健康云设计

# 1 背景

健康云与其他厂商之间的合作主要是将其他厂商的设备数据接入到京东，厂商分为有两类，一类是厂商自身已经有云端开放，一类是厂商直接使用京东云端存储，针对这两类现况，接入健康云主要有以下两种方式：

1. 通过厂商云端同步数据到京东健康云
2. 厂商直接将数据推送给京东，使用京东的数据

第1种接入方式的典型是咕咚。

接入咕咚数据的方式是云端同步，具体就是通过调用咕咚开放平台的API，获取到用户的健康数据。

接入流程为：

a)生成咕咚APP

b)咕咚接入京东账户体系，第一次登陆发送用户ID和Token相关信息

c)咕咚通知取数据

d)京东使用Token拉取数据

AppKey=2c417961a80811e38cbf5cac4cc1f077

AppSecret=f5c71a48a80711e38c1600163e020001

测试Token=498e884ee10c34a794fdd87fc85f0e28

第二种接入方式的典型是糖护科技。

糖护科技直接将数据通过接口调用推送给京东。

# 2 系统设计

## 2.1 整体架构

### 2.1.1 架构设计



智能设备



移动终端



ISV

云海

健康云

第三方云端



开发应用

开放

同步数据

用户登陆

授权

拉取

数据

上传数据

同步

查看

开发应用

推送数据

推送

数据

用户在厂商APP使用京东账号登陆后，有以下三种方式同步数据到健康云：

1. 上传数据到厂商云端，京东从厂商云端拉取数据
2. 上传数据到厂商云端，厂商云端推送数据到健康云
3. 上传数据到健康云

用户查看数据的方式：

1. 用户使用健康云的Web页面查看
2. 用户使用京东的APP查看
3. 用户使用厂商APP查看，数据来自于厂商或者京东

### 2.1.2 交互架构设计

京东与咕咚的数据同步在云端进行，通过云海开放数据，整体的一个交互架构如下：

咕咚开放平台

拉取数据

用户信息

健康数据

…

音频文件

云海

数据同步

API服务

数据服务

直接写入

土曼开放平台

…

推送用户ID和Token

更新用户Token

推送用户属性更新通知

获取更新后的用户属性

通知有新数据

接收通知

用户管理，京东统一用户管理

ISV

用户A

用户C

登陆

查看

开发应用

健康云

同步

调用API

数据分类拆分

Hbase

MySql

云盘

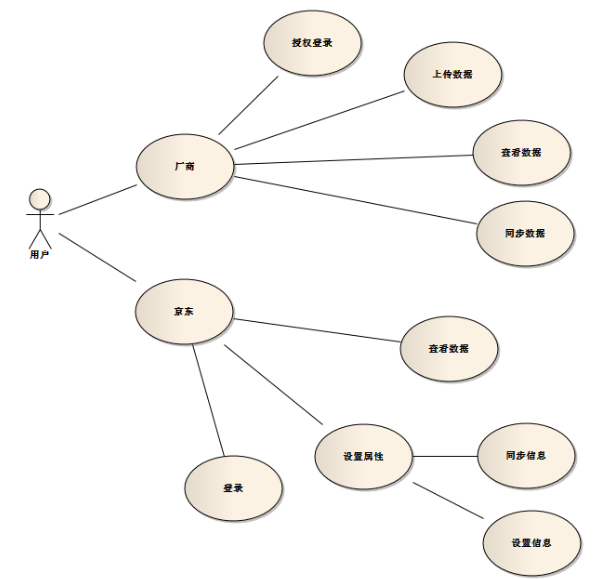
移动APP

Web展示

开放数据

# 3 模块设计

## 3.1 用例设计



用户可以通过京东和厂商使用健康设备，用户通过厂商应用使用设备的时候需要授权登录，然后将设备数据上传到厂商APP或者厂商云端，数据最终同步到健康云，用户查看数据可以从厂商读取数据，也可以从京东健康云中读取数据。用户可以通过京东应用查看数据，可以设置用户的属性信息。



管理对接入的数据进行类型的管理。

## 3.2 功能设计

健康云数据的来源方式主要有以下两种：

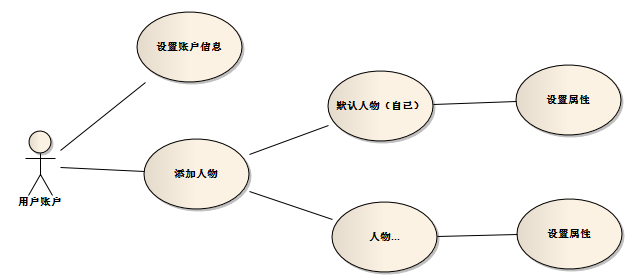
1. 由第三方通过Rest调用推送数据
2. 主动从第三方云端拉取数据

### 3.2.1 用户信息管理

用户信息管理包含用户账户人物信息、用户设备信息、用户应用信息、用户自定义数据。

#### 3.2.1.1 用户账户人物信息

##### 3.2.1.1.1 用例图



##### 3.2.1.1.2 功能描述

用户首先设置账户信息和本人信息。

用户账户信息为用户账户的基本信息，在一个用户账户下面可以添加人物信息，用户账户默认需要包含本人信息，此外可以添加与本人指定关系的人物。

用户可以使用京东账号在厂商APP上进行登录并且设置用户的账户信息和人物信息，用户也可以在京东上设置用户账户信息和人物信息，同时可以在京东上选择从其他厂商同步信息。

##### 3.2.1.1.3 处理流程

用户设置账户信息和本人信息流程。

用户

登录

设置信息

是否从其他厂商同步

选择厂商

获取数据填入表单

修改表单

提交数据

完成

是

否

用户添加人物流程。

用户

登录

添加人物

人物关系

设置人物信息

提交完成

#### 3.2.1.2 用户应用信息

##### 3.2.1.2.1 用例图



##### 3.2.1.2.2 功能描述

用户可以从厂商应用同步用户属性，需要记录用户使用的厂商应用详细信息。

用户可以通过厂商应用授权登录，此时表明该用户使用了该APP， 该用户登录了厂商APP后需要推送数据给京东，京东记录该APP的使用记录，有些应用推送过来的应用信息包含Token，如咕咚的。

厂商无APP的，默认为京东的一个APP，统一归类到京东APP。

##### 3.2.1.2.3 处理流程

用户

授权登录

推送信息到健康云

记录用户应用信息

记录App信息

APP表是否有该APP信息

否

是

是否使用过

否

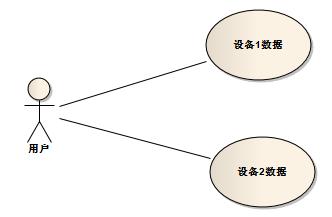
结束

是

#### 3.2.1.3 用户设备信息

只发生在需要区分具体设备的时候。

##### 3.2.1.3.1 用例图



##### 3.2.1.3.2 功能描述

用户数据需要区分具体的设备来源，在不区分的情况下则用户默认为无设备。

##### 3.2.1.3.3 处理流程

用户

登录

上传数据

是否分设备

记录应用数据

增加设备

首次上传?

是

是

否

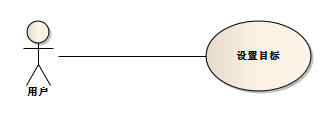
查询ID

否

#### 3.2.1.4 用户目标

用户会设置运动目标。

##### 3.2.1.4.1 用例图



##### 3.2.1.4.2 功能描述

用户会设置运动目标和体重目标，同时更新相关的体重身高信息。

##### 3.2.1.4.3 处理流程

用户

登录

设置目标

运动目标

插入Hbase

体重目标

更新Mysql

### 3.2.2 设备数据管理

#### 3.2.2.1 插入Hbase存储流程

数据

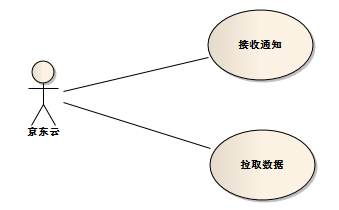
Mysql查询信息

组装Rowkey

插入数据

#### 3.2.2.2 拉取数据

##### 3.2.2.2.1 用例图



##### 3.2.2.2.2 功能描述

用户上传数据后，数据同步到厂商云端，厂商云端通知京东云有该用户的新数据，然后健康云从厂商拉取数据，存储到健康云。

##### 3.2.2.2.3 处理流程

用户

登录

上传数据

厂商云端

发送通知

健康云，区分数据来源，查询Token

拉取数据

插入记录

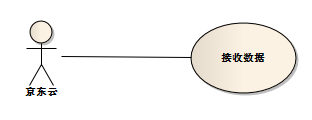
##### 3.2.2.2.4 数据示例

运动目标

|  |  |  |  |
| --- | --- | --- | --- |
| Rowkey | Column Family:g | | |
| swgsv | swgmv | swgcv |
| -880740843430871040(long) 1(int) -1(int) 1395300211775(long) 12(byte) 1(int) | 10000 | 10000 | 200000 |

#### 3.2.2.3 接收数据

##### 3.2.2.3.1 用例图



##### 3.2.2.3.2 功能描述

用户上传数据后，数据直接由厂商推送给健康云，健康云拆分存储。

##### 3.2.2.3.3 处理流程

健康云

接收数据

拆分数据

插入用户数据

##### 3.2.2.3.4 数据示例

|  |  |  |
| --- | --- | --- |
| Rowkey | Column Family:s | |
| bstt | bsgv |
| -880740843430871040(long) 1(int) -1(int) 1395300211775(long) 12(byte) 1(int) | 1 | 20 |

### 3.2.3 后台管理

1、管理接入的厂商APP

2、管理接入的数据类型

### 3.2.4 数据展示

# 4 认证设计

在对外提供Rest接口的时候需要认证调用方是否有权限，需要通过京东的认证方式来认证。京东认证方式有两种：

1. 通过JOS认证
2. 通过Boss认证方式

APP推送数据使用第一种认证方式，厂商服务端同步数据通过第二种认证方式。

目前接入健康云的应用通过人工审核的方式实现，接入合作方将JOS上申请的应用AppKey和应用名通过健康云后台管理加入到厂商APP管理中；合作方为可信用户。

## 4.1 Boss认证流程

服务端通过Boss的统一认证进行授权认证。



第三方应用

JOS

3 授权

4 获取Token

5 返回Token

健康云

6 Rest调用

7 AccessKey

SecretKey

认证

8 返回结果

9 返回处理结果

BOSS

1 获取Key

2 返回

AccessKey

SecretKey

存在的问题：

1. 需要第三方修改相关参数
2. 用户取消授权或者Token过期失效后，第三方能继续使用AccessKey和SecretKey继续推送数据到健康云

## 4.2 JOS认证流程

通过JOS的Token进行授权认证。



第三方应用

JOS

1 授权

2 获取Token

3 返回Token

健康云

4 Rest调用

5 接口调用

6 返回结果

7 返回处理结果

存在的问题：

1. JOS暂时不提供Token认证的接口
2. 需要第三方修改相关参数

# 5 数据库设计

## 5.1 Hbase与Mysql区别

数据存储采用Hbase，存储的数据有各健康设备同步的数据。

Hbase存储数据插入时只会生成实际存在的列，列本身不固定，Hbase和Mysql的区别如下所示。

1. MySql

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ID | A | B | C | D | E | F |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. Hbase

|  |  |  |  |
| --- | --- | --- | --- |
| RowKey | A | B | C |
|  |  |  |  |
|  | D | E | F |
|  |  |  |  |

## 5.2 数据形式

数据包含用户属性数据和用户设备数据。

### 5.2.1 用户属性数据

用户属性数据使用Mysql存储。

### 5.2.2 用户设备数据

数据第一次分类是按数据来源，数据有来源自咕咚的，有来源自土曼等的数据，同一个来源也有可能是多个人的。

数据在Hbase中使用一张表存储。

具体的分类如下：

数据

来源1

…

来源n

设备3

设备2

设备1

人物3

人物2

人物1

类型

## 5.3 用户数据设计

用户基本信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| ID | Integer | 否 | 自增 |
| user\_id | Bigint | 否 | 京东用户id |
| user\_name | Varchar(50) | 否 | 京东用户名最长不超过50 |
| real\_name | Varchar(50) | 是 | 用户姓名 |
| mobile\_phone | Varchar(15) | 是 | 手机号 |
| mail | Varchar(50) | 是 | 邮箱 |
| nick | Varchar(50) | 是 | 昵称 |
| address | Varchar(50) | 是 | 地址 |
| hobby | Varchar(250) | 是 | 爱好 |
| sports\_manifesto | Varchar(250) | 是 | 运动宣言 |
| routes\_count | Integer | 是 | 上传过的路线数 |
| followings | Integer | 是 | 关注数 |
| followers | Integer | 是 | 粉丝数 |
| time | Datetime | 否 | 插入时间 |

用户人物表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| ID | Integer | 否 | 自增 |
| user\_id | Bigint | 否 | 京东用户id |
| relationship | Varchar(10) | 否 | 与用户关系（自己、父亲等） |
| birthday | Date | 是 | 生日 |
| sex | Integer | 是 | 性别（0:女，1:男） |
| weight | Double | 是 | 体重 |
| height | Double | 是 | 身高 |
| portrait | Varchar(100) | 是 | 头像地址 |
| time | Datetime | 否 | 插入时间 |

厂商APP表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| ID | Integer | 否 | 自增 |
| app\_key | Varchar(50) | 否 | 应用标识 |
| app\_name | Varchar(50) | 否 | 数据厂商（咕咚等） |
| time | Datetime | 否 | 插入时间 |

用户应用表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| ID | Integer | 否 | 自增 |
| user\_id | Bigint | 否 | 京东用户id |
| app\_key | Varchar(50) | 否 | 应用标识 |
| client\_id | Varchar(50) | 是 | 京东在厂商申请的AppKey |
| token | Varchar(50) | 是 | 厂商云端Token |
| refresh\_token | Varchar(50) | 是 | 厂商云端Refresh\_token |
| Time | Datetime | 否 | 插入时间 |

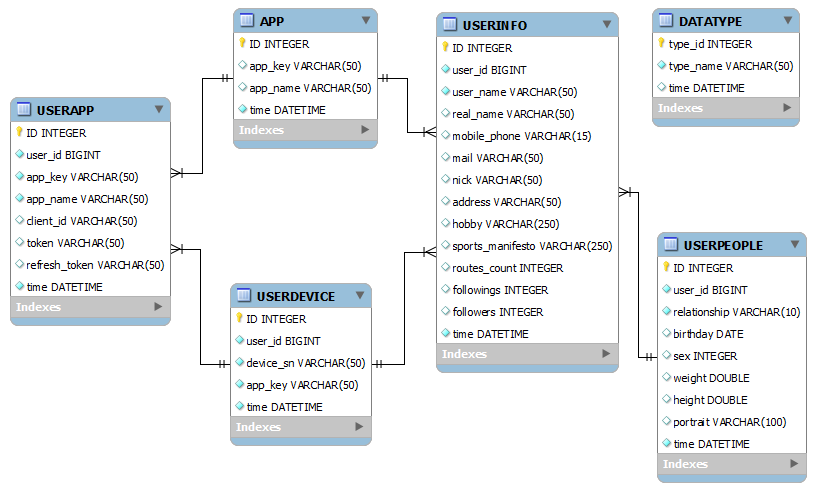
用户设备表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| ID | Integer | 否 | 自增 |
| user\_id | Bigint | 否 | 京东用户id |
| device\_sn | Varchar(50) | 否 | 设备序列号 |
| app\_key | Varchar(50) | 否 | 设备所属应用 |
| Time | Datetime | 否 | 插入时间 |

数据类型表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 可否为空 | 描述 |
| type\_id | Integer | 否 | 自增，类型ID |
| type\_name | Varchar(25) | 否 | 类型名 |
| Time | Datetime | 否 | 插入时间 |

用户个人信息表只对应账户，用户人物表是一个账户下的多个人物，用户应用表是一个账户下的多个应用，用户设备表是一个账户一个应用下的多个设备。具体关系如下：



## 5.4 健康数据设计

健康数据由于需要统计列表，同时数据量是随着时间增长而增长的，不同种类的健康数据的字段也不一致，使用Hbase存储健康类数据。

### 5.4.1 RowKey设计

RowKey包含以下四部分，用户ID，数据来源、数据类型、时间

1. 用户ID：即uid，long型(按位反) (8 byte)
2. 数据来源：APP\_ID(App\_key或者App\_name在厂商APP表中查询到的自增ID) + ID(int)(uid+device\_sn+app\_key查询用户设备表的ID，如果不区分设备则为-1) (8 byte)
3. 数据类型：Type\_id(2 Byte，类型ID Int转换为byte) + ID(int)(user\_id+relationship查询用户人物表的ID，人物为自己默认在该表中) (6 byte)
4. 时间：long (8 byte)

RowKey一共占用30 byte。

反转算法：

**public** **static** **long** reverse(**long** i){

i = (i & 0x5555555555555555L) << 1 | (i >>> 1) & 0x5555555555555555L;

i = (i & 0x0f0f0f0f0f0f0f0fL) << 4 | (i >>> 4) & 0x0f0f0f0f0f0f0f0fL;

i = (i & 0x00ff00ff00ff00ffL) << 8 | (i >>> 8) & 0x00ff00ff00ff00ffL;

i = (i << 48) | ((i & 0xffff0000L) << 16) | ((i >>> 16) & 0xffff0000L) | (i >>> 48);

**return** i;

}

### 5.4.2 Column Family

将数据分为三个列族，分别是目标数据、按天数据、单次数据。

g：目标

d：按天

s：单次

### 5.4.3 手环

运动目标

|  |  |
| --- | --- |
| "swgs":"String" | // 周目标步数(week\_goal\_steps) |
| "swgsv": Int | // 周目标步数的值(week\_goal\_steps\_value) |
| "swgm":"String" | // 周目标距离(week\_goal\_meters) |
| "swgmv": Int | // 周目标距离的值(week\_goal\_meters\_value) |
| "swgc": "String" | // 周目标卡路里(week\_goal\_calories) |
| "swgcv": Int | // 周目标卡路里的值(week\_goal\_calories\_value) |

|  |  |  |  |
| --- | --- | --- | --- |
| Rowkey | Column Family:g | | |
| swgsv | swgmv | swgcv |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

体重目标及历史

|  |  |
| --- | --- |
| "sgw": float | //目标体重(goal\_weight) |
| "sw": float | // 体重(weight) |
| "sh": float | // 身高(weight) |

|  |  |  |  |
| --- | --- | --- | --- |
| Rowkey | Column Family:g | | |
| swg | sw | sh |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

每天运动数据

|  |  |
| --- | --- |
| "smd": [1,2,3, ...] | //每十分钟的运动距离(sport\_meters\_day) |
| "std": [1,2,3, ...] | //每十分钟运动的步数(sport\_steps\_day) |
| "scd": [1,2,3, ...] | //每十分钟消耗的卡路里数(sport\_calories\_day) |
| "stcd": 0 | // 消耗卡路里(sport\_total\_calories\_day) |
| "stmd": 0 | // 运动距离(sport\_total\_meters\_day) |
| "stsd": 0 | // 步数(sport\_total\_steps\_day) |
| "stmid": 0 | // 运动时长(sport\_total\_minutes\_day) |
| "slid" | //最久不动时间(sport\_longest\_immovability\_day) |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rowkey | Column Family:d | | | | | | | |
| smd | std | scd | stcd | stmd | stsd | stmid | slid |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

单次运动

|  |  |
| --- | --- |
| "sstt": Int | // 运动持续时间，单位分钟(single\_sport\_total\_time) |
| "ssut": String | // 上传的时间，比如 "2013-09-21 18:02:58"( single\_sport\_upload\_time) |
| "sstc": Float | // 消耗卡路里(single\_sport\_total\_calories) |
| "ssst": String | // 运动开始时间，比如 "2013-09-23 17:40:14"( single\_sport\_start\_time) |
| "sst": String | // 运动方式 比如 走路、跑步、骑行(single\_sport\_type) |
| "ssri": String | // 线路图片地址(single\_sport\_route\_image) |
| "ssar": Int | // 是否达成目标 0 未完成， 1 完成(single\_sport\_activity\_result) |
| "sset": String | // 完成运动时间，比如"2013-09-24 17:41:07"( single\_sport\_end\_time) |
| "sstl": Float | // 运动距离(single\_sport\_total\_length) |
| "ssat": String | // 运动类型 比如：普通模式、目标模式、挑战模式、计划模式(single\_sport\_activity\_type) |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rowkey | Column Family:s | | | | | | | | | |
| sstt | ssut | sstc | ssst | sst | ssri | ssar | sset | sstl | ssat |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

每天睡眠数据

|  |  |
| --- | --- |
| "spd":[-1, -1, ...] | //每十分钟睡眠详细情况(sleep\_detail) |
| "spdm":60 | //深睡时间(sleep\_deep\_minutes) |
| "sptm": 80 | //总睡眠时间(sleep\_total\_minutes) |
| "splm": 20 | //轻睡时间(sleep\_light\_minutes) |
| "spe" | //有效睡眠效率(sleep\_efficiency) |
| "spst" | //睡眠开始时间(sleep\_start\_time) |
| "spet" | //睡眠结束时间(sleep\_end\_time) |
| "spdt" | //入睡耗时(sleep\_drop\_time) |
| "spat" | //清醒时间(sleep\_awake\_time) |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Rowkey | Column Family:d | | | | | | | | |
| spd | spdm | sptm | splm | spe | spst | spet | spdt | spat |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |

### 5.4.4 血糖

|  |  |
| --- | --- |
| " bsmt":long | //糖护测量时间(blood\_sugar\_ measure\_time) |
| " bstt":int | //测量时段 监测时段 早餐前:1 ，早餐后：2 ，午餐前：3，午餐后:4， 晚餐前：5， 晚餐后：6， 睡前：7.  (blood\_sugar\_time\_type) |
| " bsgv":double | //血糖值(blood\_sugar\_glucose\_value) |

|  |  |  |
| --- | --- | --- |
| Rowkey | Column Family:s | |
| bstt | bsgv |
|  |  |  |
|  |  |  |

### 5.4.5 血压

### 5.4.6 …

# 6 接口设计

## 6.1 外部接口

健康云会为接入的每一个应用开放两个统一的接口，一个接口用来推送数据到京东，一个接口用来从京东获取数据。每一个合作方使用这两个个统一的接口，通过参数区分具体的接口功能。

接口URL：https://smart.jcloud.com/health

|  |  |
| --- | --- |
| 方法名称 | 描述 |
| health\_reveive | 接收厂商推送数据 |

### 6.1.1 接收数据接口(Boss认证方式)

安全认证通过Boss的认证方式。认证信息包含：

**AccessKey**

共有密钥

Default: None

Required: Yes

**AppKey**

接入京东账户应用ID

Default: None

Required: Yes

**SecretKey**

京东云中获取到的secretkey

Default: None

Required: Yes

认证信息通过标准的HTTPS头Authorization来传递：

Authorization =AccessKey + ":" + AppKey+ ":" + SecretKey;

#### 6.1.1.1 血糖

推送血糖信息

|  |  |
| --- | --- |
| 请求方法 | POST |
| 请求URL | <https://api.cloud.jd.com/health> |
| 请求头 | 上述公用头部信息 |
| 请求参数  (JSON) | { "data-type":"blood-sugar ",  "blood-sugar-value": {  "user\_id": "17978992758",  "user\_name": "test",  "measure\_time": 1395648440560,  "time\_type": 1,  "glucose\_value": 100  },  "blood-sugar-goal": {  "user\_id": "17978992758",  "user\_name": "test",  "high\_before\_meal":100,  "low\_before\_meal":100,  "high\_after\_meal":100,  "low\_after\_meal":100,  "high\_night":100,  "low\_night":100,  }  }  若只传输血糖信息：  { "data-type":"blood-sugar ",  "blood-sugar-value": {  "user\_id": "17978992758",  "user\_name": "test",  "measure\_time": 1395648440560,  "time\_type": 1,  "glucose\_value": 100  },  "blood-sugar-goal": {}  }  若只传输目标信息：  { "data-type":"blood-sugar ",  "blood-sugar-value": {},  "blood-sugar-goal": {  "user\_id": "17978992758",  "user\_name": "test",  "high\_before\_meal":100,  "low\_before\_meal":100,  "high\_after\_meal":100,  "low\_after\_meal":100,  "high\_night":100,  "low\_night":100  }  } |
| 返回状态码 | 200 OK |
| 返回JSON | {  "status": "ok"  } |

#### 6.1.1.2 手环

推送user信息

|  |  |
| --- | --- |
| 请求方法 | POST |
| 请求URL | https://api.cloud.jd.com /health |
| 请求头 | 上述公用头部信息 |
| 请求参数  (JSON) | { "data-type":"user",  "user": {  "user\_id": "17978992758",  "user\_name": "user\_name",  "user\_token": "user\_token",  "user\_refresh\_token": "user\_refresh\_token"  }  } |
| 返回状态码 | 200 OK |
| 返回JSON | {  "status": "ok"  } |

推送通知

|  |  |
| --- | --- |
| 请求方法 | POST |
| 请求URL | [https://api.cloud.jd.com /health](https://smart.jcloud.com/health) |
| 请求头 | 上述公用头部信息 |
| 请求参数  (JSON) | { "data-type":"notify",  "notify": {  "catalog": "gps\_sports",  "end\_time": "2014-03-18 11:39:40",  "resource\_id": "route\_id",  "start\_time": "2014-03-18 11:39:40",  "user\_id": "user\_id"  }  } |
| 返回状态码 | 200 OK |
| 返回JSON | {  "status": "ok"  } |

推送手环数据

|  |  |
| --- | --- |
| 请求方法 | POST |
| 请求URL | [https://api.cloud.jd.com /health](https://smart.jcloud.com/health) |
| 请求头 | 上述公用头部信息 |
| 请求参数  (JSON) | { "data-type":"wrist-strap",  "wrist-strap-sport-goal": {  "week\_goal\_steps": int,  "week\_goal\_meters": int,  "week\_goal\_calories": int  }  "wrist-strap-body-goal": {  "goal\_weight": float,  "weight": float,  "height": float  }  "wrist-strap-sport-day": {  "sport\_meters\_day": [1,2,10…],  "sport\_steps\_day": [1,2,10…],  "sport\_calories\_day": [1,2,10…],  "sport\_total\_calories\_day":float,  "sport\_total\_meters\_day":float,  "sport\_total\_steps\_day":int,  "sport\_total\_minutes\_day":int,  "sport\_longest\_immovability\_day":int  }  "wrist-strap-sport-single": {  "single\_sport\_total\_time": int,  "single\_sport\_upload\_time": 2013-09-21 18:02:58 string,  "single\_sport\_total\_calories":float,  "single\_sport\_start\_time":2013-09-23 17:40:14 string,  "single\_sport\_type":int,  "single\_sport\_route\_image":string,  "single\_sport\_activity\_result":string,  "single\_sport\_end\_time":string,  "single\_sport\_total\_length":float,  "single\_sport\_activity\_type":string  }  "wrist-strap-sleep-day": {  "sleep\_detail": [1,2…],  "sleep\_deep\_minutes": int,  "sleep\_total\_minutes":int,  "sleep\_light\_minutes":int,  "sleep\_efficiency":float,  "sleep\_start\_time":2013-09-23 17:40:14 string,  "sleep\_end\_time":2013-09-23 17:40:14 string,  "sleep\_drop\_time":int,  "sleep\_awake\_time":int  }  "user\_id": "17978992758",  "user\_name": "user\_name"  }  若无数据则为空：  如wrist-strap-sleep-day无数据则为"wrist-strap-sleep-day":{} |
| 返回状态码 | 200 OK |
| 返回JSON | {  "status": "ok"  } |

### 6.1.2 接收数据接口(JOS认证方式)

#### 6.1.2.1 血糖

接收血糖信息

方法名：health\_reveive

系统级别输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| method | String | 是 | API接口名称 |
| access\_token | String | 是 | 采用OAuth授权方式为必填参数 |
| app\_key | String | 是 | 应用的app\_key |
| sign | String | 是 | 签名 |
| timestamp | String | 是 | 时间戳，格式为yyyy-MM-ddHH:mm:ss，例如：2011-06-16 13:23:30。京东API服务端允许客户端请求时间误差为6分钟 |
| format | String | 否 | 暂时只支持json |
| v | String | 是 | API协议版本，可选值:2.0. |

应用级输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 类型 | 是否必须 | 描述 |
| data-type | | String | 是 | 恒为blood-sugar |
| blood-sugar-value  可以为空，即{} | measure\_time | Long | 是 | 测量时间 |
| time\_type | Int | 是 | 时间类型 |
| glucose\_value | Int | 是 | 测量值 |
| blood-sugar-goal  可以为空，即{} | high\_before\_meal | Int | 是 | 餐前最高值 |
| low\_before\_meal | Int | 是 | 餐前最低值 |
| high\_after\_meal | Int | 是 | 餐后最高值 |
| low\_after\_meal | Int | 是 | 餐后最低值 |
| high\_night | Int | 是 | 睡前最高值 |
| low\_night | Int | 是 | 睡前最低值 |

#### 6.1.2.2 手环

接收血糖信息

方法名：health\_reveive

系统级别输入参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 类型 | 是否必须 | 描述 |
| method | String | 是 | API接口名称 |
| access\_token | String | 是 | 采用OAuth授权方式为必填参数 |
| app\_key | String | 是 | 应用的app\_key |
| sign | String | 是 | 签名 |
| timestamp | String | 是 | 时间戳，格式为yyyy-MM-ddHH:mm:ss，例如：2011-06-16 13:23:30。京东API服务端允许客户端请求时间误差为6分钟 |
| format | String | 否 | 暂时只支持json |
| v | String | 是 | API协议版本，可选值:2.0. |

应用级输入参数

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | | 类型 | 是否必须 | 描述 |
| data-type | | String | 是 | 恒为wrist-strap |
| wrist-strap-sport-goal可以为空，即{} | week\_goal\_steps | Int | 是 | 周目标步数 |
| week\_goal\_meters | Int | 是 | 周目标距离 |
| week\_goal\_calories | Int | 是 | 周目标卡路里 |
| wrist-strap-body-goal  可以为空，即{} | goal\_weight | Float | 是 | 目标体重 |
| weight | Float | 是 | 体重值 |
| height | Float | 是 | 身高值 |
| wrist-strap-sport-day  可以为空，即{} | sport\_meters\_day | String | 是 | 每10分钟的运动距离 |
| sport\_steps\_day | String | 是 | 每10分钟的运动步数 |
| sport\_calories\_day | String | 是 | 每10分钟的运动卡路里 |
| sport\_total\_calories\_day | Float | 是 | 一天消耗的总卡路里数 |
| sport\_total\_meters\_day | Float | 是 | 一天运动总距离 |
| sport\_total\_steps\_day | Int | 是 | 一天运动的总步数 |
| sport\_total\_minutes\_day | Int | 是 | 最久不懂时间 |
| sport\_longest\_immovability\_day | String | 是 | 每10分钟的运动卡路里 |
| wrist-strap-sport-single  可以为空，即{} | single\_sport\_total\_time | Int | 是 | 单次运动时间 |
| single\_sport\_upload\_time | String | 是 | 单次运动上传时间 |
| single\_sport\_total\_calories | Float | 是 | 单次运动消耗卡路里 |
| single\_sport\_start\_time | String | 是 | 单次运动开始时间 |
| single\_sport\_type | Int | 是 | 单次运动类型（走路、跑步、骑行） |
| single\_sport\_route\_image | String | 是 | 单次运动路线图地址 |
| single\_sport\_activity\_result | Int | 是 | 是否完成0否，1是 |
| single\_sport\_end\_time | String | 是 | 单次运动结束时间 |
| single\_sport\_total\_length | Float | 是 | 单次运动距离 |
| single\_sport\_activity\_type | String | 是 | 单次运动模式（普通、目标、挑战、计划） |
| "wrist-strap-sleep-day": {  可以为空，即{} | sleep\_detail | String | 是 | 每10分钟睡眠详情 |
| sleep\_deep\_minutes | Int | 是 | 深度睡眠时间 |
| sleep\_total\_minutes | Int | 是 | 睡眠总时间 |
| sleep\_light\_minutes | Int | 是 | 轻睡时间 |
| sleep\_efficiency | Float | 是 | 有效睡眠效率 |
| sleep\_start\_time | String | 是 | 睡眠开始时间 |
| sleep\_end\_time | String | 是 | 睡眠结束时间 |
| sleep\_drop\_time | Int | 是 | 入睡时间 |
| sleep\_awake\_time | Int | 是 | 清醒时间 |

## 6.2 内部接口