运动健康云端数据推送BI方案

# 数据推送范围

数据推送数据包括如下文档的红色部分：



在运动健康业务中涉及：数据云、活动系统、成就系统、社交系统、儿童手表 等几个子系统。

# 数据推送来源

数据推送来源包括：

* MySQL
* SDS

其中MySQL中数据，直接通过在dataPush工具配置sql语句，将数据直接从数据库中导出。

SDS中数据，因为SDS不支持将每日增量数据导出成文件形式，而BI也不支持直接从SDS中读取数据。因此，对于需要从SDS中导出的数据，需要运动健康业务系统生成日志文件的形式，通过dataPush工具将日志文件导给BI服务端。

# 数据导出细分

## 用户profile数据

用户profile数据直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

数据包括：

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| Sex | 性别 | 是 |
| Height | 身高 | 是 |
| Weight | 体重 | 是 |
| Birthdate | 出生日期 | 是 |

dataPush工具中配置脚本为：

select t.huid, t.gender, t.height, t.weight, t.unitType, t.birthday from t\_user\_basic\_info t where t.modifyTime < curdate() and t.modifyTime >= subdate(curdate() ,interval 1 day);

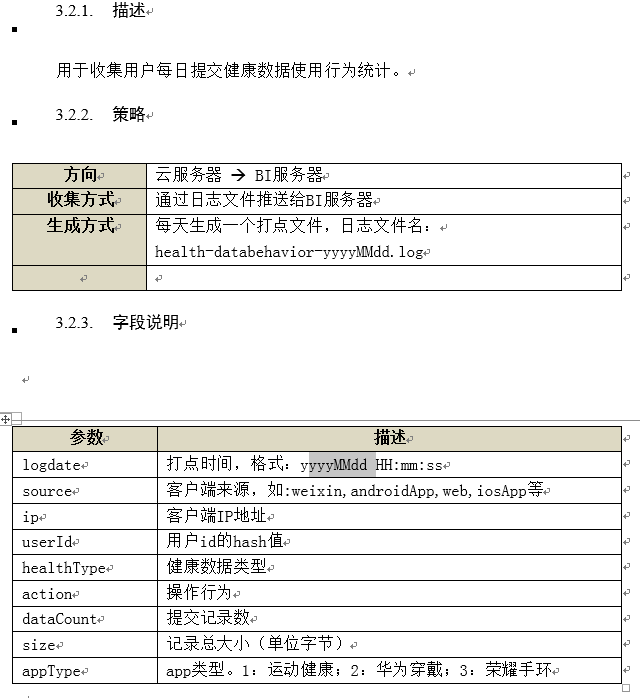
导出数据中还包含unitType字段，表示身高/体重的单位，如果字段为空则默认含义为0。

* 0：cm/kg（公制单位）
* 1：ft/lb（英制单位）

## App行为统计

此部分数据在6月份版本上线后已经提供日志文件推送给BI。

推送日志文件名为“health-databehavior-yyyyMMdd.txt”。参见如下文档说明：



需计算数据包括：

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| Data\_SYNC（BI计算得出） | 设备上报到云端的数据总次数(单日总次数) | 否 |
| APPID | APP唯一标识 | 否 |
| Client\_ip | 客户端IP地址(分析地域) | 否 |

BI根据“同一个uid + 同一个source + 同一天logdate”，计算有多少条数据，可以获得Data\_SYNC即设备单日总次数；

APPID即为日志文件中的source，即10131836、10217331等这些值；

Client\_ip为日志文件中的ip字段，为客户端真实的公网ip；

## 用户运动、睡眠数据汇总

用户运动数据汇总包含如下信息：

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| logdate | 数据记录时间，格式yyyyMMdd HH:mm:ss | 否 |
| steps | 当天汇总步数 | 是 |
| distance | 当天汇总里程，单位米 | 是 |
| calorie | 当天汇总卡路里，单位卡 | 是 |
| totalDuration | 当天睡眠总时长，单位分钟 | 是 |
| deepDuration | 当天深度睡眠时长，单位分钟 | 是 |
| lightDuration | 当天浅睡时长，单位分钟 | 是 |
| awakeTimes | 当天清醒次数，单位次 | 是 |

运动健康数据云每日生成文件，文件名为“sport-summary-yyyyMMdd.txt”。

日志文件的每一行格式为：

uid | logdate | 加密字符串

加密字符串为加密的json格式，示例如下。

{

“steps”: xxxx,

“distance”:xxxx,

“calorie”:xxx,

“totalDuration”: xxx,

“deepDuration”:xxx,

“lightDuration”:xxx,

“awakeTimes”:xxx

}

【注】有可能app只报运动数据，则json格式中只有steps、distance、calorie三个字段；有可能app只报睡眠数据，则json格式中只有totalDuration、deepDuration、lightDuration、awakeTimes四个字段；也有可能app运动和睡眠都报，则这七个字段都会有。

在实际业务场景中，一般一个app同一天会上报20次左右数据，这样同一个用户每天会有20条记录（如果用户多有个设备，这个条数还会翻翻）。

所以需要BI在处理时，针对同一个uid同一天，计算并保留当日的运动、睡眠的最大值。

运动的最大值，取多条数据中steps、distance、calorie每一项的最大值。

比如app1报了steps为1000，distance为800，calorie为300；app2报了steps为1100，distance为700，calorie为400，则计算得到的最大值为 steps：1100，distance：800，calorie：400。

健康的最大值，就取多条记录中totalDuration中最大的那条记录即可。

## 用户轨迹数据汇总

用户轨迹数据汇总包含如下信息：

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| logdate | 数据记录时间，格式yyyyMMdd HH:mm:ss | 否 |
| sportType | 轨迹运动类型。取值包括：  3：骑行  4：跑步  5：走路 | 否 |
| 通过计算得出 | 当天GPS运动次数（分类型：跑步、骑车、走路） | 不涉及 |
| totalTime | 当天GPS运动总时长（分类型：跑步、骑车、走路）。单位秒 | 是 |
| totalDistance | 当天GPS运动总距离（分类型：跑步、骑车、走路）。单位米 | 是 |
| totalCalories | 当天GPS运动卡路里（分类型：跑步、骑车、走路）。单位卡 | 是 |

运动健康数据云每日生成文件，文件名为“motionpath-summary-yyyyMMdd.txt”。

日志文件的每一行格式为：

uid | logdate | 加密字符串

加密字符串为加密的json格式，示例如下。

{

“sportType”:xxx,

“totalTime”: xxxx,

“totalDistance”:xxxx,

“totalCalories”:xxx

}

BI需要根据针对“同一个uid + 同一天 + 同一种轨迹类型sportType”，有多少条记录即可以对应的某种分类型GPS运动次数。

## 参加步步保总用户数

用户参加步步保信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| thirdAccountType | 三方账号类型，取值为：  4：新浪微博  7：QQ  22：微信  10001：薄荷网  10002：挂号网  10003：春雨医生  10004：众安步步保 | 否 |

dataPush工具中配置脚本为：

select t.huid, t.thirdAccountType from t\_third\_user t where t.lastModifyTime < curdate() and t.lastModifyTime >= subdate(curdate() ,interval 1 day);

BI基于每天导出数据计算每天有多少新增thirdAccountType取值为10004用户，即为参加步步保总用户数。

【注】同一个用户可能参加步步保，然后退出步步保，然后再参加步步保。BI需要能够适配这种场景进行处理。

## 用户参加活动信息

### 活动基本信息

活动基本信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| activityId | 活动ID | 否 |
| activityType | 活动类型（达标类、挑战类）  取值：  11: 达标赛--计步  12: 达标赛--里程  21: 排位赛--竞速  22: 排位赛--计步  23: 排位赛--里程  24: 排位赛--卡路里 | 否 |
| activityName | 参加活动名称 | 否 |
| description | 活动描述 | 否 |
| beginDate | 活动开始时间 | 否 |
| endDate | 活动结束时间 | 否 |

dataPush工具中配置脚本为：

select a.activityId, a.activityType, a.activityName, a.description, a.beginDate, a.endDate from t\_health\_activity a where a.status=1 and a.lastModifyTime < curdate() and a.lastModifyTime >= subdate(curdate() ,interval 1 day);

### 用户参加活动概要信息

用户参加活动信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| attendDate | 参加活动时间 | 否 |
| activityId | 活动ID | 否 |
| status | 活动参与状态  0:未完成  1:已完成  2:未中奖  3:已中奖  4:用户已领奖（中取实物奖品） | 否 |
| 通过计算得出 | 活动抽奖次数 | 不涉及 |
| 通过计算得出 | 中奖次数 | 不涉及 |

【注】活动抽奖次数和中奖次数根据其它值计算出来。

dataPush工具中配置脚本为：

select a.huid, a.attendDate, a.activityId, a.status from t\_health\_useractivity a, t\_health\_activity b where a.activityId = b.activityId and b.status =1 and a.modifyTime < curdate() and a.modifyTime >= subdate(curdate() ,interval 1 day);

BI需要根据status字段的取值计算“活动抽奖次数”和“中奖次数”信息。

当status为2或3或4时，计入用户抽奖次数中；

当status为3或4时，计入用户中奖次数中；

导出脚本中还包含activityId字段，标识是哪个活动。用于关联“达标活动达标详情”和“挑战活动达标详情”两个信息。

### 达标活动达标详情

用户参加达标活动时，比如活动开展周期为9月1号到9月30号，在活动期间只要有20天每天达到1万步即可以达标，需要记录在这一个月活动周期中每天是否达标的情况。

达标活动只有活动类型activityType为“11: 达标赛--计步”和“12: 达标赛--里程”两个取值的时候才会产生。

此信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| status | 当日活动达标状态  0:未完成  1:已完成 | 否 |

dataPush工具中配置脚本为：

select t.huid, t.activityId, t.status from t\_health\_userdailyreport t, t\_health\_activity b where t.activityId = b.activityId and b.status = 1 and t.modifyTime < curdate() and t.modifyTime >= subdate(curdate() ,interval 1 day);

导出脚本中还包含activityId字段，标识是哪个活动。用于和“用户参加活动概要信息”中活动ID关联。

### 挑战活动达标详情（从SDS表中获取）

用户参加挑战活动时，比如开展活动有效期在9月10号到9月20号的挑战10公里跑活动。会记录在活动有效期内用户每次跑步挑战是否成功。

挑战活动只有活动类型activityType为“21: 排位赛--竞速”、“22: 排位赛--计步”、“23: 排位赛--里程”和“24: 排位赛--卡路里”这几个个取值的时候才会产生。

此信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| status | 当日活动达标状态  0:未完成  1:已完成 | 否 |

导出脚本中还包含activityId字段，标识是哪个活动。用于和“用户参加活动概要信息”中活动ID关联。

运动健康数据云每日生成文件，文件名为“activity-sort-yyyyMMdd.txt”。

日志文件的每一行格式为：

uid | activityId| logdate | 加密字符串

加密字符串为加密的json格式，示例如下。

{

“status”: xxxx,

}

### 挑战活动名次详情

用户参加挑战活动时，比如开展活动有效期在9月10号到9月20号的挑战10公里跑活动。会记录在活动有效期内用户成绩最好的一次进行排名。

挑战活动只有活动类型activityType为“21: 排位赛--竞速”、“22: 排位赛--计步”、“23: 排位赛--里程”和“24: 排位赛--卡路里”这几个个取值的时候才会产生。

此信息直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| sort | 名次 | 否 |

dataPush工具中配置脚本为：

SELECT a.huid, a.activityId,count(\*) as rank

FROM t\_health\_activityrank a

INNER JOIN

t\_health\_activityrank b

ON a.activityId= b.activityId AND a.secs >= b.secs AND a.sportTime >= b.sportTime AND a.modifyTime < curdate() and a.modifyTime >= subdate(curdate() ,interval 1 day)

WHERE NOT EXISTS(

SELECT huid FROM t\_health\_activityrank WHERE a.activityId= b.activityId AND a.secs = b.secs AND a.sportTime = b.sportTime AND a.huid!= b.huid)

GROUP BY a.activityId,a.huid

ORDER BY a.activityId,a.secs ASC,a.sportTime ASC;

导出脚本中还包含activityId字段，标识是哪个活动。用于和“用户参加活动概要信息”中活动ID关联。

【注】活动系统中需要做修改，t\_health\_useractivity表、t\_health\_userdailyreport表、t\_health\_activityrank表中需要增加modifyTime字段，用于记录数据最后一次更改时间，基于这个时间导出每日增量数据给BI。

## 用户获得成就信息

### 用户获得勋章数据

用户获得勋章数据直接从MySQL中导出。每天导出增量数据，即前一天变化的用户数据。

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| 不涉及 | 我获取的勋章（各类型） | 否 |

dataPush工具中配置脚本为：

select t.huid, t.medalType, t.medalLevel, t.takeDate from t\_health\_usermedal t where t.modifyTime < curdate() and t.modifyTime >= subdate(curdate() ,interval 1 day);

其中medalType标识勋章类型，取值为：

* A:挑战勋章
* B:达标勋章
* C:累计勋章

medalLevel为勋章等级，取值大小代表等级。

takeDate为勋章的获得日。

【注】成就系统中需要做修改，t\_health\_usermedal表中需要增加modifyTime字段，用于记录数据最后一次更改时间，基于这个时间导出每日增量数据给BI。

### 用户获得卡卡信息

用户运动数据汇总包含如下信息：

|  |  |  |
| --- | --- | --- |
| 字段 | 字段解释 | 是否加密 |
| uid | 用户唯一标识 | 否 |
| logdate | 数据记录时间，格式yyyyMMdd HH:mm:ss | 否 |
| 通过计算得出 | 卡卡兑换次数 | 不涉及 |
| 通过计算得出 | 兑换卡卡额度（区分华为穿戴和运动健康） | 不涉及 |
| 通过计算得出 | 卡卡抽奖次数 | 不涉及 |
| 通过计算得出 | 抽奖消耗卡卡数（区分华为穿戴和运动健康） | 不涉及 |

运动健康数据云每日生成文件，文件名为“kaka-summary-yyyyMMdd.txt”。

日志文件的每一行格式为：

uid | logdate | 加密字符串

加密字符串为加密的json格式，示例如下。

{

“appType”: xxxx,

“reason”:xxxx,

“kaka”:xxx,

}

由于同一个用户一天可以兑换多次卡卡，并且可以进行多次抽奖。所以一个用户每天可能会有多条记录。

其中，appType标识为运动健康app或华为穿戴app，取值为：1—表示运动健康app；2—表示华为穿戴app；

Reason表示卡卡增减原因，取值为：

* 1：健康卡路里兑换
* 2：穿戴卡路里兑换
* 3：抽奖消耗
* 4：抽奖获得
* 5：参加活动
* 6：首次轨迹运动
* 7：发表评论
* 8：参与问卷调查
* 127：卡卡失效

Kaka为单次产生或消耗的卡卡值，可以取正值也可以取负值。

目前业务上Reason应该只会取值1，2或3，其它值暂时没有用到。

BI根据“同一个uid + 同一天 + Resaon为1和2”，计算用户每天卡卡兑换次数。

BI根据“同一个uid + 同一天 + appType + reason为1或2”，累加kaka字段，计算运动健康或华为穿戴每个用户每天兑换的卡卡数量。

BI根据“同一个uid + 同一天 + Resaon为3”，计算用户每天抽奖次数。

BI根据“同一个uid + 同一天 + appType + reason为3”，累加kaka字段，计算运动健康或华为穿戴每个用户每天抽奖消耗的卡卡数量。消耗卡卡时，kaka字段取值为负。

# 加密方法说明

Json加密字符串，通过AES128+CBC加密。IV为16字节，拼装在加密字符串前面。

具体代码实现可以参考如下代码：



json解密的密钥明文为：4632xup9357nmwp9