Topics

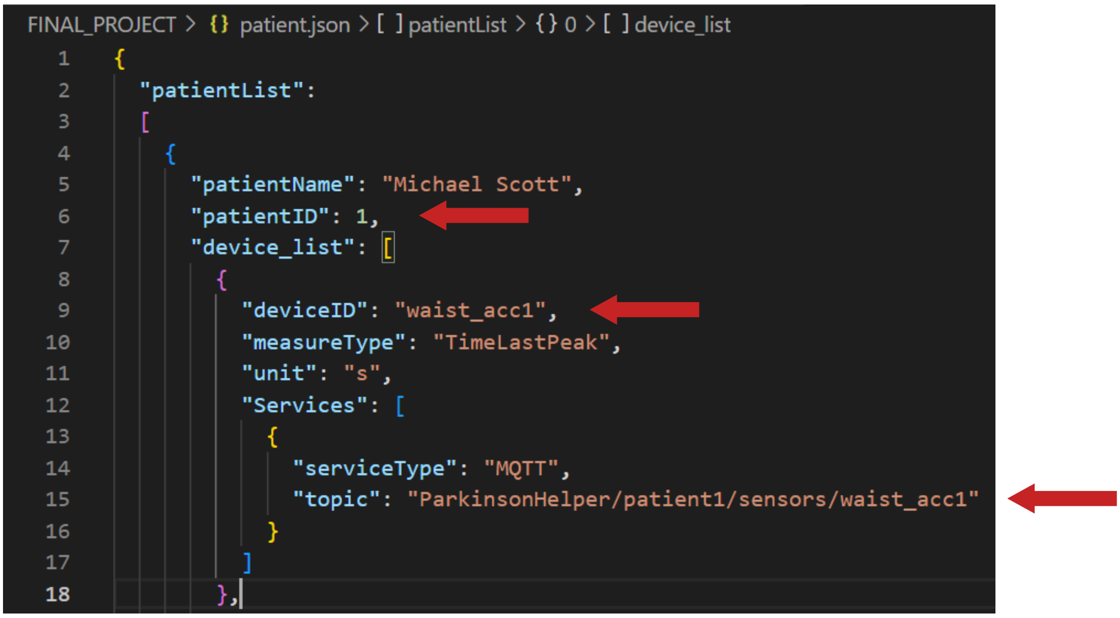
SENSORS publish under the topics:	Subscribers:
/ParkinsonHelper/patient*/sensors/wrist_acc* /ParkinsonHelper/patient*/sensors/waist_acc* /ParkinsonHelper/patient*/sensors/pressure *	Microservices are subscribed to these topics: /ParkinsonHelper/+/sensors/+
MICROSERVICES publish under the topics:	
/ParkinsonHelper/patient*/actuator/fall	➔ Emergency SMS (TeleBot) + ThingSpeak adaptor
/ParkinsonHelper/patient*/actuator/freezing	➔ DBS activation + ThingSpeak adaptor
/ParkinsonHelper/patient*/actuator/tremor	➔ Sound Alarm + ThingSpeak adaptor
/ParkinsonHelper/patient*/actuator/statistics	➔ ThingSpeak adaptor

* is referring to the patient ID

Microservices manages all the patient, so it is necessary to use the wildcard "+". It is also necessary to have the patientID and the sensorID in the message, in order to know whose measurements they are and publish under the correct topic.

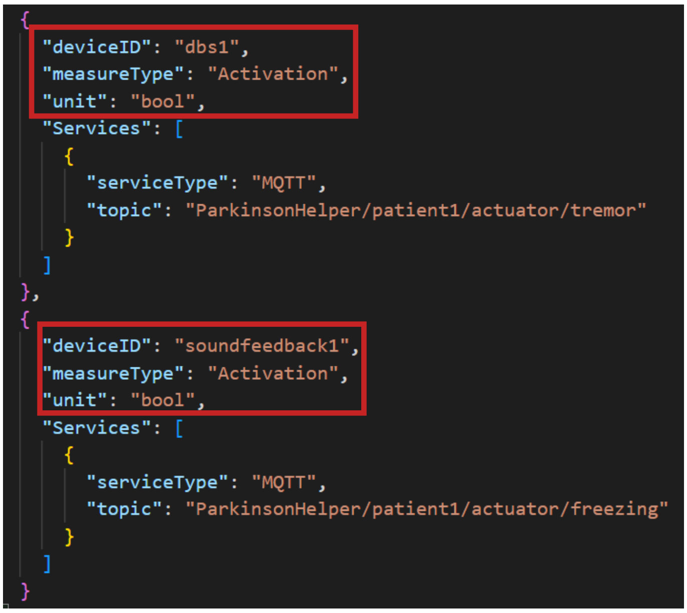
Since is NOT possible to have topics or addresses written in the script and it is necessary to collect them via REST:

- SENSORS have to collect the topic to use for publishing from the patient.json (patient.json is our STATIC catalog)
  Ex of request: localhost:8080/get_topic/patient1/waist_acc

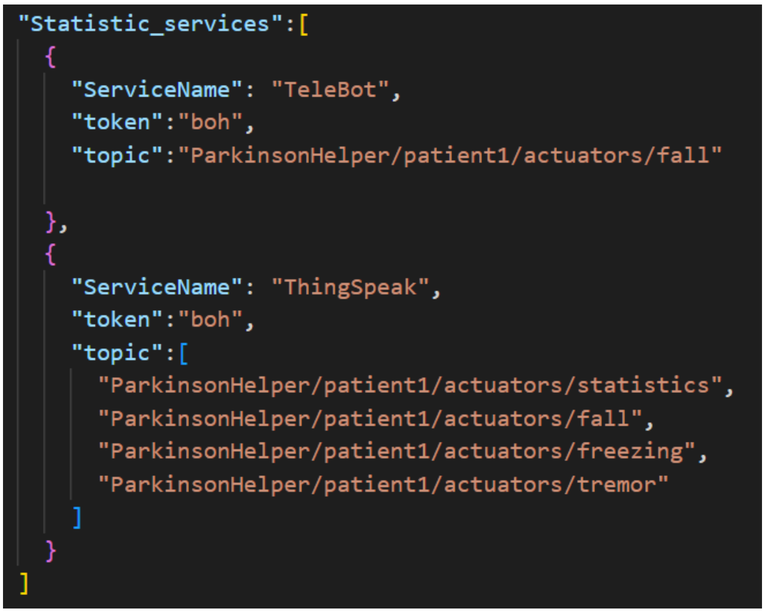

FINAL_PROJECT > {} patient.json > [ ] patientList > {} 0 > [ ] device_list

```
{
  "patientList":
  [
    {
      "patientName": "Michael Scott",
      "patientID": 1,
      "device_list": [
        {
          "deviceID": "waist_acc1",
          "measureType": "TimeLastPeak",
          "unit": "s",
          "Services": [
            {
              "serviceType": "MQTT",
              "topic": "ParkinsonHelper/patient1/sensors/waist_acc1"
            }
          ]
        },
```

- MICROSERVICES will have to do the same thing. The topics for freezing and tremor are in the patient.json catalog (actuators are also devices, so they are in the device list for each patient). MAYBE we should add under "services" also ThingSpeak and TeleBot, in order to add the topics for these services (also because they are services for the patient). An alternative, is to create another list outside "device_list" called maybe "statistics_services" and add TeleBot and ThingSpeak (because they are used to monitor the historical recordings of the patient.)



```
{
  "deviceID": "dbs1",
  "measureType": "Activation",
  "unit": "bool",
  "Services": [
    {
      "serviceType": "MQTT",
      "topic": "ParkinsonHelper/patient1/actuator/tremor"
    }
  ]
},
{
  "deviceID": "soundfeedback1",
  "measureType": "Activation",
  "unit": "bool",
  "Services": [
    {
      "serviceType": "MQTT",
      "topic": "ParkinsonHelper/patient1/actuator/freezing"
    }
  ]
}
```



```
"Statistic_services":[
  {
    "ServiceName": "TeleBot",
    "token":"boh",
    "topic":"ParkinsonHelper/patient1/actuators/fall"

  },
  {
    "ServiceName": "ThingSpeak",
    "token":"boh",
    "topic":[
      "ParkinsonHelper/patient1/actuators/statistics",
      "ParkinsonHelper/patient1/actuators/fall",
      "ParkinsonHelper/patient1/actuators/freezing",
      "ParkinsonHelper/patient1/actuators/tremor"
    ]
  }
]
```

Catalog

Type	**STATIC CATALOG**	**SERVICE CATALOG**	**REGISTER CATALOG**
Name	"patient.json"	"service_catalog.json"	"register_catalog.json"
Function	This is the catalog with all the info about the topics, services and IDs for each patient. This is the catalog that provide the topics. It communicates with REST	This is the catalog that has all the active services registered inside. Every two minutes, it checks if the service is still active and update the "lastUpdate" voice for each service (catalog_manager, statistic_manager, fall_manager…). THE FIRST THING EACH SERVICE HAS TO DO IS TO REGISTER TO THE SERVICE CATALOG.	This is the catalog that has all the active sersors and actuators registered inside. Every two minutes, it checks if the service is still active and update the "lastUpdate" voice for each device (waist_acc1, wrist_acc1,…). THE FIRST THING EACH DEVICE HAS TO DO IS TO REGISTER TO THE REGISTER CATALOG.

Every time a new patient is added, sensors, actuators, topics and statistic_services have to be created and added.