Catalog interface for microservices

DO NOT CONSIDER THIS FILE, I have to re-do it.

### GET

Get information from the catalog about the sensors’ measurements in the rooms and optimal values for the dosing of nutrients for the soil, humidity, and temperature of each kind of plant.

To simplify, the sensors on the catalog are updated all at the same time, so the timestamp will be one.

* **Read single sensor value on single shelf: rooms/roomID/towerID/shelfID/sensorName**

Example: rooms/R1/T2/S4/pH

{

“pH” : 7 # sensor value

“t” : 1234 # timestamp

}

* **Read the value of all sensors on a shelf: rooms/roomID/towerID/shelfID/# or /roomID/towerID/shelfID/\***

Example: rooms/R1/T2/S4/#

{

"typeID ": "L",

"typeName ": "Lattuce",

"plantID ": "P1",

"plantName ": "Iceberg lattuce",

"status ": "Seeding",

"height ": 10,

"N": 0.98,

"P" : 2.16,

"K" : 1.02,

"soilMoisture": 0.60,

"pH ": 6.5,

"t ": 1234}

* **Read all sensors of a tower: rooms/roomID/towerID/#**

Example: rooms/R1/T2/#

{

“shelves” : [

{

“shelfID” : “S1”,

“typeName” : “Lattuce”,

“typeID” : “L”,

“plantID” : “P1”,

“plantName” : “Iceberg lattuce”,

“sensors” : [

“status” : “Seeding”,

“height” : 10,

“N” : 0.98,

“P” : 2.16,

“K” : 1.02,

“soilMoisture” : 0.60, # 60%

“pH” : 6.5,

“t” : 1234

]

},

{

“shelfID” : “S2”,

…

},

{

“shelfID” : “S3”,

…

}

}

* **Read all sensor of one type in room: rooms/R1/#/#/sensorName or rooms/R1/roomSensorName**

Example: rooms/R1/#/#/pH

{

“towers” : [

{

“towerID” : “T1”,

“shelves” : [

{

“shelfID” : “S1”,

“pH” : 7.0,

“t” : 1234

}

{

“shelfID” : “S2”,

“pH” : 6.0,

“t” : 1234

}

…

},

{

“towerID” : “T2”,

“shelves” : […]

}

…

}

Example: rooms/R1/humidity

{

“humidity” : “0.6” # sensor value 60%

“t” : 1234 # timestamp

}

Example: rooms/R1/ventilation

{

“ventilation” : “ON”

}

* **Read all sensors of a room: /roomID/\***

Example: rooms/R1/\*

{

“CO2” : 500

“humidity” : 0.6

“temperature” : 18.2

“vetilation” : “OFF”,

“towers” : [

{

“towerID” : “T1”,

“shelves” : [

{

“shelfID” : “S1”,

“sensors” : [

“status” : “Seeding”,

“height” : 10,

“N” : 0.98,

“P” : 2.16,

“K” : 1.02,

“soilMoisture” : 0.60, # 60%

“pH” : 6.5,

“t” : 1234

},

{

}

]

}

{

“towerID” : “T2”,

“shelves” : […]

}

]

}

* **Get preset values from the database: plantTypes/typeID/state/valueName**

Example: /plantTypes/L/Seeding/N

{

“N” : 0.95

}

Example: /plantTypes/L /#/N

{

“states” : [

{

“state” : “Seeding”

“N” : 0.95

},

{

“state” : “Vegetative”

“N” : 1.89

},

{

“state” : “Mature”

“N” : 2.84

}

]

}

Example: /plantTypes/L/\*

{

“typeName” : “Lattuce”,

“plants” : [

{

“plantID” : “P1”,

“plantName” : “Iceberg lattuce”

},

{

“plantID” : “P2”,

“plantName” : “Butterhead lattuce”

}

]

“vegetativeH” : 10

“matureH” : 20

“humidityDay” : 0.5

“humidityNight” : 0.5

“lowPH”: 0.5

“highPH : 0.5

“states” : [

{

“state” : “Seeding”,

“N” : 0.95,

“P” : 0.38,

“K” : 0.76,

“liters” : 1.89,

“light” : 17,

“soilMoisture” : 0.6

},

{

“state” : “Seeding”,

“N” : 0.95

},

{

“state” : “Seeding”,

“N” : 0.95

}

]

}

Example /plantTypes/L /lowPH

{

“lowPH” : 6

}

* **Get information of a plant, name, type and where is plantes: /plants/plantID**

Tell me if more information are needed (like the sensors)

Example: /plants/P1

{

“plantName” : “Iceberg lattuce”,

“typeID” : “L”,

“typeName” : “Lattuce”

“rooms” : [ # rooms where the plant is present

{

“roomID” : “R1”

“towers” : [ # tower where the plant is present

{

“towerID” : “T1”,

“shelves” : [ # shelves where the plant is present

{

“shelfID” : “S1”

},

{

“shelfID” : “S3”

}

…

}, …

}, …

}

### POST

Add new **plants** or **data** to the database

* **Add a new plant: /plants**

**Payload:**

**{**

**“plantsID” : “P2”, # empty (““) if not specified, it will be automatic**

**“plantName” : “Iceberg lattuce”,**

**“typeName” : “Lattuce”**

**}**

Returns:

{

“plantID” : “P1”,

“typeID” : “L”

}

* **Add new type of plant with nutrients etc.: /plantTypes**

**Payload:**

**{**

**“typeID” : “L”, # leave empty if you don’t know it**

**“typeName” : “Lattuce”,**

**“vegetativeH” : 10,**

**“matureH” : 20,**

**“humidityDay” : 0.5,**

**“humidityNight” : 0.5,**

**“lowPH” : 6,**

**“highPH” : 6**

**“states” : [**

**{**

**“state” : “Seeding”,**

**“N” : 0.95,**

**“P” : 0.38,**

**“K” : 0.76,**

**“liters” : 1.89,**

**“light” : 17,**

**“soil moisture” : 0.6**

**},**

**{**

**“state” : “Vegetative”,**

**…**

**},**

**{**

**“state” : “Mature”,**

**…**

**}**

**]**

**}**

Response:

{

“typeID” : “L”

}

* **Add a new room: /rooms (no payload, ID is created automatically)**

Returns:

{

“roomID” : “R3” # R1 and R2 already exists

}

* **Add a new tower in a room /rooms/roomID (no payload)**

Example: /rooms/R2

Returns:

{

“towerID” : “T3”

}

* **Add a new shelf /rooms/roomID/towerID**

Example: /rooms/R2/T1 (no payload)

Returns:

{

“shelfID” : “S4” # for example, S1,S2,S3 were already present

}

**Add a new device (raspberryPi) to the database: /rooms**

**Payload:**

**{**

**“deviceName” : “RaspberryPi”**

**}**

Response:

{

“roomID” : “R1”,

“

### PUT

Replace values in the database, if nonexistent it creates new resources. The measurements are taken from the device connector, so they cannot be replaced by other microservices. Microservices can change:

* **Change name or characteristics of a plant /plants/plantID (or plants/# or plants/\* if you don’t know the ID but only the name)**

**Payload: # leave empty the values if you don’t want to change them**

**{**

**“plantID” : “P1”, # leave empty if you don’t know it**

**“plantName” : “Iceberg Lattuce”, # can leave empty if there is plantID**

**“typeName” : “Lattuce new name”,**

**“typeID” : “LN”**

**}**

If plantID or plantName is not found in the database a new plant will be created, otherwise it will be updated with the new name/type etc.

Returns:

{

“plantID” : “P1”,

“plantName” : “Iceberg Lattuce”,

“typeName” : “Lattuce new name”,

“typeID” : “LN”

}

* **Change values of the nutrients /plantTypes/typeID or /plantTypes/\* or # if you don’t know the ID**

**Payload:**

**{**

**“typeID” : “L”, # leave empty if you don’t know it**

**“typeName” : “Lattuce”,**

**“vegetativeH” : 10,**

**“matureH” : 20,**

**“humidityDay” : 0.5,**

**“humidityNight” : 0.5,**

**“lowPH” : 6,**

**“highPH” : 6**

**“states” : [**

**{**

**“state” : “Seeding”,**

**“N” : 0.95,**

**“P” : 0.38,**

**“K” : 0.76,**

**“liters” : 1.89,**

**“light” : 17,**

**“soil moisture” : 0.6**

**},**

**{**

**“state” : “Vegetative”,**

**…**

**},**

**{**

**“state” : “Mature”,**

**…**

**}**

**]**

**}**

Specify every value that needs to change. No need to put everything.

### DELETE

* **Delete a plant: /plants/plantID** (tell me if you need to delete it based on the name)

The plant will be deleted and the shelves with that plant will be emptied (needs to return which shelves?)

* **Delete an entire type: /plantTypes/typeID** (deletes also the corresponding plants)
* **Delete an entire room: /rooms/roomID**
* **Delete an entire tower: /rooms/roomID/towerID**
* **Delete a shelf: /rooms/roomID/towerID/shelfID**