**1.扫描手环**

**与业务无关，我们sdk自己实现**

/// \param names 蓝牙名称

///

/// \param duplicate 是否重复

///

/// \param discoveredCallback 搜索到设备回调，返回设备 MAC 地址, 名称， RSSI

///

/// \param compeletion 搜索结束回调

- (void)scanWithNames:(NSArray<NSString \*> \* \_Nonnull)names duplicate:(BOOL)duplicate discoveredCallback:(void (^ \_Nonnull)(NSString \* \_Nonnull, NSString \* \_Nonnull, NSInteger))discoveredCallback compeletion:(void (^ \_Nonnull)(NSError \* \_Nullable))compeletion;

**2.停止扫描**

**与业务无关，我们sdk自己实现**

- (void)stopScan;

**3.恢复出厂设置**

/// \param callback 返回命令发送是否成功的回调

///

- (void)resetFactoryWithCallback:(void (^ \_Nonnull)(BOOL))callback;

**4.时间同步**

/// 同步时间

- (void)syncTimeWithCallback:(void (^ \_Nullable)(BOOL))callback;

**5.获取版本**

/// \param callback 若返回 0 ，则说明获取版本失败

///

- (void)fetchDeviceVersionWithCallback:(void (^ \_Nonnull)(NSString \* \_Nullable))callback;

**6.配对**

**与业务无关，我们sdk自己实现**

- (void)pair;

**7.取消所有任务并断开连接**

/// \param finishCallback 连接断开后回调

///

-(void)cancelTaskAndUnlinkWithFinishCallback:(void(^ \_Nonnull)(void))finishCallback;

**8.状态：**

a. isConnected返回设备是否已正常连接

- (BOOL)isConnected

b. isBluetoothPoweredOn 返回蓝牙是否开启状态

- (BOOL)isBluetoothPoweredOn

**与业务无关，我们sdk自己实现**

c. isDebuging 开启时会打开日志输出

- (BOOL) isDebuging

方法名统一如下：

**enableDebug(BOOL) true 启用调试模式 false 关闭调试模式**

**BOOL isDebuging 判断当前是否是调试模式**

**9.发送振动**

/// \param callback 结果回调，返回命令是否发送成功

///

- (void)vibrateWithCallback:(void (^ \_Nonnull)(BOOL))callback;

**10.获取设备电量**

/// \param callback 若返回 0 ，则说明获取电量失败

///

-(void)fetchBatteryWithCallback:(void(^ \_Nonnull)(CGFloat))callback;

**11.屏幕显示方式**

/// 自动翻页，竖屏模式，高亮模式，左右手

/// \param config 配置内容

///

/// \param callback 设置是否成功的回调

///

-(void)setDisplayStyleWithConfig:(bongDisplayConfig \* \_Nonnull)config callback:(void (^ \_Nonnull)(BOOL))callback;

@interface bongDisplayConfig : NSObject

/// 自动翻页

@property (nonatomic) BOOL autoFlip;

/// 竖屏模式

@property (nonatomic) BOOL verticalMode;

/// 右手模式

@property (nonatomic) BOOL rightHand;

/// 高亮模式

@property (nonatomic) BOOL highlight;

/// 抬腕亮屏（bong4）

@property (nonatomic) BOOL liftup;

**12.配置消息提醒开关**

/// \param phoneCall 来电提醒开关

///

/// \param wechat 微信提醒开关

///

/// \param qq QQ 提醒开关

///

/// \param sms 短信提醒开关

///

/// \param callback 设置是否成功的回调

///

-(void)setNotificationWithPhoneCall:(BOOL)phoneCall wechat:(BOOL)wechat qq:(BOOL)qq sms:(BOOL)sms callback:(void (^ \_Nonnull)(BOOL))callback;

方法名统一如下：

setMsgNotification

**13.设置屏幕显示内容**

/// \param showHeartrate 心率测量

///

/// \param showWeather 天气显示

///

/// \param showSport 运动显示

///

**setScreenShowContent(boolean showHeartrate, boolean showWeather, boolean showSport, IBraceletCallback iBraceletCallback)**

**14.同步数据**

**15已包含该数据，可以去掉该接口**

/// \param callback 结果回调, 返回当天总结数据：步数，能量，距离，睡眠时长，若同步失败，会返回错误详情

///

* (void)syncWithCallback:(void (^ \_Nonnull)(bongSummary \* \_Nullable, NSError \* \_Nullable))callback;

@interface bongSummary : NSObject

/// 步数

@property (nonatomic) NSInteger steps;

/// 距离(单位米)

@property (nonatomic) NSInteger distance;

/// 能量（单位：焦尔）

@property (nonatomic) NSInteger energyInJ;

/// 能量 （单位：千卡）

@property (nonatomic, readonly) CGFloat energyInKCal;

/// 睡眠时长（分钟）

@property (nonatomic) NSInteger sleepMinutes;

/// 睡眠分数（质量）

@property (nonatomic) float sleepScore;

**15.获取某一天的总结数据**

**void getSportData(Date date, IBraceletCallBack<BraceletSport> iBraceletCallBack);**

**参数说明：**

**Date 日期类型的参数**

**BraceletSport 包含以下属性**

**private List<TimeLine> timeLineData;//** **运动时间线数据**

**private Summary sumData;//** **运动基本统计数据**

**TimeLine 时间线包含以下属性**

**private Long startMillis;//运动的开始时间**

**private Long endMillis;//** **运动的结束时间**

**private Float energy;//大卡**

**private Integer steps;//如果code表示的运动类型是散步，则为具体步数**

**private Integer code;//** **具体的运动类型 0 未佩戴手环 1 充电 2 浅睡眠 3 深睡眠 4 清醒 5 静 6 活动 7 散步 8 坐车 9 热身 10 健走 11 跑步 12 运动 13 游泳 14 自行车 15 健身 16 羽毛球**

**Summary运动基本统计数据保护眼以下属性**

**private Float distance;** **//行走距离**

**private Integer steps;//行走步数**

**private Float sleepScore;** **//睡眠质量，大于4.5为优，大于3小于4.5为良，3以下为一般**

**private Integer sleepMinutes;** **//睡眠时长，单位为分钟，最大值为480**

**private Integer energyinkcal;** **//消耗能量，单位为大卡，最大值200**

**IBraceletCallBack 手环通用回调接口，参见22.添加锁具关联**

/// \param date 时间，传入任意时间，会返回时间所在天的总结

///

/// \param compeletion 完成回调

///

- (void)readSummaryAt:(NSDate \* \_Nonnull)date compeletion:(void (^ \_Nonnull)(bongSummary \* \_Nullable))compeletion;

**16.获取当天的活动详情**

**15已包含该数据，可以去掉该接口**

/// \param date 时间，传入任意时间，会返回时间所在当天的 Activity

///

/// \param compeletion 完成回调

///

* (void)readActivityAt:(NSDate \* \_Nonnull)date compeletion:(void (^ \_Nonnull)(NSArray<bongActivity \*> \* \_Nonnull))compeletion;

@interface bongActivity : NSObject

/// 开始时间

@property (nonatomic, readonly, strong) NSDate \* \_Nonnull startTime;

/// 结束时间

@property (nonatomic, readonly, strong) NSDate \* \_Nonnull endTime;

/// 活动类型

@property (nonatomic, readonly) enum bongActivityType activityType;

/// 能量

@property (nonatomic, readonly) NSInteger energy;

/// 步数

@property (nonatomic, readonly) NSInteger steps;

/// 距离

@property (nonatomic, readonly) CGFloat distance;

/// 划臂次数 [ 暂未启用 ]

@property (nonatomic, readonly) NSInteger swings;

/// 圈数 [ 暂未启用 ]

@property (nonatomic, readonly) NSInteger cycles;

/// 活动是否已结束（最后一个活动，可能会继续。例如识别到用户现在正在跑步，但跑步可能还在继续，此种 activity 为 unfinished）

@property (nonatomic, readonly) BOOL finished;

/// 活动详情，每分钟的步数、能量、心率信息 <可能有>

@property (nonatomic, copy) NSArray<bongActivityDetail \*> \* \_Nonnull details;

**17.手环升级**

/// \param callback 返回命令发送是否成功的回调

///

- (void)enterDFUWithCallback:(void (^ \_Nonnull)(BOOL))callback;

方法名统一如下：

**//更新手环固件**

**void updateFireware(String braceletMac, String firewarePath, OnSimpleDfuProgressListener onSimpleDfuProgressListener);**

public interface OnSimpleDfuProgressListener {  
  
 */\*\*  
 \* 进入固件更新模式成功  
 \** ***@param*** *deviceAddress 设备mac地址  
 \*/* void inputFirewareUpdateMode(String deviceAddress);  
  
 */\*\*  
 \* 开始更新  
 \** ***@param*** *deviceAddress 设备mac地址  
 \*/* void onDfuProcessStarting(final String deviceAddress);  
  
 */\*\*  
 \* 更新完成  
 \** ***@param*** *deviceAddress 设备mac地址  
 \*/* void onDfuCompleted(final String deviceAddress);  
  
 */\*\*  
 \* 正在更新  
 \** ***@param*** *deviceAddress 设备mac地址  
 \** ***@param*** *percent 当前进度值  
 \** ***@param*** *speed 当前更新速度  
 \** ***@param*** *avgSpeed 平均更新速度  
 \** ***@param*** *currentPart 当前更新的片区索引  
 \** ***@param*** *partsTotal 总片区数  
 \*/* void onProgressChanged(final String deviceAddress, final int percent, final float speed, final float avgSpeed, final int currentPart, final int partsTotal);  
  
 */\*\*  
 \* 更新失败  
 \** ***@param*** *deviceAddress 设备mac地址  
 \** ***@param*** *errorCode 错误码  
 \*/* void onError(final String deviceAddress, final int errorCode);  
}

**18.设置当前目标设备**

/// \param mac MAC 地址

///

/// \param bindDate 绑定日期

///

- (void)configCurrentDeviceWithMac:(NSString \* \_Nonnull)mac name:(NSString \* \_Nonnull)name bindDate:(NSDate \* \_Nonnull)bindDate;

**19.设置久坐提醒**

/// \param standReminder 起身提醒

///

/// \param callback 设置是否成功的回调

///

* (void)setStandReminderWithStandReminder:(bongStandReminder \* \_Nonnull)standReminder callback:(void (^ \_Nonnull)(BOOL))callback;

@interface bongStandReminder : NSObject

/// 开关

@property (nonatomic) BOOL isOn;

/// 开始时间的小时

@property (nonatomic) NSInteger beginHour;

/// 开始时间的分钟

@property (nonatomic) NSInteger beginMinute;

/// 结束时间的小时

@property (nonatomic) NSInteger endHour;

/// 结束时间的分钟

@property (nonatomic) NSInteger endMinute;

/// 每周开启天的标记

/// <ul>

/// <li>

/// Bool 数组，依次为 sun. mon. the. wed. thu. fri. sat

/// </li>

/// </ul>

/// <blockquote>

/// 如：周一 ~ 五 开启，则为：[false, true, true, true, true, true, false]

///

/// </blockquote>

@property (nonatomic, copy) NSArray<NSNumber \*> \* \_Nonnull enableWeekDays;

**20.设置闹钟**

/// \param alarms 闹钟列表

/// \param callback 设置是否成功的回调

///

* (void)setAlarmsWithAlarms:(NSArray<bongAlarm \*> \* \_Nonnull)alarms callback:(void (^ \_Nonnull)(BOOL))callback;

@interface bongAlarm : NSObject

/// 闹钟 ID： 1 ~ 5

@property (nonatomic) NSInteger alarmID;

/// 是否开启

@property (nonatomic) BOOL isOn;

/// 懒人模式：

@property (nonatomic) BOOL lazyMode;

/// 小时

@property (nonatomic) NSInteger hour;

/// 分钟

@property (nonatomic) NSInteger minute;

/// 每周开启天的标记

/// <ul>

/// <li>

/// 依次为 sun. mon. thu. wed. thu. fri. sat. 的开启状态

/// </li>

/// </ul>

/// <blockquote>

/// 如：周一 ~ 五 开启，则为：[false, true, true, true, true, true, false]

///

/// </blockquote>

@property (nonatomic, copy) NSArray<NSNumber \*> \* \_Nonnull enableDaysInWeek;

/// 浅睡眠提醒秒数。会在闹钟时间前 x 秒内的浅睡眠提前振动。

/// 如设置了 600s，即为 10 分钟，闹钟时间在 8:00，则会在 7:50 ~ 8:00 之间的浅睡眠中振动

@property (nonatomic) NSInteger lightSleepSeconds;

**21.重启手环**

- (void)restart;

**22.添加锁具关联**

**void bindBracelet(ParamBind paramBind, IBraceletCallBack<ParamBindSuccess> iBraceletCallBack);**

**参数说明：**

**ParamBind 绑定手环传入的参数，包含以下属性：**

**private String Name;//手环名称，蓝牙扫描出来的名称**

**private String userAccount;//添加手环的用户账号**

**private String braceletMacAddress;//手环mac地址**

**private byte[] opensecretkey;//开门秘钥**

**private String relativedDeviceMac;//关联设备mac地址**

**private String relativedDeviceChannelPwd;//关联设备信道密码**

**ParamBindSuccess 手环绑定成功后回传的数据，包含以下属性：**

**private String passString;//开门秘钥字符串**

**private String name;//手环名称**

**private String userAccount;//用户账号**

**private String relativedDeviceMac;//关联设备的mac地址**

**private String braceletMacAddress;//手环的mac地址**

**IBraceletCallBack 手环回调接口，其他接口通用该回调接口**

**public interface IBraceletCallBack<T> {**

**//操作成功**

**void onSuccess(List<T> data);**

**//操作失败 errorCode 错误码 error 错误描述**

**void onFail(int errorCode, String error);**

**}**

**23.解除锁具关联**

**void unBindBracelet(ParamUnBind paramUnBind, IBraceletCallBack iBraceletCallBack);**

**参数说明：**

**ParamUnBind 解绑手环传入的参数，包含以下属性：**

**private String bongMacAddress;//Bong的mac地址**

**private byte[] bongPass;//要删除的Bong的开门秘钥**

**private String relationDeviceMac;//手环关联设备的mac地址**

**IBraceletCallBack 手环回调接口，参考22添加锁具关联**