

PaiBandRN 通讯协议

changelog

20170419

增加身高体重设置grow-setting

初始化参数page增加两个页面，sportChart sleepChart

20170411

修改js调用native第二个参数为必填回调方法

20170407

初始化参数加入page字段，指定初始打开的页面

20170328

加入初始化参数说明

加入NativeModules说明

20170327

增加同步数据协议

修改心率测定type错误 修改抬腕亮屏type错误

通讯方式

初始化参数

需要参数为：

appid 应用id

token 用户验证

uid 用户id

cid 当前选择的孩子id

page 需要显示的页面：运动数据 sport 睡眠数据 sleep 心率数据 heart 心愿小花 flower
成长曲线 grow 安全距离 security 手环闹钟 clock 手环设置 setting 运动数据统计
sportChart 睡眠数据统计 sleepChart

js侦听native推送的消息

消息名： PaiBandServiceEvent

消息体：

```
{
  type: 'xxxx', //消息类型
  subType: 'xxxx', //根据需要，可以扩充此字段，子类型
  data: {} //消息数据
}
```

实现：

```
//js实现侦听
NativeAppEventEmitter.addListener('PaiBandServiceEvent', event => console.log(event))
```

```
//oc发送消息
rtc_root_view.bridge.eventDispatcher().(withName: "PaiBandServiceEvent", body:
```

js调用native接口

native提供react-native的NativeModules实例PaiBandService，并暴露方法 `execute` 给js调用，需要传入回调方法作为此方法的第二个参数。

js引入 `NativeModules.PaiBandService` 进行接口调用

```
//nativeServiceModule为js需要引入的native模块，做为通讯服务
//需要暴露一个方法execute
//第一个参数为数据对象必填，第二个参数为回调方法
NativeModules.PaiBandService.execute(params, callback)
```

第一个参数为数据json对象(或字典类型)，第二个参数为回调方法。

第一个参数：

```
{
  type: 'xxxxx', //消息类型
  data: { //data字段可选
    'request_id': 'xxxx', //如果传入了此数值，有回调方法时，需要带上此数据，表示该次请求的id
  }
}
```

第二个参数：

```
/*
* 此回调方法有两个参数，第一个为错误error，第二个为数据data
* 当发生异常时，error为json对象，表示具体的异常
* error: { type: 'xxxx', message: 'xxxx' } type为异常类型，message为异常描述
* 当没有异常发生时，error为空(nil, null等)，data表示正常数据
* data: { type: 'xxxx', data: {} } 跟execute的第一个参数类似。
*/
callback(error, data)
```

业务协议

侦听菜单功能选择

```
{
  type: 'menu-select',
  data: {
    page: '' //
  } //消息数据
}
```

后台唤醒侦听

native发送事件:

```
{
  type: 're-awaken',
  data: {} //消息数据
}
```

蓝牙状态变化侦听

```
{
  type: 'bluetooth-status',
  data: {
    device_id: 'xxx',
    status: 0 //0未打开手机蓝牙 1已经连接 2连接中 3未找到指定设备
  }
}
```

```
}
```

调用身高体重设置

js调用native:

```
{
  type: 'grow-setting'
  data: { }
}
```

native回调js:

data:

```
{
  type: 'grow-setting',
  data: { }
}
```

异常error:

```
{
  type: 'grow-setting',
  message: 'xxxx'
}
```

获取手环系统信息

js调用native:

```
{
  type: 'system-info'
  data: {
    device_id: 'xxxx',
    request_id: 'xxxx'
  }
}
```

native回调js:

data:

```
{
  type: 'system-info',
  data: {
    device_id: 'xxx',
    version: '1.1.1', //手环主控版本号M1软件
    battery: 20, //0~100
    qrcode: '.....',
    sensor_version: 'xxxx', //传感器版本号M0系统
    sensor_mark: 0 //int 标记 0 M0 M1都安装 1 只装了M0 没有装M1
  }
}
```

异常error:

```
{
  type: 'system-info',
  message: 'xxxx'
}
```

打开安全距离监控

js调用native:

```
{
  type: 'out-safeDistance',
  data: {
    status: true
  }
}
```

native回调js:

data:

```
{
  type: 'out-safeDistance',
  data: {
    status: true
  }
}
```

获取IP

js调用native:

```
{
  type: 'get-ip',
  data: {}
}
```

native回调js:

data:

```
{
  type: 'get-ip',
  data: {
    city: 'xx',
    cityip: 'xxx.xx.xx.xx'
  }
}
```

error:

```
{
  type: 'get-ip',
  message: 'xxxx'
}
```

保存图片到相册

js调用native:

```
{
  type: 'save-pic',
  data: {}
}
```

打开app通知设置

js调用native:

```
{
  type: 'notification-open',
  data: {}
}
```

native回调js:

```
json
{
  type: 'notification-open',
  data: {
    status: 0 //0成功, 1失败
  }
}
```

录入身高体重

js调用native:

```
{
  type: 'update-height',
  data: {
    cid: 'xxxxxxx',
    height: 111 //cm
  }
}
```

```
{
  type: 'update-weight',
  data: {
    cid: 'xxxxxxx',
    weight: 111 //kg
  }
}
```

native回调js:

data:

```
{
  type: 'update-height',
```

```
data: {
  status: 0 //0成功, 1失败
}
```

```
{
  type: 'update-weight',
  data: {
    status: 0 //0成功, 1失败
  }
}
```

小红花

js调用native:

```
{
  type: 'flowers',
  data: {
    device_id: 'xxxxxx',
    current_num: 12, //当前小花数
    num_info: 5, //需要增加或者减少的小红花数量
    action: 'xxxx' //add_num 加 minus_num 减
  }
}
```

native回调js:

data:

```
{
  type: 'flowers',
  data: {
    status: 0 //0 1
  }
}
```

抬腕亮屏

js调用native:

```
{
```



```
type: 'bright-screen',
data: {
  device_id: 'xxxxx',
  begin_hour: 12, //开始的小时
  begin_minute: 5, //开始的分钟
  end_hour: 12, //结束的小时
  end_minute: 5, //结束的分
  close: true //true关闭, false打开
}
}
```

native回调js:

data:

```
{
  type: 'bright-screen',
  data: {
    status: 0 //0成功 1失败
  }
}
```

对手环进行更新

js调用native:

```
{
  type: 'ota-update',
  data: {
    device_id: 'xxxxx',
    ota_version: '1.1.1',
    ota_pack_urls: ['http://xxxxxx.com/xxxx.zip', '']
  }
}
```

native回调js:

data:

```
{
  type: 'ota-update',
  data: {
    device_id: 'xxxx',
    status: 0, //0 1
    update_state: 'load_pack', //load_pack write_pack restart
  }
}
```

```
        update_progress: 20,  //0~100
        version: '1.1.1',
        pack_name: ''
    }
}
```

关闭PaiBand页面

js调用native:

```
{
  type: 'close',
  data: {}
}
```

获取指定设备的蓝牙状态

js调用native:

```
{
  type: 'bluetooth-status',
  data: {
    device_id: 'xxxxx'
  }
}
```

native回调js:

data:

```
{
  type: 'bluetooth-status',
  data: {
    device_id: 'xxx',
    status: 0 //0未打开手机蓝牙 1已经连接 2连接中 3未找到指定设备
  }
}
```

开启手机蓝牙

js调用native:

```
{
  type: 'bluetooth-open',
  data: {}
}
```

native回调js:

data:

```
{
  type: 'bluetooth-open',
  data: {
    status: 0 //0成功 1失败
  }
}
```

实时心率测定

js调用native:

```
{
  type: 'heartrate-determine',
  data: {
    device_id: 'xxxx',
    cid: 123123
  }
}
```

native回调js:

data:

```
{
  type: 'heartrate-determine',
  data: {
    device_id: 'xxxx',
    cid: 213123,
    status: 0, //0 1
    heart_rate: 0
  }
}
```

手环数据同步

js调用native:

| sync_type | data_type | 操作方式 |
|-----------|------------|--------------------------------|
| 'write' | null/nil/无 | 写入手环grow clock heart_clock数据 |
| 'read' | null/nil/无 | 读取手环motion sleep heart数据并上报服务器 |
| 'write' | xxx | 写入手环xxx数据 |
| 'read' | xxx | 读取手环xxxt数据并上报服务器 |

```
type: 'bluetooth-sync'
data: {
  //可选
  request_id: 'xxxxx'
  //设备id
  device_id: 'xxxxx'
  //手环绑定的孩子id
  cid: 2132132
  /*同步类型write|read  write为写入手环的数据同步，包含grow成长数据(身高体重)、clock闹钟、heart_clock定时心率，read为读取手环的数据同步，包含motion运动数据、sleep睡眠、heart心率*/
  sync_type: 'write'
  /*数据类型 motion运动数据  sleep睡眠  heart心率  grow成长数据(身高体重)  clock闹钟  heart_clock定时心率。如果有此字段并不为空，则同步指定数据类型，否则同步sync_type指定的所有数据类型*/
  data_type: 'motion'
  data: {
    //sync_type为write data_type为grow孩子名字
    nickname: 'xxxx'
    //sync_type为write data_type为grow身高数据单位CM
    height: 120
    //sync_type为write data_type为grow体重数据单位KG
    weight: 40
    //sync_type为write data_type为clock工作日数据，无数据时不存在此字段
    clock: [
      {
        time: '10:23'
        //闹钟类型 getup起床 study学习 motion运动 sleep睡眠 other其他
        type: 'getup'
        //十进制整型，1字节，换算成二进制从高位到低位为对应的星期1-7
        //如11010000=225，表示周1、2、4重复
        //最后一位 始终为0
        repeat: 255
      }
    ]
  }
}
```

```

        }
        .....
    ]
    //sync_type为write data_type为heart_clock定时闹钟，无数据时不存在此字段
    clock_list: ['06:01','01:47']
}
}

```

native调用js:

```

type: 'bluetooth-sync'
data: {
    request_id: 'xxxx' //如果之前传入，则原样返回
    device_id: 'xxxxx' //设备id
    cid: 31234 //手环绑定的孩子id
    sync_type: 'xxx' //同步类型
    data_type: 'xxx' //数据类型
    status: 0 //0同步成功 1同步失败
}

```