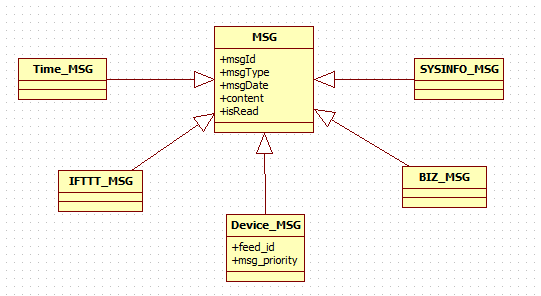
## 消息数据结构



注：msgId有一定业务规则。时间戳+2位随机数+业务类型。\*\*\*\*\*1表纯设备消息，\*\*\*\*\*2表系统通知，\*\*\*\*\*3表促销活动。由数据表中的msg\_type匹配。同时也是一个全局自增型LONG。

## 用户设置表

* 用户在超级APP是设置消息接收项。
* 数据库表：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 说明 | 例子 |
| user\_pin | Varchar(128) | 主键 | Lua\_aa |
| msg\_set | Varchar(128) | 消息设置。五位数分别对应  "device\_info"， "device\_warn"， "sys\_info"， "biz\_info"，"other\_info"。中间以逗号（半角）分隔 | 0,0,0,0,0 表不接收任何消息 |
|  |  |  |  |

注：此表可为扩展字段以适应用户其它设置项。

## 消息推送模块

### 纯设备消息推送

* 业务逻辑

当设备上报状态后，系统判断是否满足厂家自定义设备消息发送条件。如满足，则查询出此设备相关用户（绑定及分享关系）。逐一根据《用户设置表》筛选出发待推送送用户后，将消息体持久化至数据库《纯设备消息推送及内容表》。

如果用户在线，则发送；不在线，则存储在redis，设置消息失效期，待用户上线后推送。

注：设备一次可能上报同一steam多个时间点数据。消息触发只考虑最后一条记录。如果一次上报多个stream，triger了多个消息发送条件，则发多条。 如果同一纯设备消息发送给多个用户，msg\_id相同。

* Redis数据结构
* 用户待推送消息set： key为用户pin, value 为Set。存放用户待推送MSG ID。
* 消息ID： key 为MSG ID, value为空。有失效期。
* 产品消息推送条件设置表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 说明 | 例子 |
| Id | Long | 主键 | 自增 |
| product\_id | Long |  |  |
| msg\_name | Varchar128 | 消息名称 | 风速过高提醒 |
| msg\_priority | Smallint | 消息级别（0:提醒 1:告警） | 1 |
| msg\_dur | Long | 消息失效（分钟） | 60 |
| condition | Varchar1024 | Triger条件。4个或8个单词。中间以空格分隔。最多只支持二个条件 | If wind > 20 and swith == 0 |
| content | Varchar1024 | 内容 | 空调风速过高 |

* 纯设备消息推送及内容表HBASE：

Rowkey: userPing+feedid+msgId

列簇：f

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 说明 | 例子 |
| msg\_priority | Smallint | 消息级别（0:提醒 1:告警） | 1 |
| create\_date | Date | 消息产生日期 |  |
| push\_date | Date | 推送日期 |  |
| is\_read | Smallint | 是否已读 0未 1已读 | 0 |
| content | Varchar1024 | 内容 | 空调风速过高 |

注：纯设备消息无msg\_type列。缺省为0001。

* 非设备消息内容表

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 说明 | 例子 |
| msg\_id | Long |  | 全局自增型 |
| msg\_type | Char4 | 消息类型（系统通知0002, 促销活动0003…） | 0002 |
| create\_date | Date | 消息产生日期 |  |
| content | Varchar1024 | 内容 | 促销信息\*\*\*\*\* |

* 非设备消息推送表HBASE：

Rowkey: userPing+ +msgId

列簇：f

列：

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 类型 | 说明 | 例子 |
| msg\_id | Long |  | 全局自增型 |
| msg\_type | Char4 | 消息类型（系统通知0002, 促销活动0003…） | 0002 |
| create\_date | Date | 消息产生日期 |  |
| is\_read | Smallint | 是否已读 0未 1已读 | 0 |
| content | Varchar1024 | 内容 | 促销信息\*\*\*\*\* |

### IFTTT消息推送

当IFTTT触发控制设备后，系统拼装消息体，筛选出发待推送送用户后，持久化至数据库《非设备消息内容表》和《非设备消息推送表》。然后推送消息，处理逻辑同上。

### OPS群发

Smartcloud里开发一个群发消息的saf接口，OPS调用。

消息先持久化至《非设备消息内容表》。遍历在线用户，持久化至《非设备消息推送表》。然后推送。同时，为了在消息时效期内，后续登陆的用户也能收到此消息，群发消息需持久化至redis里。

* Redis数据结构
* 群发消息set： key为一固定值, value 为Set。存放群发消息的MSG ID。
* 消息ID： key 为群发消息的MSG ID, value为空。有失效期。

### 定时器消息推送

TBD。

## 用户上下线模块

用户在超级APP上线后，将上线信息添加至app\_login表中。即设置用户为上线状态。（每个用户只有一条记录）

随后处理用户待发送的消息：

系统查询redis中此用户待推送消息set。遍历set里的msg\_id是否在redis存在，存在即推送。如果是纯设备消息，则更新《纯设备消息推送及内容表》里推送时间；如果是非设备消息，则更新《非设备消息推送表》里推送时间。最后删除此set。

系统从redis的群发消息set里遍历出群发消息，逐一查看是否过期（在redis是否存在，不存在即从set里删除），然后查看《非设备消息推送表》是否已推送。未推送则推送，并添加《非设备消息推送表》里记录。

下线后删除上线记录。

## 消息中心模块

APP端根据user\_pin查询《非设备消息推送表》和《非设备消息内容表》。如果请求参数msg\_id为空。取最新的20条非设备消息。否则返回msg\_id后面的20条消息及其它值。参见接口。

## 纯设备消息查询模块

APP端根据user\_pin查询《纯设备消息推送及内容表》。如果请求参数msg\_id为空。取最新的20条非设备消息。否则返回msg\_id后面的20条消息。参见接口。

## 消息已读信息接口

APP调用后更新消息的已读字段。参见接口。

## OPS后面审核消息（待定）

## 消息感知等级及时效（待定）

由服务端还是客户端实现（待定）

## 业务说明

* IFTTT类消息，根据云端能获取的四种变量来定义文案如下：

一对一（触发成功）：您定义的“设备互联名称”被触发，当前“被触发设备名称”的状态为“状态值”，请查看。

一对一（触发失败）：您定义的“设备互联名称”未被触发，请检查您的”被触发设备名称“是否正常运行。

一对多（全部成功）：您定义的“设备互联名称”被触发，当前“被触发设备名称”的状态为“状态值”，“被触发设备名称”的状态为“状态值”。。。请查看。

一对多（部分成功）：您定义的“设备互联名称”未全部触发，设备名称”成功响应，状态为“状态值”，“未被触发设备名称”未响应，请检查设备是否正常运行。

一对多（全部失败）：您定义的“设备互联名称”未被触发，请检查您的”被触发设备名称“、”被触发设备名称“是否正常运行。

* 关于设备同时满足两个条件才被触发的情况，推送机制原则：只有检测到条件从不满足到满足的变化时，才给用户推送。比如：

空调厂家定义达到条件：温度>25 && 扫风=左右扫风 推送消息

当前温度24，扫风由上下扫风变成左右扫风，不推送，因为条件未同时满足

当前温度26，扫风由上下扫风变成左右扫风，推送，因为厂家定义的条件由不满足变成了满足

当前扫风为左右扫风，温度由24变成26，推送，因为厂家定义的条件由不满足变成了满足

当前扫风为左右扫风，温度由26变成27，不推送，因为厂家定义的条件一直是满足状态，未检测到变化。