1<<17,  //运动统计报告+运动详情纪录 | Sports statistics report + sports details record
    FB_ManualMeasurementData           = 1<<18,  //手动测量数据 | Manual measurement data
}FB_MULTIPLERECORDREPORTS;

// 其他参数参考对应的数据模型
```

22、获取个人用户信息

```
- (void)fbGetPersonalUserInforWithBlock:(FBGetPersonalUserInforBlock _Nonnull)fbBlock;
/*
 用户个人信息 | User personal information
*/
@interface FBUserInforModel : NSObject
```

```objc
/**
 用户 ID（大于 0，小于 0xFFFFFFFF）｜User ID (greater than 0, less than 0xFFFFFF)
*/
@property (nonatomic, assign) NSInteger userId;
/**
 用户昵称（长度小于或等于 31 个字节，用户昵称超出最大长度，自动截取）｜User nickname (the length is less than or equal to 31 bytes, and the user nickname exceeds the maximum length, automatically intercepted)
*/
@property (nonatomic, copy) NSString *userNickname;
/**
 用户身高（单位 cm，大于 100，小于 250）｜User's height (in cm, greater than 100, less than 250)
*/
@property (nonatomic, assign) NSInteger userHeight;
/**
 用户体重（单位 kg， 大于 30， 小于 250）｜User's weight (in kg, more than 30, less than 250)
*/
@property (nonatomic, assign) NSInteger userWeight;
/**
 时区偏移时间（分钟）｜Time zone offset time (minutes)
*/
@property (nonatomic, assign) NSInteger userTimeZoneMinute;
/**
 用户性别｜User gender
*/
@property (nonatomic, assign) FB_USERGENDER UserGender;
/**
 用户年龄（大于 5 岁，小于 130 岁）｜User age (over 5, under 130)
*/
@property (nonatomic, assign) NSInteger userAge;
/**
 用户步幅（单位 cm）｜User stride (in cm)
*/
@property (nonatomic, assign) NSInteger userStride;
/**
 女性生理周期信息｜Female physiological cycle information
 */
@property (nonatomic, strong) FBFemalePhysiologyModel *physiologyModel;
@end

typedef enum {
    FB_UserMale   = 0,  //男性｜Male
    FB_UserFemale = 1,  //女性｜Female
}FB_USERGENDER;
```

23、设置用户个人信息

```
- (void)fbSetPersonalUserInforWithUserModel:(FBUserInforModel *_Nonnull)userModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

24、获取记事提醒/闹铃信息

```
- (void)fbGetClockInforWithBlock:(FBGetClockInforBlock _Nonnull)fbBlock;
/*
 记事提醒/闹钟信息 | Reminder / alarm clock
*/
@interface FBAlarmClockModel : NSObject
#pragma mark 以下值，通用，必传 | The following values, general, must be passed
/**
 序号 ID（0，1，2，3，4）最多 5 个闹钟 | Serial number ID (0, 1, 2, 3, 4) up to 5 alarm clocks
*/
@property (nonatomic, assign) NSInteger clockID;
/**
 闹铃类别：备忘提醒（年月日小时分钟有效），定时闹钟（仅小时分钟有效） | Alarm category: reminder（valid for hours and minutes), Time alarm clock（only hours and minutes）
*/
@property (nonatomic, assign) FB_ALARMCATEGORY clockCategory;
/**
 使能开关 NO:关 YES:开（默认 YES） | Enable switch NO: off YES: on (default YES)
*/
@property (nonatomic, assign) BOOL clockEnableSwitch;
/**
 稍后提醒开关 NO:关 YES:开（默认 NO） | Remind switch later NO: OFF YES: ON (default NO)
*/
@property (nonatomic, assign) BOOL remindLater;
/**
 描述，长度小于等于 23 个字节 | Description, the length of description is less than or equal to 23 bytes
*/
@property (nonatomic, copy) NSString *clockDescribe;

#pragma mark 当 FB_ALARMCATEGORY==FB_Reminders 时，为备忘提醒，以下值，必传 | When FB_ALARMCATEGORY==FB_Reminders, for reminders, the following values must be passed
/**
 年月日小时分钟，格式：YYYY-MM-dd HH:mm（当闹铃类别为 FB_Reminders:备忘提醒，必传；为 FB_AlarmClock:定时闹钟可不传） | Month, year, day, hour and minute, format: YYYY-MM-dd HH:mm (When the alarm type is FB_Reminders: reminder, it must be sent; when it is FB_AlarmClock: fixed time alarm, it can not be sent)
*/
@property (nonatomic, copy) NSString *clockYMDHm;
```

```
#pragma mark 当 FB_ALARMCATEGORY==FB_AlarmClock 时，为定时闹钟，以下值，必传 | When FB_
ALARMCATEGORY==FB_AlarmClock, for alarm clock, the following values must be passed
/**
 重复性，YES:周期有效，NO:一次有效 | Repeatability, YES: cycle effective, NO: once effective
*/
@property (nonatomic, assign) BOOL isRepeat;
/**
 星期选中标记（星期日、星期一、星期二、星期三、星期四、星期五、星期六；必须设置固定七个数据的数组，传 0「未选中」或 1「选中」） | Week check mark (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday; fixed array of seven data must be set, transfer 0 (unselected) or 1 (selected))
*/
@property (nonatomic, strong) NSArray <NSNumber *> *clockRepeatArray;
/**
 小时分钟,格式:HH:mm(当闹铃类别为 FB_AlarmClock:定时闹钟,必传;为FB_Reminders:备忘提醒可不传) | Hours and minutes, format: HH: mm (When the alarm type is FB_AlarmClock: timed alarm, it must be sent; when it is FB_Reminders: reminder, it can not be sent)
*/
@property (nonatomic, copy) NSString *clockHm;
@end

typedef enum {
    FB_Reminders  = 0,  //备忘提醒 | Reminders
    FB_AlarmClock = 1,  //定时闹钟 | Alarm clock
}FB_ALARMCATEGORY;
```

25、设置记事提醒/闹铃信息

```
- (void)fbSetClockInforWithClockModel:(FBAlarmClockModel *
_Nonnull)clockModel withRemoved:(BOOL)isRemoved
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

26、获取消息推送开关信息

```
- (void)fbGetMessagePushSwitchWithBlock:(FBGetMessagePushSwitchBlock
_Nonnull)fbBlock;
/*
 消息推送开关信息 | Message push switch information
*/
@interface FBMessageModel : NSObject
/**
 没有，其他类型 | No, other types
*/
@property (nonatomic, assign) BOOL none;
```

```
/**
 电话 | Telephone
*/
@property (nonatomic, assign) BOOL telephone;
/**
 短信 | SMS
*/
@property (nonatomic, assign) BOOL sms;
/**
 微信 | WeChat
*/
@property (nonatomic, assign) BOOL weChat;
/**
 QQ
*/
@property (nonatomic, assign) BOOL qq;
/**
 脸书 | Facebook
*/
@property (nonatomic, assign) BOOL facebook;
/**
 推特 | Twitter
*/
@property (nonatomic, assign) BOOL twitter;
/**
 领英 | LinkedIn
*/
@property (nonatomic, assign) BOOL linkedIn;
/**
 Whatsapp
*/
@property (nonatomic, assign) BOOL whatsapp;
/**
 Line
*/
@property (nonatomic, assign) BOOL line;
/**
 照片墙 | Instagram
*/
@property (nonatomic, assign) BOOL instagram;
/**
 色拉布 | Snapchat
*/
@property (nonatomic, assign) BOOL snapchat;
```

```
/**
 Skype
*/
@property (nonatomic, assign) BOOL skype;
/**
 谷歌邮箱 | Gmail
*/
@property (nonatomic, assign) BOOL gmail;
/**
 Outlook
*/
@property (nonatomic, assign) BOOL outlook;
/**
 Messenger
*/
@property (nonatomic, assign) BOOL messenger;
/**
 Viber
*/
@property (nonatomic, assign) BOOL viber;
/**
 Googletalk
*/
@property (nonatomic, assign) BOOL googletalk;
/**
 Vkontakte
*/
@property (nonatomic, assign) BOOL vkontakte;
/**
 Imo
*/
@property (nonatomic, assign) BOOL imo;
/**
 Imobeta
*/
@property (nonatomic, assign) BOOL imobeta;
/**
 Imolite
*/
@property (nonatomic, assign) BOOL imolite;
/**
 Chatapp
*/
@property (nonatomic, assign) BOOL chatapp;
```

```
/**
 Kik
*/
@property (nonatomic, assign) BOOL kik;
/**
 Skred
*/
@property (nonatomic, assign) BOOL skred;
/**
 Telegramx
*/
@property (nonatomic, assign) BOOL telegramx;
/**
 Beechat
*/
@property (nonatomic, assign) BOOL beechat;
/**
 Teamtalk
*/
@property (nonatomic, assign) BOOL teamtalk;
/**
 Kakao
*/
@property (nonatomic, assign) BOOL kakao;
/**
 Ftalk
*/
@property (nonatomic, assign) BOOL ftalk;
/**
 Rimet
*/
@property (nonatomic, assign) BOOL rimet;
/**
 Wework
*/
@property (nonatomic, assign) BOOL wework;
/**
 红包 | Red envelope/Hong Bao
*/
@property (nonatomic, assign) BOOL HongBao;
/**
 Missedcall
*/
@property (nonatomic, assign) BOOL missedcall;
```

```
/**
 Calendar
*/
@property (nonatomic, assign) BOOL calendar;
/**
 Applemusic
*/
@property (nonatomic, assign) BOOL applemusic;
/**
 Googlemaps
*/
@property (nonatomic, assign) BOOL googlemaps;
/**
 Likee
*/
@property (nonatomic, assign) BOOL likee;
/**
 Messages
*/
@property (nonatomic, assign) BOOL messages;
/**
 Mono
*/
@property (nonatomic, assign) BOOL mono;
/**
 Odnoklassniki
*/
@property (nonatomic, assign) BOOL odnoklassniki;
/**
 Privat
*/
@property (nonatomic, assign) BOOL privat;
/**
 Youtube
*/
@property (nonatomic, assign) BOOL youtube;
/**
 Youtubemusic
*/
@property (nonatomic, assign) BOOL youtubemusic;
/**
 Zoom
*/
@property (nonatomic, assign) BOOL zoom;
```

```
/**
 Telegram
*/
@property (nonatomic, assign) BOOL telegram;
/**
 Tiktok
*/
@property (nonatomic, assign) BOOL tiktok;
/**
 Pinterest
*/
@property (nonatomic, assign) BOOL pinterest;
/**
 总开关 | Maste rSwitch
*/
@property (nonatomic, assign) BOOL masterSwitch;
@end
// 更多类型请查看 SDK 内 FBMacro.h 文件
```

27、设置消息推送开关信息

```
- (void)fbSetMessagePushSwitchWithData:(FBMessageModel *)messageModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

28、获取久坐提醒信息

```
- (void)fbGetLongSitInforWithBlock:(FBGetLongSitInforBlock _Nonnull)fbBlock;
/*
 久坐提醒信息 | Sedentary reminder
*/
@interface FBLongSitModel : NSObject
/**
 久坐提醒开关 NO:关闭 YES:打开（默认 YES） | Sedentary reminder switch NO: off YES: on (Default: YES)
*/
@property (nonatomic, assign) BOOL enable;
/**
 检测起始时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间）（默认 480，即 08:00） | Detection start time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time) (the default is 480, i.e. 08:00)
*/
@property (nonatomic, assign) NSInteger startTime;
/**
```

```
检测结束时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间）（默认 1080，即 18:00）| Detection end time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time) (the default is 1080, i.e. 18:00)
*/
@property (nonatomic, assign) NSInteger endTime;
/**
久坐持续时间检测时间（分钟），在这个时间内步数不达标，进行久坐提醒（默认 45）| The detection time of sedentary duration (minutes). If the steps are not up to standard within this time, the sedentary reminder will be given (Default: 45)
*/
@property (nonatomic, assign) NSInteger continueTime;
/**
目标步数，在持续时间内低于这个值，进行久坐提醒（默认 100）| If the target step number is lower than this value in the duration, the sedentary reminder will be given (Default: 100)
*/
@property (nonatomic, assign) NSInteger targetSteps;
@end
```

29、设置久坐提醒信息

```
- (void)fbSetLongSitInforWithModel:(FBLongSitModel *_Nonnull)longSitModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

30、获取心率等级判定信息

```
- (void)fbGetHeartRateInforWithBlock:(FBGetHeartRateInforBlock _Nonnull)fbBlock;
/*
 心率等级判定信息 | Heart rate level determination information
*/
@interface FBHeartRateRatingModel : NSObject
/**
超过这个百分比，认定为 mix_hr / If the percentage exceeds this, it will be regarded as mix_hr
*/
@property (nonatomic, assign) NSInteger min_hr;
/**
超过这个百分比，认定为 moderate | If the percentage exceeds this, it will be regarded as moderate
*/
@property (nonatomic, assign) NSInteger moderate;
/**
超过这个百分比，认定为 vigorous | If the percentage exceeds this, it will be regarded as vigorous
*/
@property (nonatomic, assign) NSInteger vigorous;
/**
```

```
超过这个百分比，认定为 max_hr | If it exceeds this percentage, it is regarded as max_hr
*/
@property (nonatomic, assign) NSInteger max_hr;
/**
最高心率值 | Maximum heart rate
*/
@property (nonatomic, assign) NSInteger heigt_hr;
/**
只有心率值在这个时间宽度都在某个级别以 上，才确定新等级 | Only when the heart rate value is above a certain level in this time width can the new level be determined
*/
@property (nonatomic, assign) NSInteger other_hr;
@end
```

31、设置心率等级判定信息

```
- (void)fbSetHeartRateInforWithModel:(FBHeartRateRatingModel *_Nonnull)heartAlgoModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

32、获取喝水提醒信息

```
- (void)fbGetDrinkWaterWithBlock:(FBGetDrinkWaterBlock _Nonnull)fbBlock;
/*
 喝水提醒信息 | Water drinking reminder information
*/
@interface FBWaterClockModel : NSObject
/**
提醒起始时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间）（默认 480，即 08:00） | Reminder start time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time) (the default is 480, i.e. 08:00)
*/
@property (nonatomic, assign) NSInteger startTime;
/**
提醒结束时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间）（默认 1080，即 18:00） | Reminder end time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time) (the default is 1080, i.e. 18:00)
*/
@property (nonatomic, assign) NSInteger endTime;
/**
提醒周期，如果为 0 只提醒一次（默认 60） | Reminder cycle, if it is 0, only remind once (Default: 60)
*/
@property (nonatomic, assign) NSInteger repeatCount;
```

```
/**
 提醒开关，NO:关闭  YES:打开（默认 YES） | Reminder switch, NO: off, YES: on (Default: YES)
*/
@property (nonatomic, assign) BOOL alterSwitch;
@end
```

33、设置喝水提醒信息

```
- (void)fbSetDrinkWaterWithModel:(FBWaterClockModel *
_Nonnull)drinkWaterModel withBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
// 参考上述参数
```

34、获取勿扰提醒信息

```
- (void)fbGetNotDisturbWithBlock:(FBGetNotDisturbBlock _Nonnull)fbBlock;
/*
 勿扰提醒信息 | Do not disturb reminder message
*/
@interface FBNotDisturbModel : NSObject
/**
 勿扰提醒起始时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间）（默认 360，即 06:00） | Do not disturb reminder start time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time)
*/
@property (nonatomic, assign) NSInteger startTime;
/**
 勿扰提醒结束时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间）（默认 1260，即 21:00） | Do not disturb reminder end time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time)
*/
@property (nonatomic, assign) NSInteger endTime;
/**
 勿扰提醒开关，NO:关闭  YES:打开（默认 NO） | Reminder switch, NO: off, YES: on (Default: NO)
*/
@property (nonatomic, assign) BOOL alterSwitch;
@end
```

35、设置勿扰提醒信息

```
- (void)fbSetNotDisturbWithModel:(FBNotDisturbModel *
_Nonnull)notDisturbModel withBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
// 参考上述参数
```

36、获取心率检测信息

```
- (void)fbGetHeartTestPeriodsWithBlock:(FBGetHeartTestPeriodsBlock _Nonnull)fbBlock;
/*
 心率检测信息，为全天候检测，建议只可修改提醒周期 | The heart rate detection information is all-weather detection, and it is recommended that only the reminder cycle can be modified
*/
@interface FBHrCheckModel : NSObject
/**
 心率检测起始时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间） | Start time of heart rate detection, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time)
*/
@property (nonatomic, assign) NSInteger startTime;
/**
 心率检测结束时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间） | End time of heart rate detection, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time)
*/
@property (nonatomic, assign) NSInteger endTime;
/**
 心率检测周期，分钟，如果为 0 只检测一次，为 10 的整倍数 | Heart rate detection cycle, minutes, if it is 0, only detect once, it is an integral multiple of 10
*/
@property (nonatomic, assign) NSInteger repeatCount;
/**
 自动检测开关，NO:关闭 YES:打开 | Automatic detection switch, NO: off, YES: on
*/
@property (nonatomic, assign) BOOL alterSwitch;
@end
```

37、设置心率检测信息

```
- (void)fbSetHeartTestPeriodsWithModel:(FBHrCheckModel *_Nonnull)hrCheckModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

38、获取抬腕亮屏信息

```
- (void)fbGetWristTimeWithBlock:(FBGetWristTimeBlock _Nonnull)fbBlock;
/*
 抬腕亮屏信息 | Wrist lifting light screen information
*/
@interface FBWristModel : NSObject
```

```
/**
 抬腕有效起始时间，当天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间）（默认 360，即 06:00）| Effective starting time of wrist lifting, absolute minutes of the day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time)  (the default is 360, i.e. 06:00)
*/
@property (nonatomic, assign) NSInteger startTime;
/**
 抬腕有效结束时间，当天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间）（默认 1320，即 22:00）| Effective end time of wrist lifting, absolute minutes of the day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time)  (the default is 1320, i.e. 22:00)
*/
@property (nonatomic, assign) NSInteger endTime;
/**
 抬腕亮屏开关，NO:关闭  YES:打开（默认 YES）| Wrist lifting light screen switch, NO: off, 1: YES (Default: YES)
*/
@property (nonatomic, assign) BOOL alterSwitch;
@end
```

39、设置抬腕亮屏信息

```
- (void)fbSetWristTimeWithModel:(FBWristModel * _Nonnull)wristModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

40、获取运动目标信息

```
- (void)fbGetSportsTagargetWithBlock:(FBGetSportsTagargetBlock _Nonnull)fbBlock;
/*
 运动目标信息 | Moving target information
*/
@interface FBSportTargetModel : NSObject
/**
 目标步数开关：NO:关  YES:开 | Target step switch：NO:off YES: on
*/
@property (nonatomic, assign) BOOL stepSwitch;
/**
 目标卡路里开关：NO:关  YES:开 | Target calorie switch：NO:off YES: on
*/
@property (nonatomic, assign) BOOL caculateSwitch;
/**
 目标距离开关：NO:关  YES:开 | Target distance switch：NO:off YES: on
*/
@property (nonatomic, assign) BOOL distanceSwitch;
```

```
/**
 运动目标参数开关：NO:关  YES:开 | Moving target parameter switch：NO:off YES: on
*/
@property (nonatomic, assign) BOOL sportSwicth;
/**
 目标之步数 | Step target
*/
@property (nonatomic, assign) NSInteger stepTarget;
/**
 目标之卡路里消耗（千卡） | Target calorie consumption (kcal)
*/
@property (nonatomic, assign) NSInteger calorieTarget;
/**
 目标之里程数（米） | Distance target (m)
*/
@property (nonatomic, assign) NSInteger distanceTarget;
/**
 目标之运动时间（分钟） | Sport time target (minute)
*/
@property (nonatomic, assign) NSInteger sportTimeTarget;
@end
```

41、设置运动目标信息

```
- (void)fbSetSportsTagargetWithModel:(FBSportTargetModel *_Nonnull)sportTargetModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

42、设置今日天气详情

```
- (void)fbPushTodayWeatherDetailsWithModel:(FBWeatherDetailsModel *_Nonnull)model withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/*
 推送今日天气详情消息参数 | Push today's weather details message parameters
*/
@interface FBWeatherDetailsModel : NSObject
/**
 天气 | Weather
*/
@property (nonatomic, assign) FB_WEATHER Weather;
/**
 空气质量等级 | Air quality level
*/
```

```objc
@property (nonatomic, assign) FB_AIRLEVEL AirCategory;
/**
 空气温度 | air temperature
*/
@property (nonatomic, assign) NSInteger airTemp;
/**
 体感温度（C） | Somatosensory temperature (c)
*/
@property (nonatomic, assign) NSInteger somatTemp;

/**
 最低温度，可以为负数 | The lowest temperature can be negative
*/
@property (nonatomic, assign) NSInteger tempMin;
/**
 最高温度，可以为负数 | The highest temperature can be negative
*/
@property (nonatomic, assign) NSInteger tempMax;
/**
 日出时间，小时 | Sunrise time, hours
*/
@property (nonatomic, assign) NSInteger sunriseHours;
/**
 日出时间，分钟 | Sunrise time, minutes
*/
@property (nonatomic, assign) NSInteger sunriseMinutes;
/**
 日落时间，小时 | Sunset time, hours
*/
@property (nonatomic, assign) NSInteger sunsetHours;
/**
 日落时间，分钟 | Sunset time, minutes
*/
@property (nonatomic, assign) NSInteger sunsetMinutes;
/**
 湿度（%） | Humidity (%)
*/
@property (nonatomic, assign) NSInteger humidity;
/**
 风向 | Wind direction
*/
@property (nonatomic, assign) EM_WINDDIRECTION WindDirection;
/**
 风速度（米/秒） | Wind speed (M / s)
```

```
*/
@property (nonatomic, assign) NSInteger windSpeed;
/**
 最近 2 小时降水概率（%）| Precipitation probability in the last 2 hours (%)
*/
@property (nonatomic, assign) NSInteger probability;
/**
 降水量（毫米）| Precipitation (mm)
*/
@property (nonatomic, assign) NSInteger precipitation;
/**
 气压（百帕）| Air pressure (HPA)
*/
@property (nonatomic, assign) NSInteger airPressure;
/**
 能见度（米）| Visibility (m)
*/
@property (nonatomic, assign) NSInteger visibility;
/**
 紫外线指数 | UV index
*/
@property (nonatomic, assign) NSInteger UV_index;
@end

typedef enum {
    WT_SUNNY              = 0,   //晴 | Sunny
    WT_PARTLY_CLOUDY       = 1,   //多云 | Cloudy
    WT_WIND               = 2,   //风 | Wind
    WT_CLOUDY             = 3,   //阴天 | Overcast
    WT_LIGHT_RAIN         = 4,   //小雨 | Light rain
    WT_HEAVY_RAIN         = 5,   //大雨 | Heavy rain
    WT_SNOW               = 6,   //雪 | Snow
    WT_THUNDER_SHOWER      = 7,   //雷阵雨 | Thunder shower
    WT_SUNNY_NIGHT        = 8,   //晴晚上 | Sunny night
    WT_PARTLY_CLOUDY_NIGHT = 9,   //多云晚上 | Cloudy night
    WT_STANDSTORM         = 10,  //沙尘暴 | Sand storm
    WT_SHOWER             = 11,  //阵雨 | Shower
    WT_SHOWER_NIGHT        = 12,  //阵雨晚上 | Shower night
    WT_SLEET              = 13,  //雨夹雪 | Sleet
    WT_SMOG               = 14,  //雾、霾 | Fog and haze
    WT_LIGHT_SNOW         = 15,  //小雪 | Light snow
    WT_HEAVY_SNOW         = 16,  //大雪 | Heavy snow
    WT_MODERATE_RAIN       = 17,  //中雨 | Moderate rain
    WT_RAINSTORM          = 18,  //暴雨 | Rainstorm
```

```
    WT_UNKNOW             = 255, //未知天气 | Unknown weather
}FB_WEATHER;

typedef enum {
    AL_BAD     = 0,  //差 | Bad
    AL_GOOD    = 1,  //良 | Good
    AL_WONDFUL = 2,  //优 | Wonderful
}FB_AIRLEVEL;

typedef enum{
    WD_0 = 0,  //无风 | No wind
    WD_1 = 1,  //东风 | East wind
    WD_2 = 2,  //东南风 | Southeast wind
    WD_3 = 3,  //南风 | South wind
    WD_4 = 4,  //西南风 | Southwest wind
    WD_5 = 5,  //西风 | Westerly
    WD_6 = 6,  //西北风 | Northwest wind
    WD_7 = 7,  //北风 | North wind
    WD_8 = 8,  //东北风 | Northeasterly wind
}EM_WINDDIRECTION;
```

43、设置未来天气预报信息

```
- (void)fbPushWeatherMessageWithModel:(NSArray <FBWeatherModel *>
*)weatherArray support:(BOOL)support_14days_Weather
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
/*
 推送天气消息参数 | Push weather message parameters
*/
@interface FBWeatherModel : NSObject
/**
 序号 ID，0:昨天、1:今天、2:明天、3:后天..... | Serial number ID, 0: yesterday, 1: today, 2: tomorrow, 3: the day after tomorrow

 @note 根据 FBAllConfigObject.firmwareConfig.support_14days_Weather 来标识是否支持未来 14 天天气预报(YES :序号 ID 支持 0-15，
NO：序号 ID 支持 0-6） | According to FBAllConfigObject.firmwareConfig.support_14days_Weather to identify whether the weather
forecast for the next 14 days is supported (YES: serial number ID supports 0-15, NO: serial number ID supports 0-6)
*/
@property (nonatomic, assign) NSInteger days;
/**
 天气 | Weather
*/
@property (nonatomic, assign) FB_WEATHER Weather;
/**
```

```
最低温度，可以为负数 | The lowest temperature can be negative
*/
@property (nonatomic, assign) NSInteger tempMin;
/**
最高温度，可以为负数 | The highest temperature can be negative
*/
@property (nonatomic, assign) NSInteger tempMax;
/**
空气质量等级 | Air quality level
*/
@property (nonatomic, assign) FB_AIRLEVEL AirCategory;
/**
PM2.5 等级 | PM2.5
*/
@property (nonatomic, assign) FB_PM25 PM2p5;
@end

typedef enum {
    PM_LEVEL1 = 0,  //优 | Wonderful
    PM_LEVEL2 = 1,  //良 | Good
    PM_LEVEL3 = 2,  //轻度污染 | Light pollution
    PM_LEVEL4 = 3,  //中度污染 | Moderate pollution
    PM_LEVEL5 = 4,  //重度污染 | Heavy pollution
}FB_PM25;
```

~~44、app 推送手机定位信息~~

```
~~- (void)fbPushMobileLocationInformationWithLongitude:(float)longitude withLatitude:(float)latitude withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;~~
```

45、获取女性生理周期信息

```
-(void)fbGetFemaleCircadianCycleWithBlock:(FBGetFemaleCircadianCycleBlock _Nonnull)fbBlock;
/*
 女性生理周期信息 | Female physiological cycle information
*/
@interface FBFemalePhysiologyModel : NSObject
/**
健康模式设置 | Health mode setting
*/
@property (nonatomic, assign) FB_FEMALEPHYSIOLOGICALHEALTHMODEL HealthModel;
/**
```

```
 经期开始提醒提前天数，范围 1-3 天 ｜The number of days in advance of menstruation start reminder, ranging from 1 to 3 days
*/
@property (nonatomic, assign) NSInteger daysInAdvance;
/**
 经期的天数，范围 3-15 天｜The number of days of menstruation, ranging from 3 to 15 days
*/
@property (nonatomic, assign) NSInteger daysMenstruation;
/**
 周期长度，范围 17-60 天｜Cycle length, ranging from 17 to 60 days
*/
@property (nonatomic, assign) NSInteger cycleLength;
/**
 最近一次月经，年（最近两年）｜Last menstruation, year (last two years)
*/
@property (nonatomic, assign) NSInteger lastYear;
/**
 最近一次月经，月｜Last menstruation, month
*/
@property (nonatomic, assign) NSInteger lastMonth;
/**
 最近一次月经，日｜The last menstruation, day
*/
@property (nonatomic, assign) NSInteger lastDay;
/**
 孕期提醒方式：NO 提示已怀孕天数，YES 提示距离预产期天数｜Pregnancy reminder: no indicates the number of days pregnant, yes indicates the number of days away from the expected delivery date
*/
@property (nonatomic, assign) BOOL isPreProduction;
/**
 提醒时间，小时｜Reminder time, hours
*/
@property (nonatomic, assign) NSInteger reminderHours;
/**
 提醒时间，分钟｜Reminder time, minutes
*/
@property (nonatomic, assign) NSInteger reminderMinutes;
/**
 设备提醒开关，NO 关，YES 开｜Device reminder switch, no off, yes on
*/
@property (nonatomic, assign) BOOL reminderSwitch;
@end

typedef enum {
    FB_HealthModel_NotUsed             = 0,  //未启用｜Not used
```

```
    FB_HealthModel_Menstrual           = 1,  //月经期|Menstrual period
    FB_HealthModel_PregnancyPreparation = 2,  //备孕期|Pregnancy preparation period
    FB_HealthModel_Pregnancy           = 3,  //怀孕期|Pregnancy
}FB_FEMALEPHYSIOLOGICALHEALTHMODEL;
```

46、设置女性生理周期信息

```
- (void)fbSetFemaleCircadianCycleWithModel:(FBFemalePhysiologyModel *_Nonnull)physiologyModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

47、获取心率异常提醒信息

```
-(void)fbGetAbnormalHeartRateReminderWithBlock:(FBGetAbnormalHeartRateReminderBlock _Nonnull)fbBlock;
/*
 心率异常提醒信息|Abnormal heart rate reminder information
*/
@interface FBHrReminderModel : NSObject
/**
 心率异常提醒开关 NO:关闭 YES:打开|Abnormal heart rate reminder switch No: off yes: on
*/
@property (nonatomic, assign) BOOL enable;
/**
 心率提醒上限，心率超高提醒|Heart rate reminder upper limit, heart rate ultra-high reminder
*/
@property (nonatomic, assign) NSInteger highReminder;
/**
 心率提醒下限，心率过低提醒|Low heart rate reminder
*/
@property (nonatomic, assign) NSInteger lowReminder;
/**
 心率值连续超标次数（达到超标的次数时才会提醒）|The number of times the heart rate value exceeds the standard continuously (it will be reminded only when the number exceeds the standard)
*/
@property (nonatomic, assign) NSInteger exceedanceTimes;
/**
 检测起始时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，起始时间小于结束时间）|Detection start time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, start time less than end time)
*/
@property (nonatomic, assign) NSInteger startTime;

/**
```

```
检测结束时间，一天的绝对分钟（大于等于 0 分钟，小于 1440 分钟，结束时间大于起始时间）| Detection end time, absolute minutes of a day (greater than or equal to 0 minutes, less than 1440 minutes, end time greater than start time)
*/
@property (nonatomic, assign) NSInteger endTime;
@end
```

48、设置心率异常提醒信

```
- (void)fbSetAbnormalHeartRateReminderWithModel:(FBHrReminderModel *_Nonnull)hrReminderModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

49、GPS 运动互联数据交互

```
-(void)fbGPSMotionInterconnectionWithModel:(FBMotionInterconnectionModel * _Nonnull)model withBlock:(FBMotionInterconnectionBlock _Nonnull)fbBlock;
/*
 GPS 运动互联数据交互信息 | Interactive information of GPS motion interconnection data
 */
@interface FBMotionInterconnectionModel : NSObject
#pragma mark - 以下数据由 APP 提供填充 | The following data is filled in by app
/** 当前时间(UTC) | Current time (UTC) */
@property (nonatomic, assign) NSInteger currentTimeUTC;
/** 运动 ID，用运动的开始时间作为每笔运动的唯一识别码 | Motion ID, using the start time of the motion as the unique identification code of each motion */
@property (nonatomic, assign) NSInteger motionID;
/** 运动开始时间戳(UTC) | Motion start timestamp (UTC) */
@property (nonatomic, assign) NSInteger startMotionUTC;
/** 运动结束时间(UTC)，进行中的运动填 0 | Motion end time (UTC), fill in 0 for ongoing exercise */
@property (nonatomic, assign) NSInteger endMotionUTC;
/** 当前运动总时间，单位秒 | Total current movement time, in seconds */
@property (nonatomic, assign) NSInteger totalTime;
/** 当前运动总卡路里（千卡） | Total calories of current exercise (kcal) */
@property (nonatomic, assign) NSInteger totalCalories;
/** 当前本次运动轨迹运动距离（单位米，通过 gps 定位计算） | Motion distance of current trajectory (unit: m, calculated by GPS positioning) */
@property (nonatomic, assign) NSInteger motionDistance;
/** 运动模式 | Motion mode */
@property (nonatomic, assign) FB_MOTIONMODE MotionMode;
/** 本次运动最大步频（单位：步/分钟） | Maximum step frequency of this movement (unit: step / minute) */
@property (nonatomic, assign) NSInteger maxStepFrequency;
/** 本次运动平均步频 =步数/时间（单位：步/分钟） | Average step frequency of this exercise = steps / time (unit: steps / minute) */
```

```
@property (nonatomic, assign) NSInteger avgStepFrequency;

/** 重复运动的周期数（来回次数，圈数）（单位：圈）| Number of cycles of repeated motion (number of turns, turns) (unit: turns) */
@property (nonatomic, assign) NSInteger cyclesNumber;
/** 本次运动最大速度（单位：米/秒）| Maximum speed of this movement (unit: M / s) */
@property (nonatomic, assign) CGFloat maxSpeed;
/** 本次运动平均速度 = 距离/用时（单位：米/秒）| Average speed of this movement = distance / time (unit: M / s) */
@property (nonatomic, assign) CGFloat avgSpeed;
/** 本次有轨迹运动配速（单位：秒/公里）| This time there is track movement pace (unit: S / km) */
@property (nonatomic, assign) NSInteger trackPace;

#pragma mark - 以下数据 APP 或手表由提供（双方都有权修改）| The following data is provided by app or watch (both parties have the right to modify)
/** 中途休息次数 | Number of breaks */
@property (nonatomic, assign) NSInteger breaksNumber;
/** 运动状态,0 停止,1 进行中,2 暂停(表明当前运动状态,非控制指令) | Motion state, 0 stop, 1 in progress, 2 pause (indicating current motion state, non control command) */
@property (nonatomic, assign) NSInteger motionState;

#pragma mark - 以下数据由手表填充返回，APP 无需设置 | The following data is filled and returned by the watch. App does not need to be set
/** 当前运动总步数 | Total current motion steps */
@property (nonatomic, assign) NSInteger totalSteps;
/** 本次运动当前实时心率（单位：次/分钟）| Current real-time heart rate during this exercise (unit: times / minute) */
@property (nonatomic, assign) NSInteger currentHeartRate;
/** 本次运动最大心率（单位：次/分钟）| Maximum heart rate of this exercise (unit: times / minute) */
@property (nonatomic, assign) NSInteger maxHeartRate;
/** 本次运动最小心率（单位：次/分钟）| Minimum heart rate of this exercise (unit: times / minute) */
@property (nonatomic, assign) NSInteger minHeartRate;
/** 本次运动实时平均心率（单位：次/分钟）| Real time average heart rate of this exercise (unit: times / minute) */
@property (nonatomic, assign) NSInteger avgHeartRate;
/** 当前心率处于的区间（热身，燃脂，有氧，高强有氧，无氧）| The range of current heart rate (warm-up, fat burning, aerobic, high-strength aerobic, anaerobic) */
@property (nonatomic, assign) FB_MOTIONHEARTRATERANGE currentHrRange;
/** 到当前为止，热身运动时间，单位分钟，随时刷新 | Up to now, the warm-up exercise time, in minutes, is refreshed at any time */
@property (nonatomic, assign) NSInteger heartRate_level_1;
/** 到当前为止，燃脂运动时间，单位分钟，随时刷新 | Up to now, the fat burning movement time, in minutes, can be refreshed at any time */
@property (nonatomic, assign) NSInteger heartRate_level_2;
/** 到当前为止，有氧耐力运动时间，单位分钟，随时刷新 | So far, aerobic endurance exercise time, in minutes, can be refreshed at any time */
@property (nonatomic, assign) NSInteger heartRate_level_3;
/** 到当前为止，高强有氧运动时间，单位分钟，随时刷新 | So far, the time of high-strength aerobic exercise, in minutes, can be refreshed at any time */
```

```
@property (nonatomic, assign) NSInteger heartRate_level_4;
/** 到当前为止，无氧运动时间，单位分钟，随时刷新 | So far, the anaerobic exercise time, in minutes, can be refreshed at any time */
@property (nonatomic, assign) NSInteger heartRate_level_5;
@end
```

50、获取常用联系人信息

```
-(void)fbGetFavoriteContactListWithBlock:(FBGetFavoriteContactListBlock _Nonnull)fbBlock;
/*
 常用联系人信息 | Frequently used contact information
 */
@interface FBFavContactModel : NSObject
/**
 联系人姓名(长度小于或等于 64 个字节,超出最大长度,自动截取) | Contact name (less than or equal to 64 bytes in length, automatically intercepted if the maximum length is exceeded)
 */
@property (nonatomic, copy) NSString *contactName;
/**
 号码归属地（长度小于或等于 64 个字节，超出最大长度，自动截取） | Number location (if the length is less than or equal to 64 bytes, it will be automatically intercepted if the maximum length is exceeded)
 */
@property (nonatomic, copy) NSString *QCellCore;
/**
 联系人号码（长度小于或等于 20 个字节，超出最大长度，自动截取） | Contact number (less than or equal to 20 bytes in length, automatically intercepted if the maximum length is exceeded)
 */
@property (nonatomic, copy) NSString *contactNumber;
@end
```

51、设置常用联系人信息

```
- (void)fbSetFavoriteContactListWithModel:(NSArray <FBFavContactModel *> *)modelList withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
// 参考上述参数
```

52、读取片外 flash 空间数据（设备意外重启信息，供固件分析问题）

```
- (void)fbReadOffChipFlashWithAddress:(NSInteger)address withLength:(NSInteger)length withBlock:(FBRequestDeviceLogsBlock _Nonnull)fbBlock;
```

53、请求获取设备日志（埋点数据，读取 OTA 缓存数据，总共 60K）

```
- (void)fbRequestDeviceLogsWithBlock:(FBRequestDeviceLogsBlock
_Nonnull)fbBlock;
```

54、获取系统功能开关信息

```
-
(void)fbGetSystemFunctionSwitchInformationWithBlock:(FBRequestSystemFun
ctionSwitchInfoBlock _Nonnull)fbBlock;
```

55、设置系统功能开关信息

```
-
(void)fbSetSystemFunctionSwitchInformation:(FBSystemFunctionSwitchModel
*_Nonnull)switchModel withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

56、推送 AGPS 位置基础信息(经纬度 UTC)

```
- (void)fbPushAGPSLocationInformation:(FBAGPSLocationModel *
_Nonnull)locationModel withBlock:(FBResultCallBackBlock
_Nonnull)fbBlock;
```

57、获取日程信息

```
- (void)fbGetScheduleInforWithBlock:(FBGetScheduleInforBlock
_Nonnull)fbBlock;
```

58、设置日程信息

```
- (void)fbSetSchedulenforWithScheduleModel:(FBScheduleModel *
_Nonnull)scheduleModel withRemoved:(BOOL)isRemoved
withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

59、获取紧急联系人信息

```
-
(void)fbGetEmergencyContactListWithBlock:(FBGetFavoriteContactListBlock
_Nonnull)fbBlock;
```

60、设置紧急联系人信息

```
- (void)fbSetEmergencyContactListWithModel:(NSArray <FBFavContactModel *> *)modelList withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

61、获取系统空间使用信息

```
- (void)fbGetSystemSpaceUsageInforWithBlock:(FBGetSystemSpaceInforBlock _Nonnull)fbBlock;
```

62、获取表盘列表文件信息

```
- (void)fbGetDialListFileInforWithBlock:(FBGetListFileInforBlock _Nonnull)fbBlock;
```

63、获取 JS 应用列表文件信息

```
- (void)fbGetJsAppListFileInforWithBlock:(FBGetListFileInforBlock _Nonnull)fbBlock;
```

64、删除表盘列表文件信息

```
- (void)fbDeleteDialListFileInfor:(NSArray <FBListFileInforModel *> * _Nonnull)modelList withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

65、删除 JS 应用列表文件信息

```
- (void)fbDeleteJsAppListFileInfor:(NSArray <FBListFileInforModel *> * _Nonnull)modelList withBlock:(FBResultCallBackBlock _Nonnull)fbBlock;
```

➢ 获取流数据 API（FBAtCommand）

1. 开启数据流 `fbUpDataStreamData: withBlock:` 指令后，有数据更新时，设备会按设定的时间间隔通过此回调返回数据

```
- (void)fbStreamDataHandlerWithBlock:(FBStreamDataHandlerBlock _Nonnull)fbBlock;
/*
 流数据 | Stream data
*/
@interface FBStreamDataModel : NSObject
/**
 流帧计数，自动递增，0-255，溢出后清零 | Stream frame count, auto increment, 0-255, clear after overflow
*/
@property (nonatomic, assign) NSInteger streamCount;
/**
 当前心率（次/分钟） | Current heart rate (times / minute)
*/
@property (nonatomic, assign) NSInteger currentHeartRate;
```

```objc
/**
 当前心率等级 | Current heart rate level
*/
@property (nonatomic, assign) FB_CURRENTHEARTRANGE HeartRateRange;
/**
 当前累计步数 | Current cumulative steps
*/
@property (nonatomic, assign) NSInteger currentStepCount;
/**
 当前累计距离（米） | Current cumulative distance (m)
*/
@property (nonatomic, assign) NSInteger currentDistance;
/**
 当前累计消耗卡路里（千卡） | Current cumulative calories consumed (kcal)
*/
@property (nonatomic, assign) NSInteger currentCalories;
@end
```

➢ OTA 工具 API（FBBluetoothOTA）

1. 生成自定义表盘 bin 文件数据（FBCustomDataTools）

```
- (NSData
*)fbGenerateCustomDialBinFileDataWithDialModel:(FBCustomDialModel *
_Nonnull)dialModel;
```

2. ~~生成多项目自定义表盘 bin 文件数据（FBCustomDataTools）~~

```
- (NSData
*)fbGenerateMultiProjectCustomDialBinFileDataWithDialsModel:(FBMultiple
CustomDialsModel * _Nonnull)dialsModel;
// NOTE: 由于表盘内存空间有限，自定义内容最多不能超过 16 个控件，不同样式所占用的控件个数都有所不同。添加自定义内容前需要检查表盘空间是否足够
+ (BOOL)checkForOverflow:(NSArray <FBCustomDialItem *> *)items;
```

3. 生成自定义运动类型 bin 文件数据（多个运动类型 Bin 文件压缩合并成一个 Bin 文件）

（FBCustomDataTools）

```
- (NSData *)fbGenerateCustomMultipleMotionBinFileDataWithItems:(NSArray
<NSData *> * _Nonnull)items;
```

4. 传入不同的 OTAType 进行 OTA 数据同步，为避免变砖，请先确认项目是否支持

```
- (void)fbStartCheckingOTAWithBinFileData:(NSData * _Nonnull)binFile
withOTAType:(FB_OTANOTIFICATION)OTAType
withBlock:(FBSetOtaUpgradeManagerBlock _Nonnull)fbBlock;
```

```
typedef enum {
    FB_OTANotification_Firmware           = 0,    //升级固件 | Update Firmware
    FB_OTANotification_ClockDial          = 1,    //升级默认动态表盘 | Upgrade default dynamic dial
    FB_OTANotification_SmallFont          = 2,    //升级小字库 | Upgrade small font
    FB_OTANotification_BigFont            = 3,    //升级大字库 | Upgrade big font
    FB_OTANotification_UIPictureResources = 4,    //升级 UI 图片资源 | Upgrade UI image resources
    FB_OTANotification_2_3_4AtTheSameTime = 5,    //同时升级 2,3,4 | Upgrade 2, 3, 4 at the same time
    FB_OTANotification_Motion             = 6,    //推送运动模式 | Push motion mode
```

```
    FB_OTANotification_UI                = 7,    //增量升级 UI 图片 | Incrementally upgrade UI images
    FB_OTANotification_Multi_Dial        = 8,    //多表盘压缩数据包 | Multi-dial compressed data package
    FB_OTANotification_Multi_Sport       = 9,    //多运动类型压缩数据包 | Multi-sport type compressed data package
    FB_OTANotification_DynamicClockDial  = 10,   //+n，升级动态表盘 n | +n. Upgrade dynamic dial n
    FB_OTANotification_CustomClockDial   = 20,   //+n，升级自定义表盘 n | +n. Upgrade custom dial n
    FB_OTANotification_AGPS_Package      = 30,   //推送 AGPS 定位数据包 | Push AGPS positioning data package
    FB_OTANotification_Multi_Dial_Built_in = 200,  //厂线推送内置表盘压缩数据包 | The factory line pushes the built-in dial compressed data package
    FB_OTANotification_Multi_Sport_Built_in = 201,  //厂线推送内置多运动类型压缩数据包 | The factory line pushes the built-in multi-sport type compressed data package
    FB_OTANotification_Busy              = 254,  //设备处于禁止 OTA 状态，稍后再试 | The device is in OTA prohibited state, please try again later
    FB_OTANotification_Cancel            = 255,  //放弃当前升级 | Discard current upgrade
}FB_OTANOTIFICATION;
// 更多类型请查看 SDK 内 FBMacro.h 文件

/*
 OTA 完成信息 | OTA completion information
*/
@interface FBOTADoneModel : NSObject
/**
 ota 类型 | OTA type
*/
@property (nonatomic, assign) FB_OTANOTIFICATION type;
/**
 bin 二进制文件 | Bin binary
*/
@property (nonatomic, retain) NSData *binFile;
/**
 ota 升级总时长（单位秒） | Total OTA upgrade time (in seconds)
*/
@property (nonatomic, assign) NSInteger totalInterval;
/**
 平均速率（单位 KB/s） | Average velocity (in KB / s)
*/
@property (nonatomic) float velocity;
@end
```

➢ 错误码枚举定义 FB_RET_CMD

```
typedef enum {
    //协议定义的通讯错误 | Protocol defined communication error
    RET_EXEC_ER        = 3,    //执行失败 | Execution failed
```

```
    RET_DATA_INVA       = 4,    //数据无效（格式错误）|Invalid data (format error)
    RET_COMM_BUSY       = 5,    //通信中（系统忙）|Communication (system busy)
    RET_COMD_INVA       = 6,    //无效指令|Invalid instruction
    RET_PVER_ER         = 7,    //协议版本不符|Protocol version does not match
    RET_DATA_OK         = 8,    //数据正确|The data is correct
    RET_DATA_ER         = 9,    //数据错误|Data error
    RET_DATA_TO         = 10,   //接收超时|Receive timeout
    RET_RAM_OF          = 11,   //内存溢出|Out of memory
    RET_CHKS_ER         = 12,   //校验和错误|Check sum error
    RET_PARA_ER         = 13,   //参数错误|Parameter error
    RET_LENG_ER         = 14,   //数据长度错误|Data length error

    //自定义蓝牙通信错误|Custom Bluetooth communication error
    RET_FB_ERR_OTA      = 15,   //OTA 失败，不支持 OTA|OTA failed, OTA not supported
    RET_FB_ERR_OFF      = 16,   //蓝牙未打开或不支持|Bluetooth is not on or not supported
    RET_FB_ERR_NOT      = 17,   //尚未连接到设备|Not yet connected to the device
    RET_FB_ERR_NOTREADY = 18,   //设备尚未初始化完成|The device has not been initialized
    RET_FB_ERR_NET      = 19,   //写指令失败，或缺少连接参数|Write instruction failed, or connection
parameters are missing
    RET_FB_ERR_AT       = 20,   //AT 指令异常，无效指令/无效参数|At instruction exception, invalid
instruction / invalid parameter
    RET_FB_ERR_DATA     = 21,   //失败，数据校验未通过|Failed, data verification failed
    RET_FB_ERR_TIMEROUT = 22,   //应答超时|Response timeout

    //自定义数据传输状态|Data transmission status
    FB_INDATATRANSMISSION     = 101,    //数据传输中|In data transmission
    FB_DATATRANSMISSIONDONE   = 200,    //数据传输完成|Data transmission complete
    FB_DATATRANSMISSIONFAILED = 500,    //数据传输失败|Data transfer failed

    //GPS 运动状态执行错误|GPS motion status execution error
    FB_GPS_MOTION_STATE_LOWPRESSUREERROR     = 200012, //执行失败，低压无法执行|Execution failed, low
voltage cannot be executed
    FB_GPS_MOTION_STATE_COMMANDSTATUSERROR   = 200013, //指令状态错误|Command status error
    FB_GPS_MOTION_STATE_INREGULARMOTIONERROR = 200014, //常规运动中，请先停止当前运动|In normal motion,
please stop the current motion first
    FB_GPS_MOTION_STATE_INCALLERROR          = 200015, //正在通话中，无法执行此指令|This command cannot be
executed while a call is in progress
    FB_GPS_MOTION_STATE_CANCELS              = 200017, //手表取消开启运动|The watch cancels the movement
    FB_GPS_MOTION_STATE_NONE                 = 200019, //本地无此运动信息|There is no local sports
information
}FB_RET_CMD;
```

➢ 运动模式枚举定义 FB_MOTIONMODE

```
typedef enum {
    FBNotUsed               = 0,   //不使用 | Not used
    FBRunning               = 1,   //跑步 | Running
    FBMountaineering        = 2,   //登山 | Mountaineering
    FBCycling               = 3,   //骑行 | Cycling
    FBFootball              = 4,   //足球 | Football
    FBSwimming              = 5,   //游泳 | Swimming
    FBBasketball            = 6,   //篮球 | Basketball
    FBNo_designation        = 7,   //无指定 | No designation
    FBOutdoor_running       = 8,   //户外跑步 | Outdoor running
    FBIndoor_running        = 9,   //室内跑步 | Indoor running
    FBFat_reduction_running = 10,  //减脂跑步 | Fat reduction running

    FBOutdoor_walking       = 11,  //户外健走 | Outdoor walking
    FBIndoor_walking        = 12,  //室内健走 | Indoor walking
    FBOutdoor_cycling       = 13,  //户外骑行 | Outdoor cycling
    FBIndoor_cycling        = 14,  //室内骑行 | Indoor cycling
    FBFree_training         = 15,  //自由训练 | Free training
    FBFitness_training      = 16,  //健身训练 | Fitness training
    FBBadminton             = 17,  //羽毛球 | Badminton
    FBVolleyball            = 18,  //排球 | Volleyball
    FBTable_Tennis          = 19,  //乒乓球 | Table Tennis
    FBElliptical_machine    = 20,  //椭圆机 | Elliptical machine

    FBRowing_machine        = 21,  //划船机 | Rowing machine
    FBYoga_training         = 22,  //瑜伽 | Yoga
    FBStrength_training     = 23,  //力量训练（举重） | Strength training (weightlifting)
    FBCricket               = 24,  //板球 | Cricket
    FBRope_skipping         = 25,  //跳绳 | Rope skipping
    FBAerobic_exercise      = 26,  //有氧运动 | Aerobic exercise
    FBAerobic_dancing       = 27,  //健身舞 | Aerobic dancing
    FBTaiji_boxing          = 28,  //太极 | Tai Chi
    FBAuto_runing           = 29,  //自动识别跑步运动 | Automatically recognize running
    FBAuto_walking          = 30,  //自动识别健走运动 | Automatic recognition of walking movement

    FBWALK                  = 31,  //室内步行 | Indoor walking
    FBSTEP_TRAINING         = 32,  //踏步 | Step training
    FBHORSE_RIDING          = 33,  //骑马 | Ride a horse
    FBHOCKEY                = 34,  //曲棍球 | Hockey
    FBINDOOR_CYCLE          = 35,  //室内单车 | Aerodyne bike
    FBSHUTTLECOCK           = 36,  //毽球 | Shuttlecock
    FBBOXING                = 37,  //拳击 | Boxing
    FBOUTDOOR_WALK          = 38,  //户外走 | Outdoor walk
    FBTRAIL_RUNNING         = 39,  //越野跑 | Cross country running
```

```
    FBSKIING                = 40,  //滑雪 | Skiing

    FBGYMNASTICS            = 41,  //体操 | Artistic Gymnastics
    FBICE_HOCKEY            = 42,  //冰球 | Ice hockey
    FBTAEKWONDO             = 43,  //跆拳道 | Taekwondo
    FBVO2MAX_TEST           = 44,  //有氧运动 | Aerobic exercise
    FBAIR_WALKER            = 45,  //漫步机 | Walking machine
    FBHIKING                = 46,  //徒步 | On foot
    FBTENNIS                = 47,  //网球 | Tennis
    FBDANCE                 = 48,  //跳舞 | Dance
    FBTRACK_FIELD           = 49,  //田径 | Athletics
    FBABDOMINAL_TRAINING    = 50,  //腰腹运动 | Lumbar abdominal movement

    FBKARATE                = 51,  //空手道 | Karate
    FBCOOLDOWN              = 52,  //整理放松 | Organize and relax
    FBCROSS_TRAINING        = 53,  //交叉训练 | Cross training
    FBPILATES               = 54,  //普拉提 | Pilates
    FBCROSS_FIT             = 55,  //交叉配合 | Cross fit
    FBUNCTIONAL_TRAINING    = 56,  //功能性训练 | Functional training
    FBPHYSICAL_TRAINING     = 57,  //体能训练 | Physical training
    FBARCHERY               = 58,  //射箭 | Archery
    FBFLEXIBILITY           = 59,  //柔韧度 | Flexibility
    FBMIXED_CARDIO          = 60,  //混合有氧 | Mixed aerobic

    FBLATIN_DANCE           = 61,  //拉丁舞 | Latin dance
    FBSTREET_DANCE          = 62,  //街舞 | Hip hop
    FBKICKBOXING            = 63,  //自由搏击 | Free fight
    FBBARRE                 = 64,  //芭蕾舞 | Ballet
    FBAUSTRALIAN_FOOTBALL   = 65,  //澳式足球 | Australian football
    FBMARTIAL_ARTS          = 66,  //武术 | Australian football
    FBSTAIRS                = 67,  //爬楼 | Climb a building
    FBHANDBALL              = 68,  //手球 | Handball
    FBBASEBALL              = 69,  //棒球 | Baseball
    FBBOWLING               = 70,  //保龄球 | Bowling

    FBRACQUETBALL           = 71,  //壁球 | Squash
    FBCURLING               = 72,  //冰壶 | Curling
    FBHUNTING               = 73,  //打猎 | Go hunting
    FBSNOWBOARDING          = 74,  //单板滑雪 | Snowboarding
    FBPLAY                  = 75,  //休闲运动 | Leisure sports
    FBAMERICAN_FOOTBALL     = 76,  //美式橄榄球 | American football
    FBHAND_CYCLING          = 77,  //手摇车 | Handcart
    FBFISHING               = 78,  //钓鱼 | Go fishing
    FBDISC_SPORTS           = 79,  //飞盘 | Frisbee
```

```
FBRUGBY                = 80,  //橄榄球 | Rugby

FBGOLF                 = 81,  //高尔夫 | Golf
FBFOLK_DANCE           = 82,  //民族舞 | Folk dance
FBDOWNHILL_SKIING      = 83,  //高山滑雪 | Alpine skiing
FBSNOW_SPORTS          = 84,  //雪上运动 | Snow Sports
FBMIND_BODY            = 85,  //舒缓冥想类运动 | Soothing meditation exercise
FBCORE_TRAINING        = 86,  //核心训练 | Core training
FBSKATING              = 87,  //滑冰 | Core training
FBFITNESS_GAMING       = 88,  //健身游戏 | Fitness games
FBAEROBICS             = 89,  //健身操 | Aerobics
FBGROUP_TRAINING       = 90,  //团体操 | Group Gymnastics

FBKENDO                = 91,  //搏击操 | Kickboxing
FBLACROSSE             = 92,  //长曲棍球 | Lacrosse
FBROLLING              = 93,  //泡沫轴筋膜放松 | Foam shaft fascia relax
FBWRESTLING            = 94,  //摔跤 | Wrestling
FBFENCING              = 95,  //击剑 | Fencing
FBSOFTBALL             = 96,  //垒球 | Softball
FBSINGLE_BAR           = 97,  //单杠 | Horizontal bar
FBPARALLEL_BARS        = 98,  //双杠 | Parallel bars
FBROLLER_SKATING       = 99,  //轮滑 | Roller-skating
FBHULA_HOOP            = 100, //呼啦圈 | Hu la hoop

FBDARTS                = 101, //飞镖 | Darts
FBPICKLEBALL           = 102, //匹克球 | Pickleball
FBSIT_UP               = 103, //仰卧起坐 | Abdominal curl
FBHIIT                 = 104, //HIIT | HIIT
FBWAIST_TRAINING       = 105, //腰腹训练 | Waist and abdomen training
FBTREADMILL            = 106, //跑步机 | Treadmill
FBBOATING              = 107, //划船 | Rowing
FBJUDO                 = 108, //柔道 | rowing
FBTRAMPOLINE           = 109, //蹦床 | Trampoline
FBSKATEBOARDING        = 110, //滑板 | Skate

FBHOVERBOARD           = 111, //平衡车 | Balance car
FBBLADING              = 112, //溜旱冰 | Roller skating
FBPARKOUR              = 113, //跑酷 | Parkour
FBDIVING               = 114, //跳水 | Diving
FBSURFING              = 115, //冲浪 | Surfing
FBSNORKELING           = 116, //浮潜 | Snorkeling
FBPULL_UP              = 117, //引体向上 | Pull up
FBPUSH_UP              = 118, //俯卧撑 | Push up
FBPLANKING             = 119, //平板支撑 | Plate support
```

```
    FBROCK_CLIMBING         = 120, //攀岩 | Rock Climbing

    FBHIGHTJUMP             = 121, //跳高 | High jump
    FBBUNGEE_JUMPING        = 122, //蹦极 | Bungee jumping
    FBLONGJUMP              = 123, //跳远 | Long jump
    FBSHOOTING              = 124, //射击 | Shooting
    FBMARATHON              = 125, //马拉松 | Marathon
    FBVO2MAXTEST            = 126, //最大摄氧量测试 | VO2max test
    FBKITE_FLYING           = 127, //放风筝 | Kite Flying
    FBBILLIARDS             = 128, //台球 | Billiards
    FBCARDIO_CRUISER        = 129, //有氧运动巡洋舰 | Cardio Cruiser
    FBTUGOFWAR              = 130, //拔河比赛 | Tug of war

    FBFREESPARRING          = 131, //免费的陪练 | Free Sparring
    FBRAFTING               = 132, //漂流 | Rafting
    FBSPINNING              = 133, //动感单车 | Spinning
    FBBMX                   = 134, //BMX | BMX
    FBATV                   = 135, //ATV | ATV
    FBDUMBBELL              = 136, //哑铃 | Dumbbell
    FBBEACHFOOTBALL         = 137, //沙滩足球 | Beach Football
    FBKAYAKING              = 138, //皮划艇 | Kayaking
    FBSAVATE                = 139, //法国式拳击 | Savate
    FBBEACHVOLLEYBALL       = 140, //沙滩排球 | Beach Volleyball

    FBOther_reservation     = 255, //其他预留 | Other reservation
}FB_MOTIONMODE;
// 更多类型请查看 SDK 内 FBMacro.h 文件
```