Digital Twin基于数据实现，在TSDB数据基础上，通过时序数据来定义Digital Twin的属性，每个属性由一个时序作为数据源。

## 数据流格式定义

### Sensor通过MQTT到Leap Stream

设备上报MQTT的数据格式：

$LEAP/device/{deviceId}/stream/up

{

"timestamp": 1523848208232,

"profileVersion": 1,

"metric":"xxx.xxxx.xxxxx",

"reported": {

"sensor1": 25.98,

"sensor2": 22,

…

}

}

### Leap Stream转发给Web App

Leap stream转发的数据格式：

[{

"timestamp":1523848208232,

"profileVersion": 1,

"value":22.2,

"metric":"xxx.xxxx.xxxxx",

"tags":{

"deviceId":"xxxxx",

"tagk1":"tagv1",

…

}

},

…]

*约定：*

*约定tags中必须有一个tagk=”deviceId”，表示设备在系统中的唯一标识。反控设备时根据此值更新目标设备状态。*

### Leap Stream 存储到TSDB

## Digital Twin API定义

/digitaltwin

### 创建

/add.do

前端调用。

输入参数:

{

"name": "digital twin name",

"desc": "",

"attr": [{

"name": "xxx1",

"desc": "",

"unit": "",

"metric": "metric name1",

"tags": {

"tagk1": "tagv1",

"tagk2": "tagv2"

}

},

{

"name": "xxx2",

"desc": "",

"unit": "",

"metric": "metric name2",

"tags": {

"tagk1": "tagv1",

"tagk2": "tagv2"

}

}]

}

注意：

1. Digital Twin创建完成后自动根据Metric和Tags，查询TSDB对应时序的最新值，用以更新Digital Twin的属性值。

TSDB查询格式：

http://tsdbhost:port/api/query/last

{

"queries": [

{

"metric": "metric name",

"tags": {

"tagk1": "tagv1",

"tagk2": "tagv2"

...

}

}

],

"backScan":240

}

TSDB返回值：

[]

或

[

{

"timestamp": 1525685325123,

"value": "120.2",

"tsuid": "00000D00000100003600000200001B"

}

]

1. 限定Digital Twin的所有 deviceId只能来源于一个设备？不限制。

### 更新

#### 更新Digital Twin名称

/update.do

{

"id": 1,

"name": "digital twin name",

"desc": "",

"attr": [{

"id": 1,

"name": "xxx1",

"desc": "",

"unit": "xxx1",

"metric": "metricname1",

"tags": {

"tagk1": "tagv1",

"tagk2": "tagv2"

}

},

{

"id": 2,

"name": "xxx2",

"desc": "",

"unit": "xxx2",

"metric": "metricname2",

"tags": {

"tagk1": "tagv1",

"tagk2": "tagv2"

}

}]

}

#### 更新Digital Twin属性值（leap stream调用）

/update\_value.do

前端不能调用。

Digital Twin属性值是不能直接更新的，必须通过Leap Stream更新。

此接口供leap stream调用，Web server接收Leap Stream转发的流数据，通过此接口自动更新Digital Twin状态。

Metric + Tags唯一确定一个属性值。

Leap stream转发的数据格式：

[{

"timestamp":1523848208232,

"profileVersion": 1,

"value":22.2,

"metric":"metric name",

"tags":{

"deviceId":"xxxxx",

"tagk1":"tagv1",

…

}

},

…]

### 删除

#### 删除整个Digital Twin

/delete.do

根据id删除Digital Twin及其所有属性。

传入参数：

{"id": 1}

### 查询

/get.do

根据主键查询单个Digital Twin

传入参数：

{"id": 1}

/get\_list.do

根据关键字查询Digital Twin列表（不包含属性）

传入参数：

{"name": ""}

### 反控

/set.do

传入参数：

{

"id": 1,

"attr": [

{

"id": 1,

"value": 11

},

{

"id": 2,

"value": 12

}  
  
]

}

设置Digital Twin值，将该值更新到设备。

1. 首先更新Digital Twin表的“期望值”
2. 然后通过MQTT发送消息：

$LEAP/control/{deviceId}/digitaltwin/set

{

"metric":"xxx.xxxx.xxxxx",//

"desired": {

" xxx ":11,//tag{ lable :xxx}

"sensor2":12

}

}

## Digital Twin表结构

主表

Id

Digital Twin名称**(varchar 128)**

描述

创建时间（long）

属性子表

Id

fk

**属性名称(varchar 128)**

描述

单位(varchar 32)

属性值（Double）

期望值（Double）

创建时间（long）

更新时间（long）

Metric**(varchar 256)**

TagKV列表**(varchar 1024)**