Data Ingestion Tutorial

Data Ingestion Step

- **HOW TO Create an IOT Device Model:** https://www.snap4city.org/591
- **HOW TO: Create an IOT Device Instance**: https://www.snap4city.org/590

  [Entity/IoT Directory tool - IotAPP]
- **HOW TO Develop an IOT Application for Data Ingestion** https://www.snap4city.org/593
- **HOW TO Managing Notifications on IOT Application** https://www.snap4city.org/142

Data Ingestion Example

Open Weather data, **Open Pollution data** and **Open Sea Condition data** for each pilot.

Data Source	Data Source Main Address	Periodicity for Data ingestion	Variables
Open Weather	https://openweathermap.org/api	30 mins	• airHumidity • airTemperature • cloudCoverPerc • feelsLike • groundLevel • maxTemperature • minTemperature • pressure • seaLevel • sunrise • sunset • visibility • windDirection • windSpeed

Data Ingestion Example

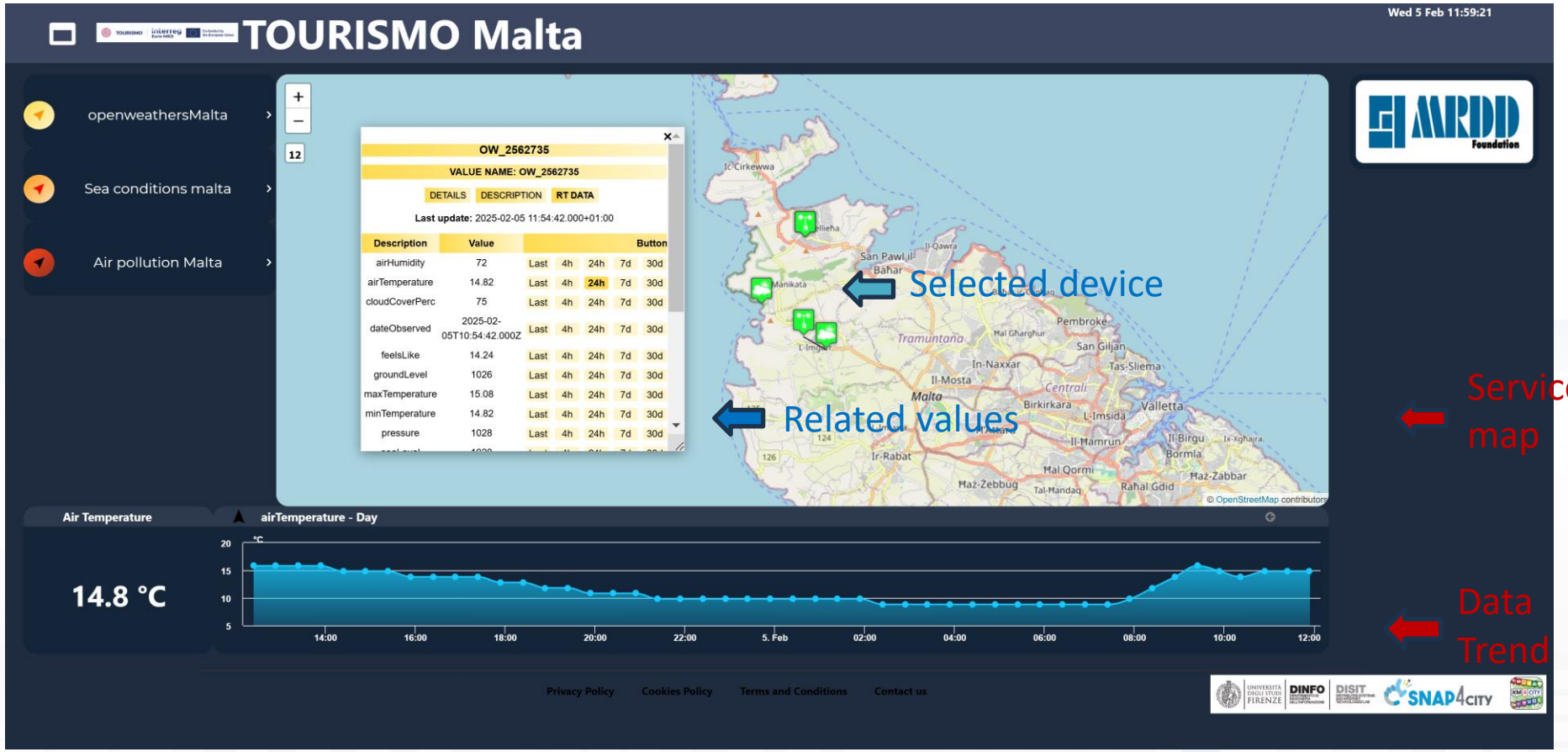
Data Source	Data Source Main Address	Periodicity for Data ingestion	Variables
Sea Conditions	https://open-meteo.com/	60 mins	• oceanCurrentDirection • oceanCurrentVelocity • swellWaveDirection • swellWaveHeight • swellWavePeakPeriod • swellWavePeriod • waveDirection • waveHeight • wavePeriod • windWaveDirection • windWaveHeight • windWavePeakPeriod • windWavePeriod

Data Ingestion Example

Data Source	Data Source Main Address	Periodicity for Data ingestion	Variables
Open Pollution	https://openweathermap.org/api/air-pollution	30 mins	• CO • NO • NO2 • NH3 • O3 • SO2 • PM2.5 • PM10

Dashboard:
https://www.snap4city.org/dashboardSmartCity/view/newTheme.php?iddasboard=NDMzNQ==

Data Visualization in a Dashboard



Devices selector

Selected device

Related values

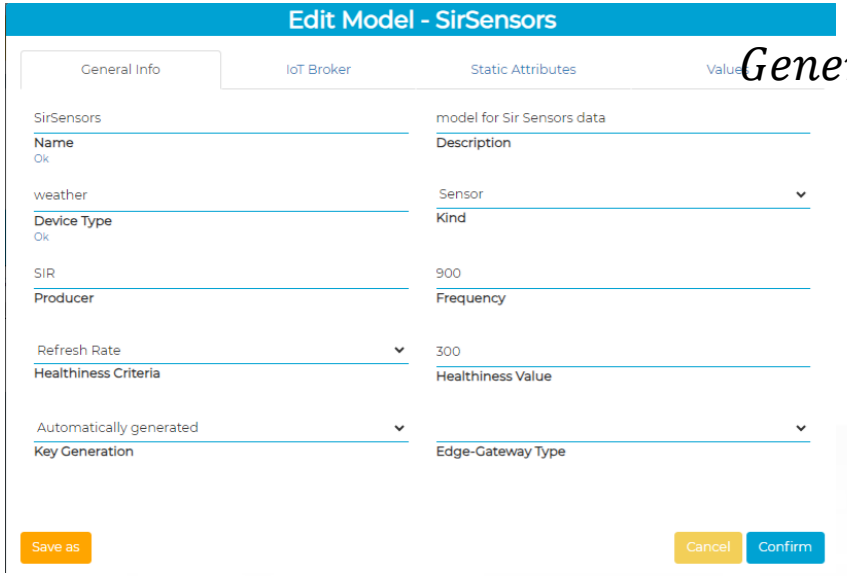
Partner link

Service map

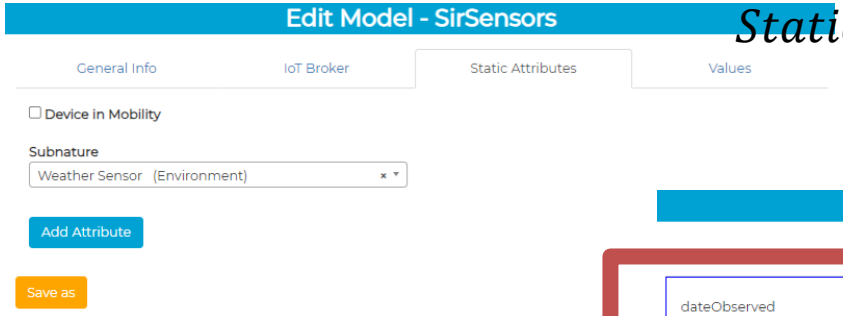
Data Trend

HOW TO Create an IOT Device Model

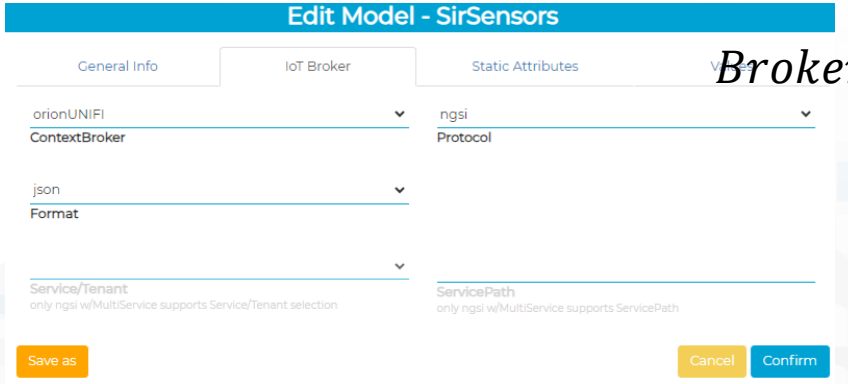
General Info



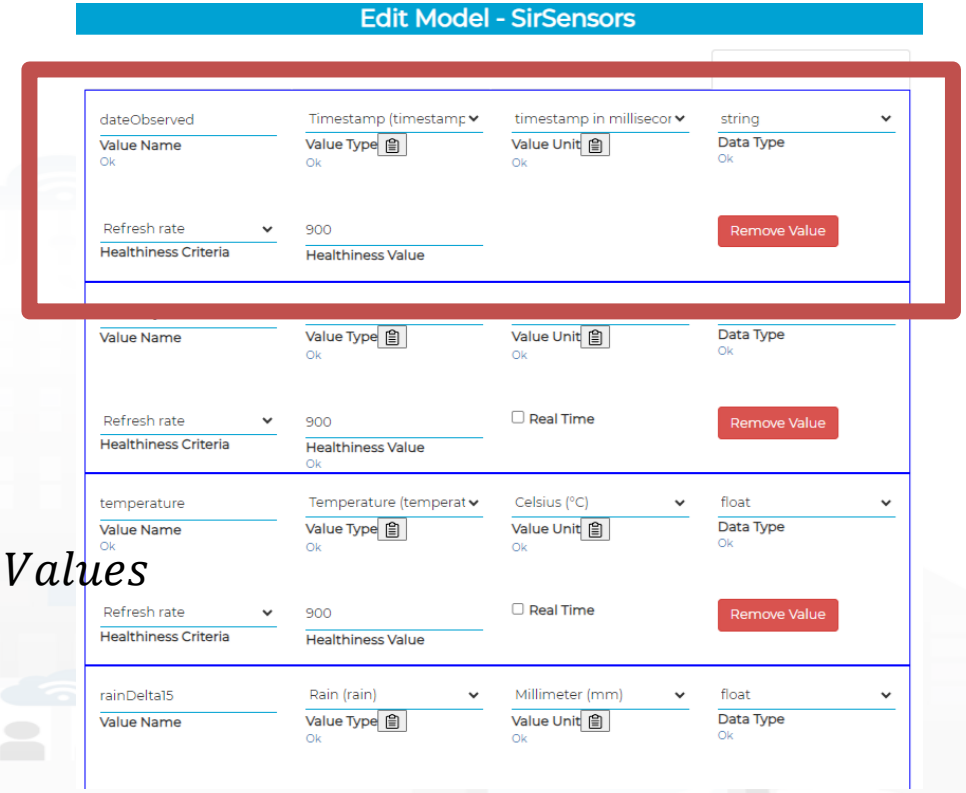
Static Attributes



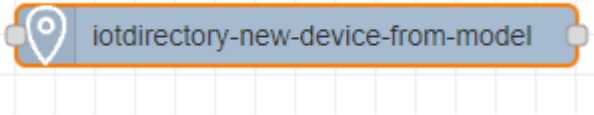
Broker

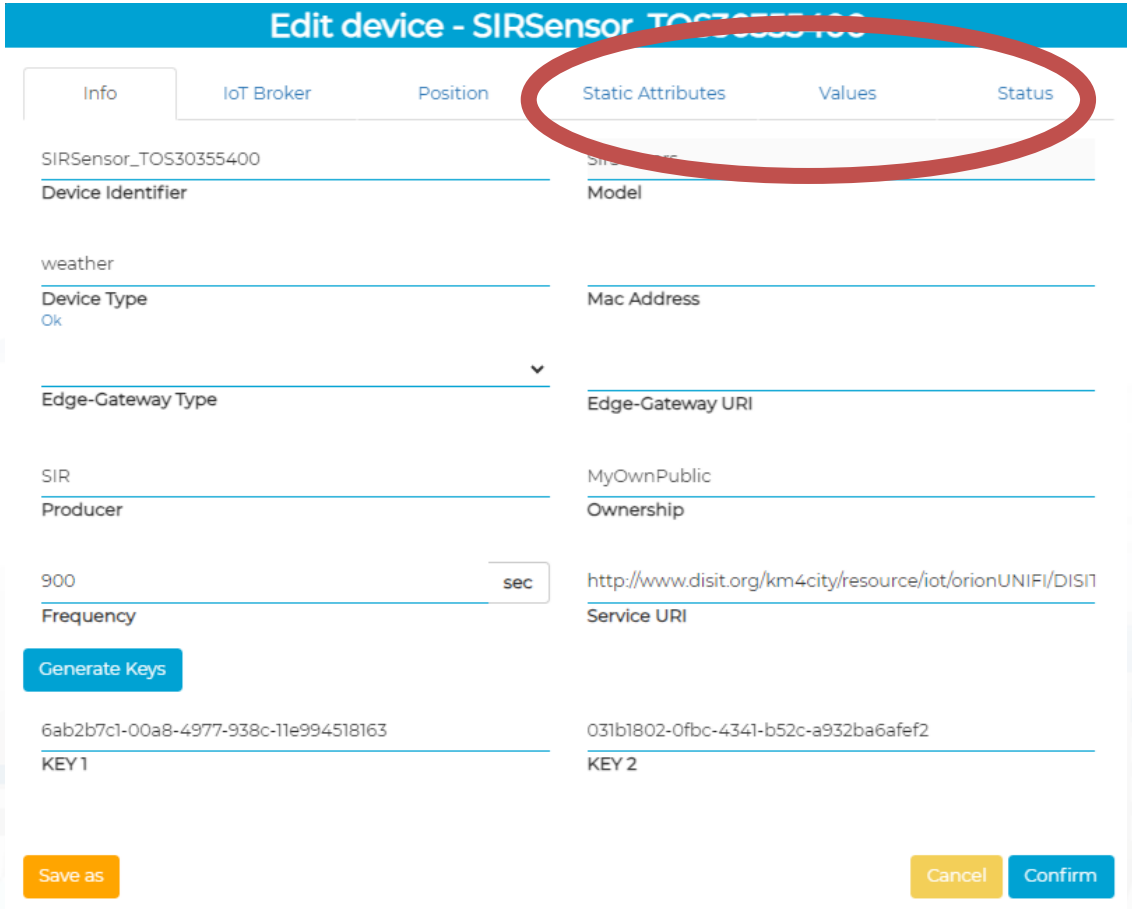


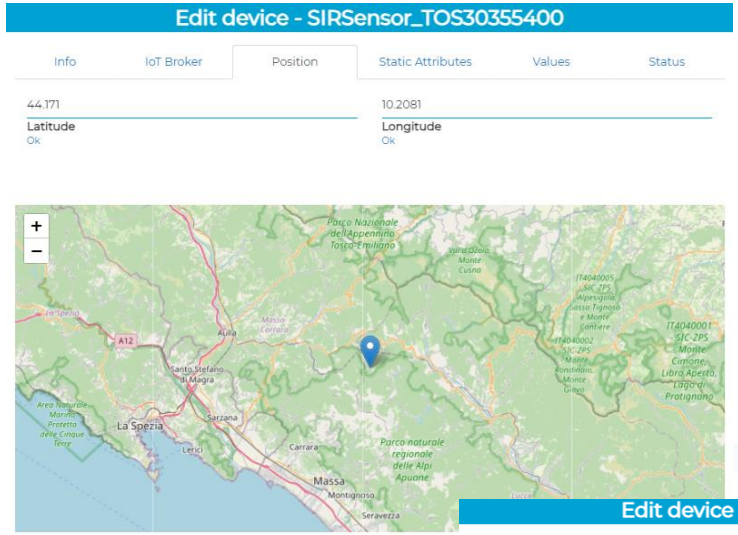
Values

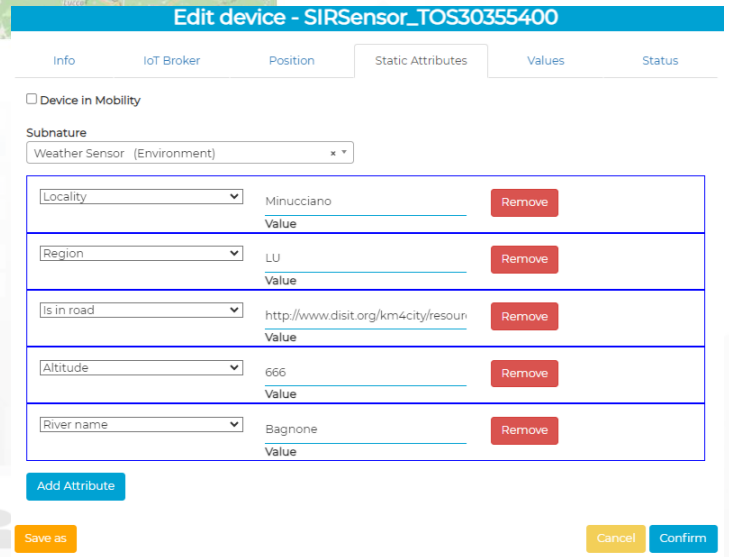


HOW TO Create an IOT Device instance

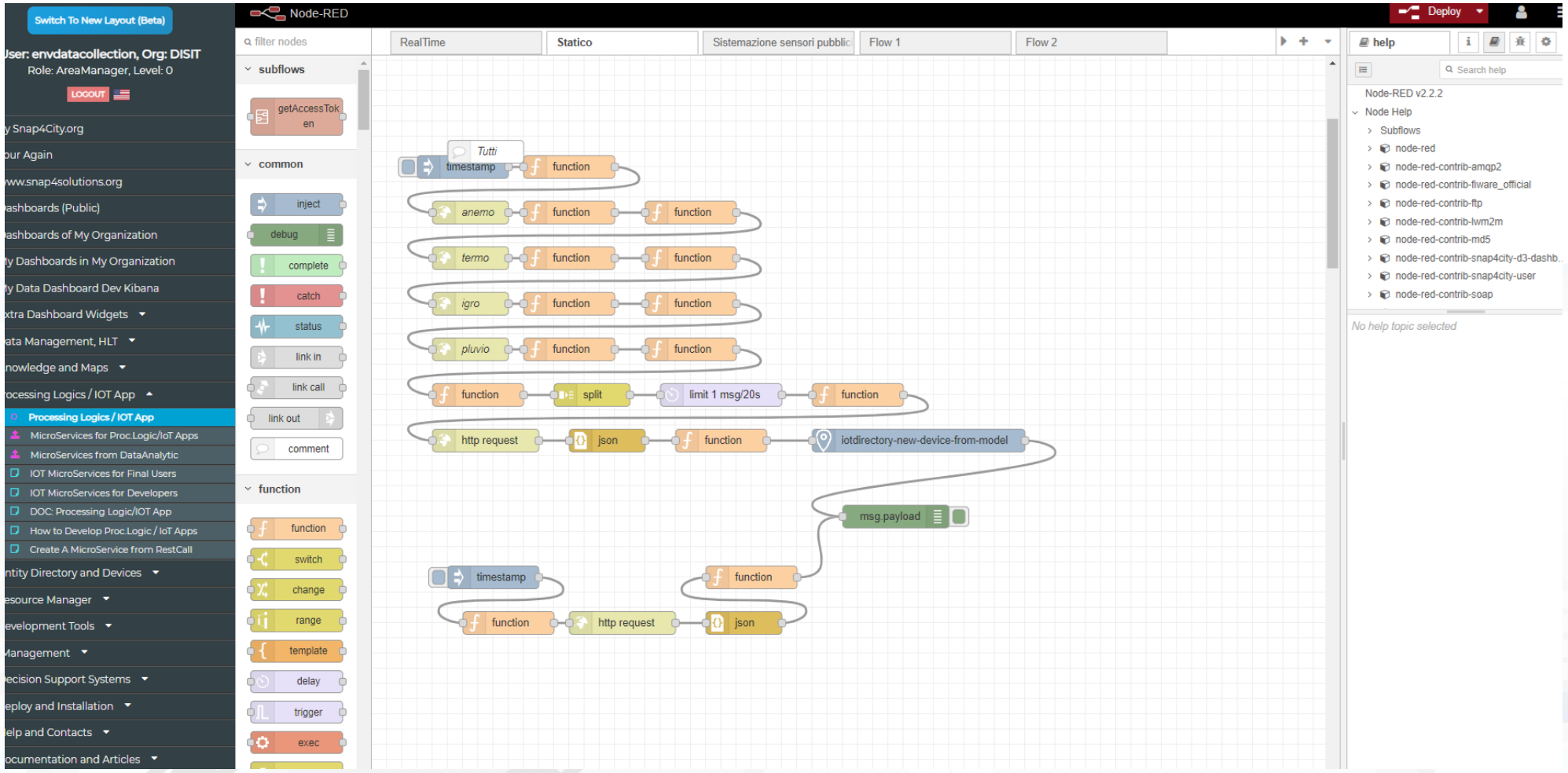




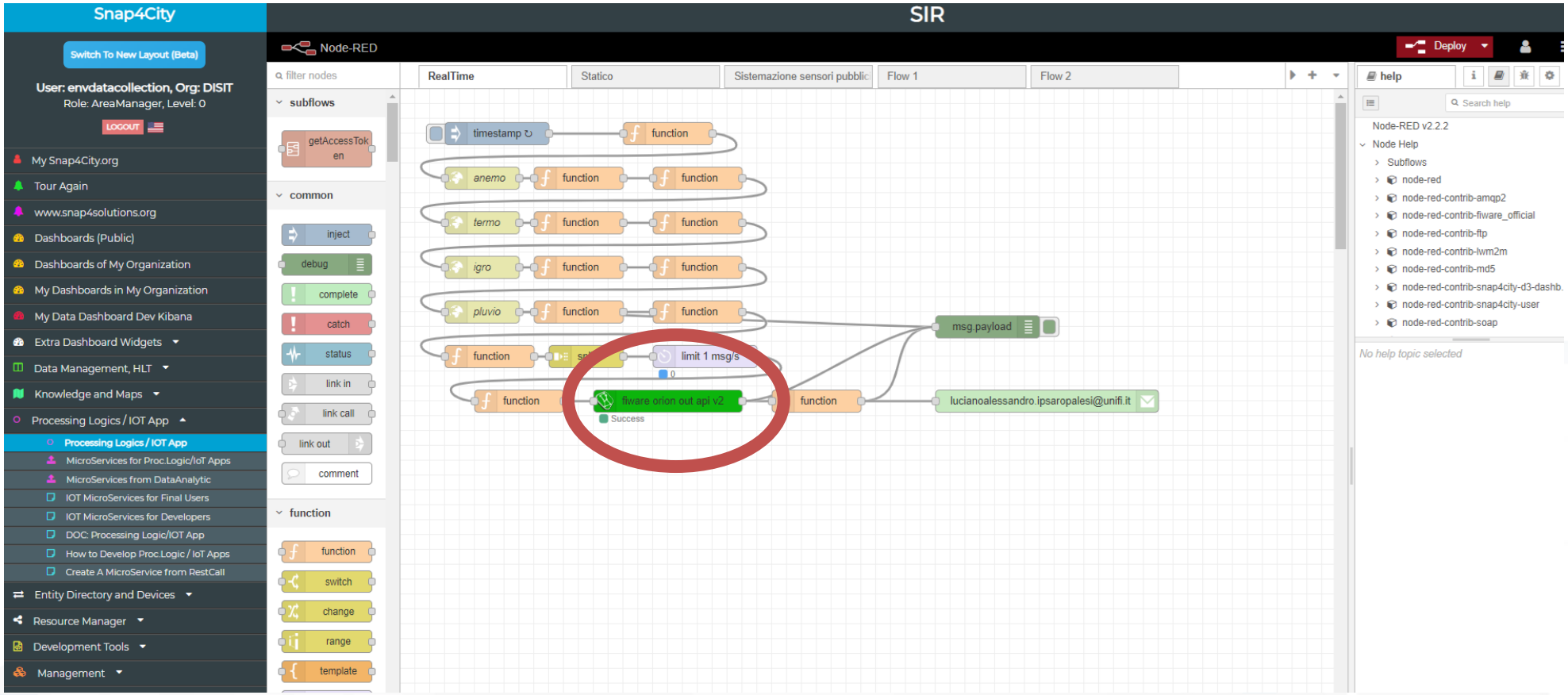




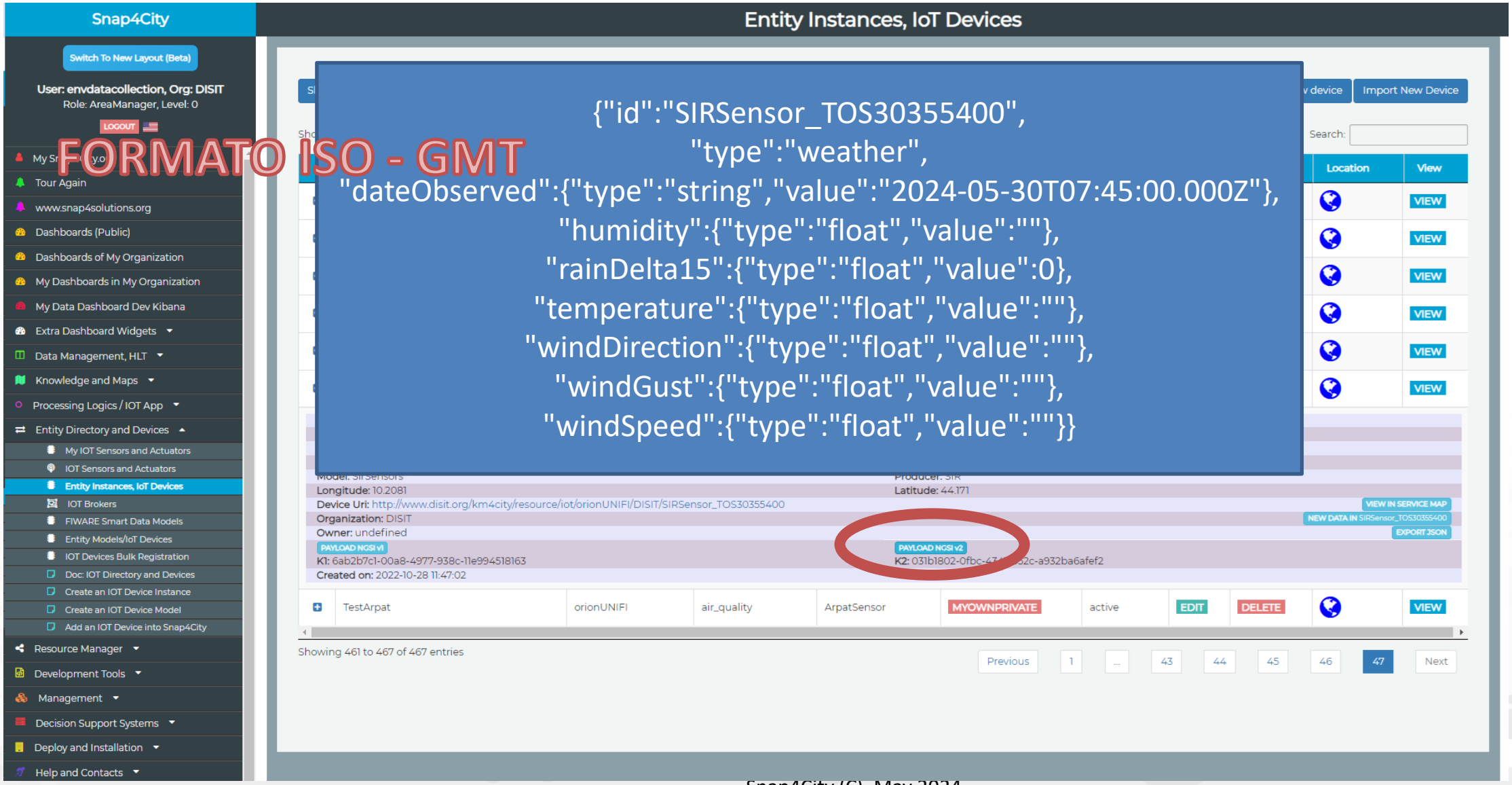
HOW TO Create an IOT Device instance



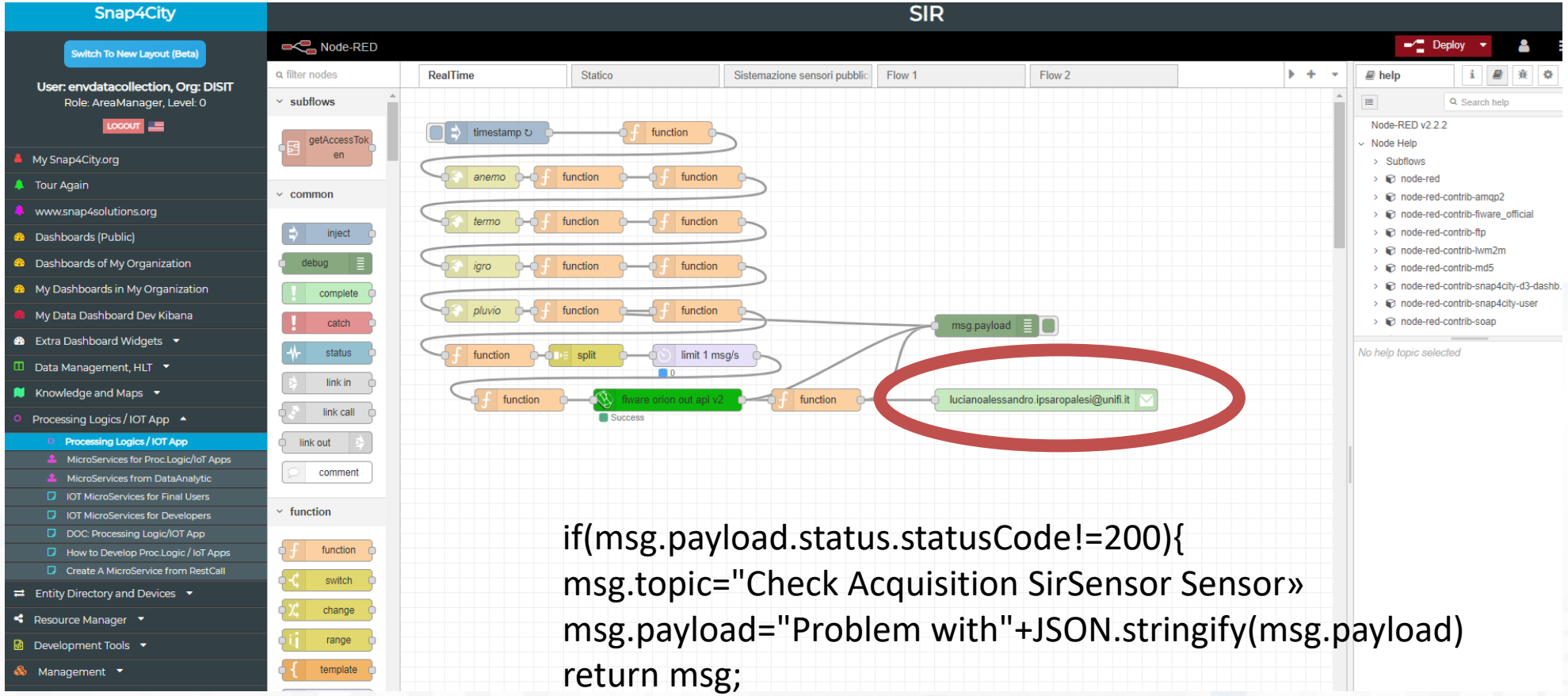
HOW TO Create an IOT Application for Data Ingestion



HOW TO Create an IOT Application for Data Ingestion

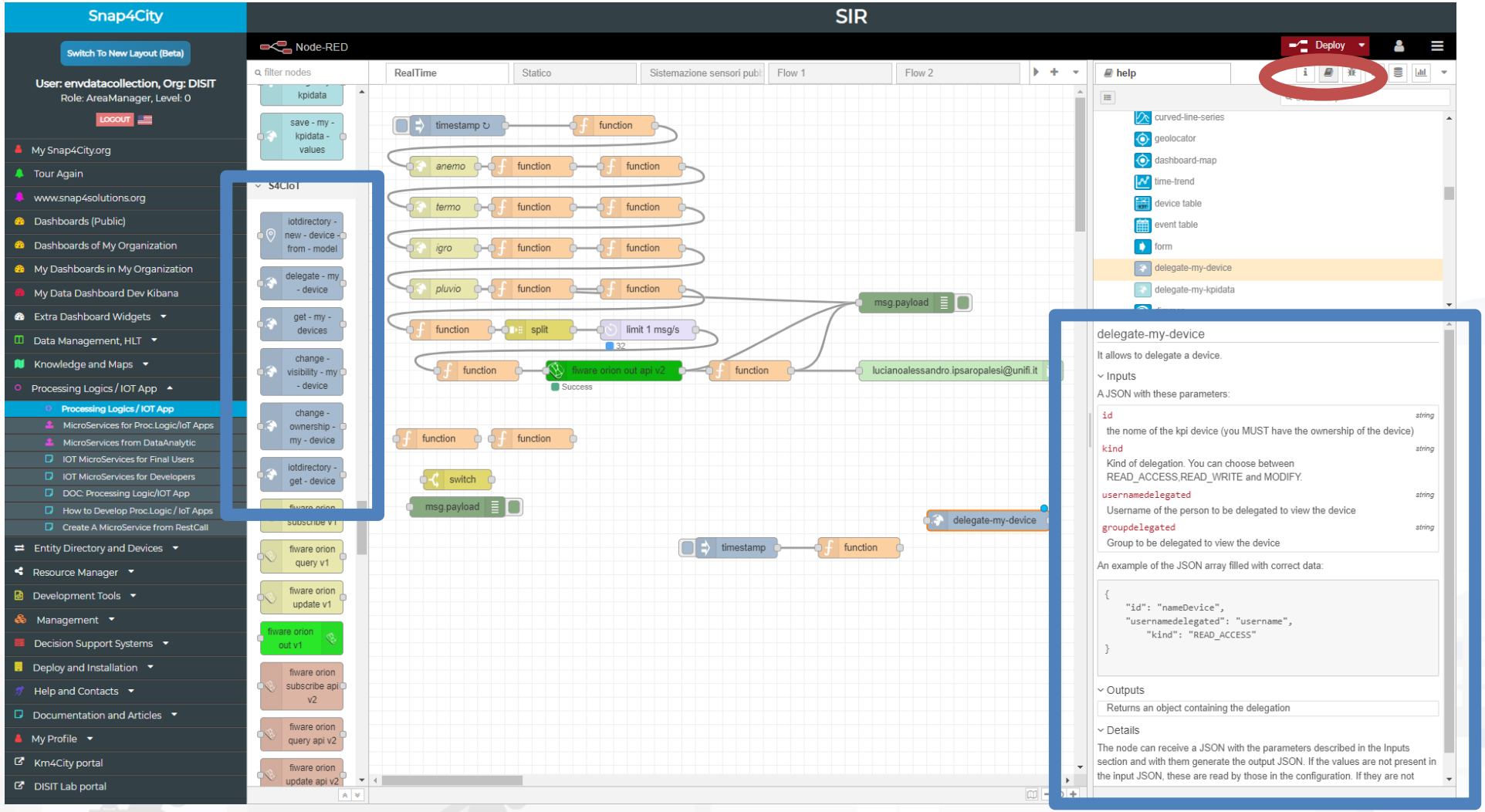


HOW TO Managing Notifications on IOT Application




```
if(msg.payload.status.statusCode!=200){
msg.topic="Check Acquisition SirSensor Sensor»
msg.payload="Problem with"+JSON.stringify(msg.payload)
return msg;
}
return null;
```

IotApp


Snap4City
SIR
Node-RED
Deploy
Switch To New Layout (Beta)
User: envdatacollection, Org: DISIT
Role: AreaManager, Level: 0
LOGOUT
My Snap4City.org
Tour Again
www.snap4solutions.org
Dashboards (Public)
Dashboards of My Organization
My Dashboards in My Organization
My Data Dashboard Dev Kibana
Extra Dashboard Widgets
Data Management, HLT
Knowledge and Maps
Processing Logics / IOT App
Processing Logics / IOT App
MicroServices for Proc.Logic/IoT Apps
MicroServices from DataAnalytic
IOT MicroServices for Final Users
IOT MicroServices for Developers
DOC: Processing Logic/IOT App
How to Develop Proc.Logic / IoT Apps
Create A MicroService from RestCall
Entity Directory and Devices
Resource Manager
Development Tools
Management
Decision Support Systems
Deploy and Installation
Help and Contacts
Documentation and Articles
My Profile
Km4City portal
DISIT Lab portal
RealTime
Statico
Sistemazione sensori publ
Flow 1
Flow 2
help
delegate-my-device
It allows to delegate a device.
Inputs
A JSON with these parameters:
id
the nome of the kpi device (you MUST have the ownership of the device)
kind
Kind of delegation. You can choose between READ_ACCESS,READ_WRITE and MODIFY.
usernamedelegated
Username of the person to be delegated to view the device
groupdelegated
Group to be delegated to view the device
An example of the JSON array filled with correct data:
Outputs
Returns an object containing the delegation
Details
The node can receive a JSON with the parameters described in the Inputs section and with them generate the output JSON. If the values are not present in the input JSON, these are read by those in the configuration. If they are not

CSBLTutorial

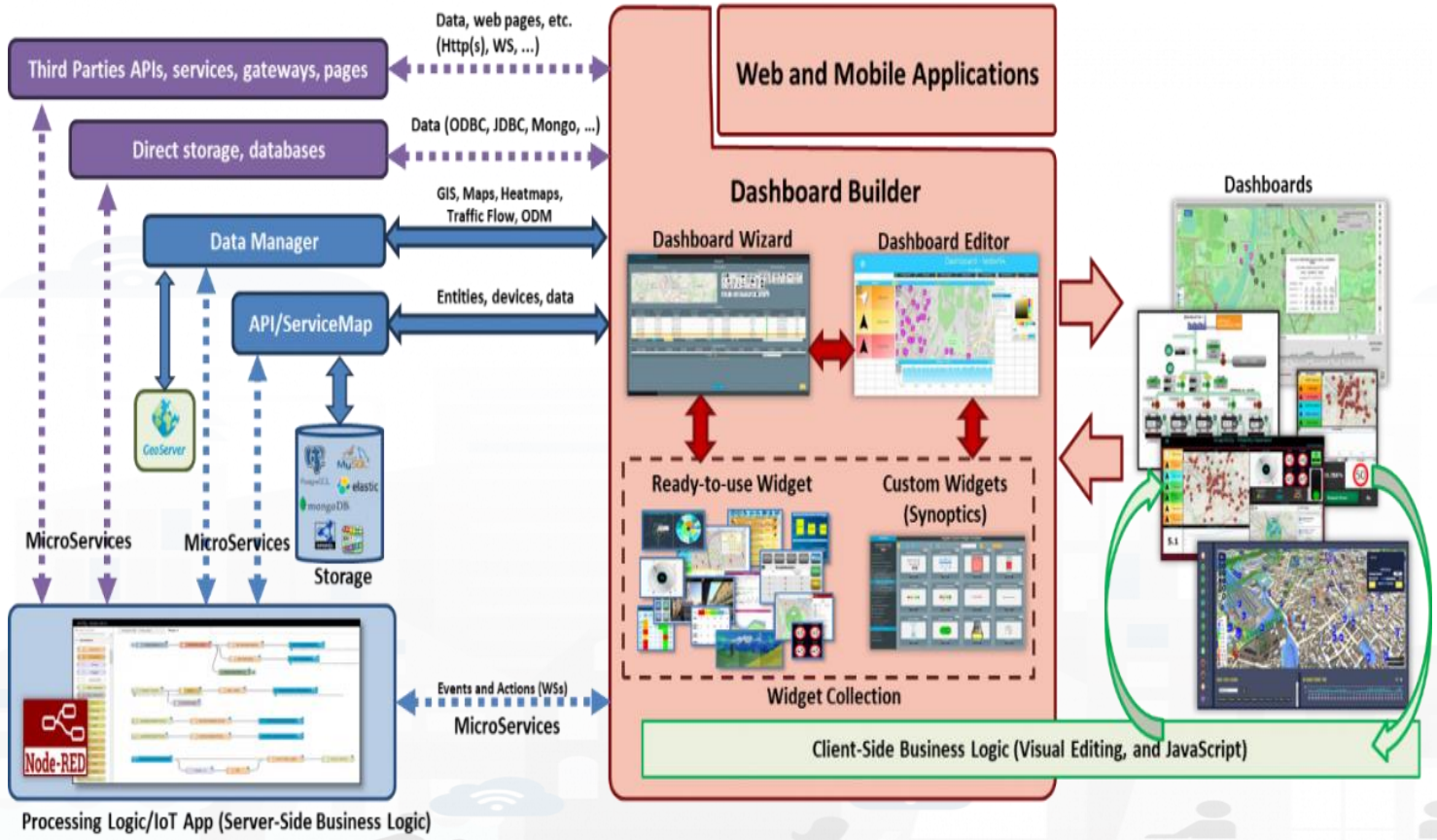
UNIVERSITÀ DEGLI STUDI FIRENZE | DINFO DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE | DISIT DISTRIBUTED SYSTEMS AND INTERNET TECHNOLOGIES LAB

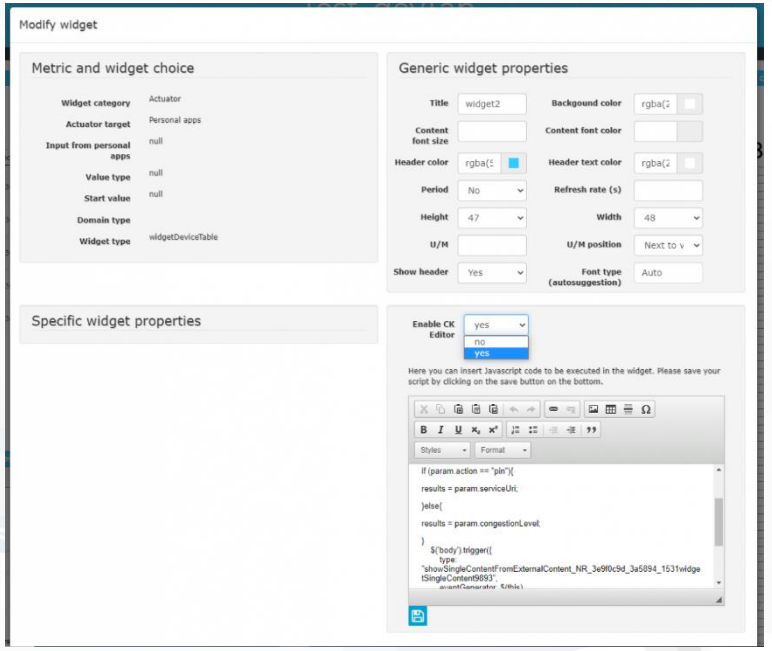
CSBL

SNAP4CITY KM4CITY

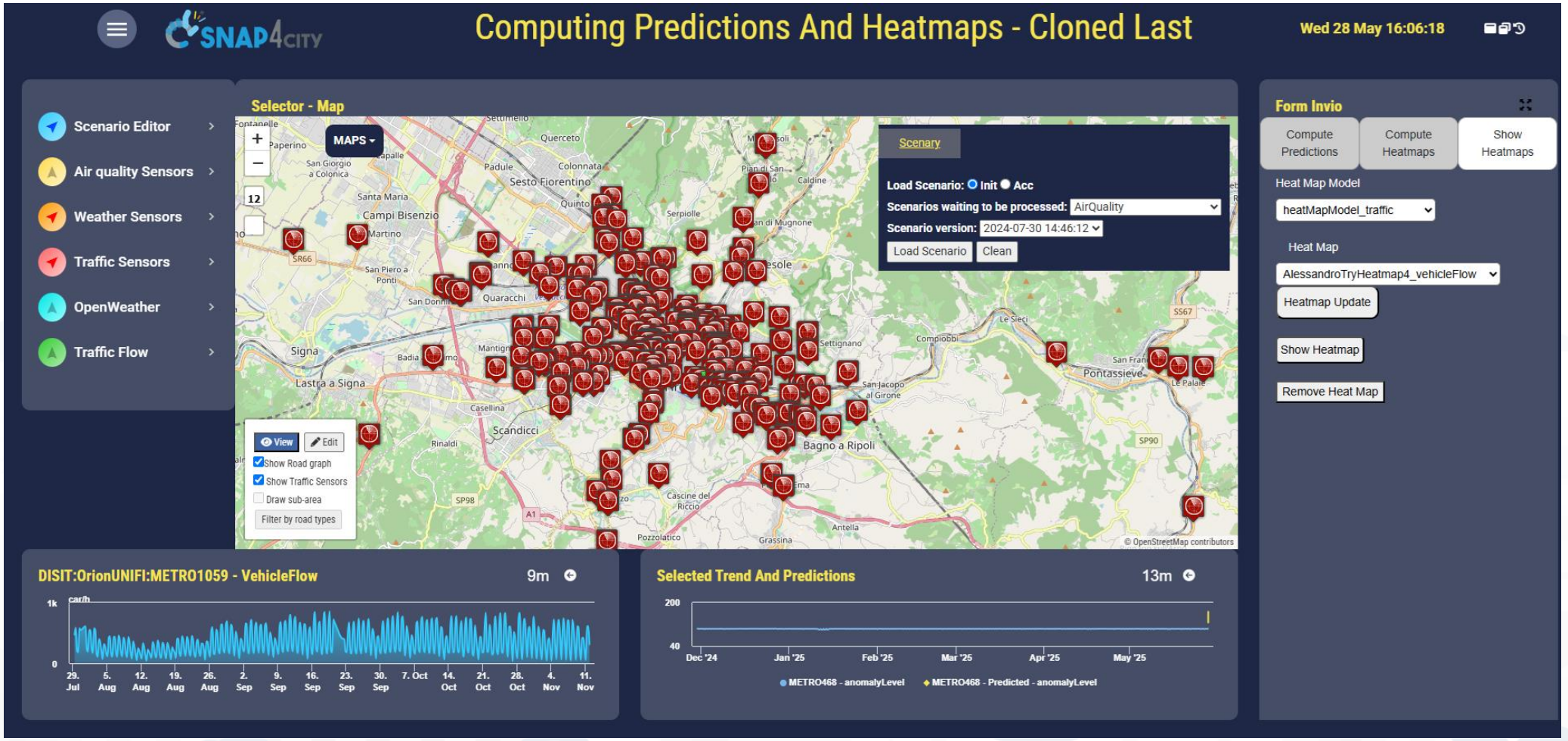
Client-Side Business Logic on Snap4city.org platform

Manual: https://www.snap4city.org/download/video/Snap4Tech-Development-Life-Cycle.pdf





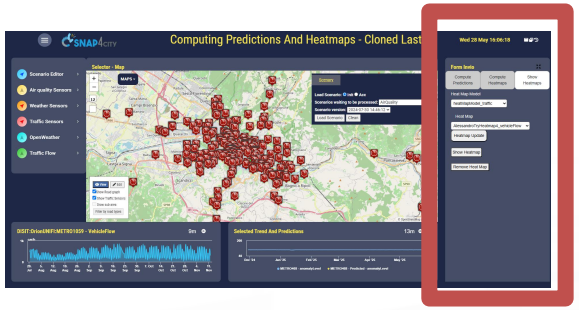
Computing Predictions and Heatmaps



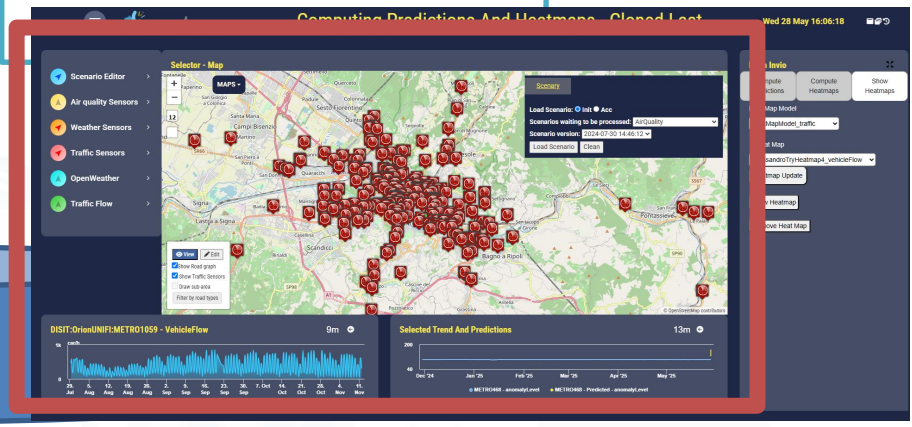
CSBL

API PYTHON

CSBL



Storage



Javascript Request

```javascript
$.ajax({
  url: 'https://api.example.com/data',
  type: 'POST',
  contentType: 'application/json',
  data: JSON.stringify({
    name: 'John',
    age: 30
  }),
  headers: {
    'Authorization': 'Bearer YOUR_ACCESS_TOKEN'
  },
  success: function(response) {
    console.log('Success:', response);
  },
  error: function(xhr, status, error) {
    console.error('Error:', error);
  }
});
```

url: Target API endpoint.

type: 'POST': Method of the request.

contentType: Tells the server the format of the data (application/json).

data: Payload sent to the server (must be stringified).

headers: Include Authorization if needed.

success: Callback for a successful response.

error: Callback for failures.

API Flask

Introduction to Flask

What is Flask?

- Flask is a micro web framework for Python.
- Flexible and ideal for creating APIs and web applications.
- Based on Werkzeug (server) and Jinja2 (templating engine).

Main Features:

- Minimalist: only what's needed to start.
- Extendable with plugins and libraries.
- Great for small or medium projects.

Main Components:

- Routing: associates URLs with functions
- Request/Response: handles data from forms, JSON, URLs
- Modular structure for large projects
- Many useful extensions.

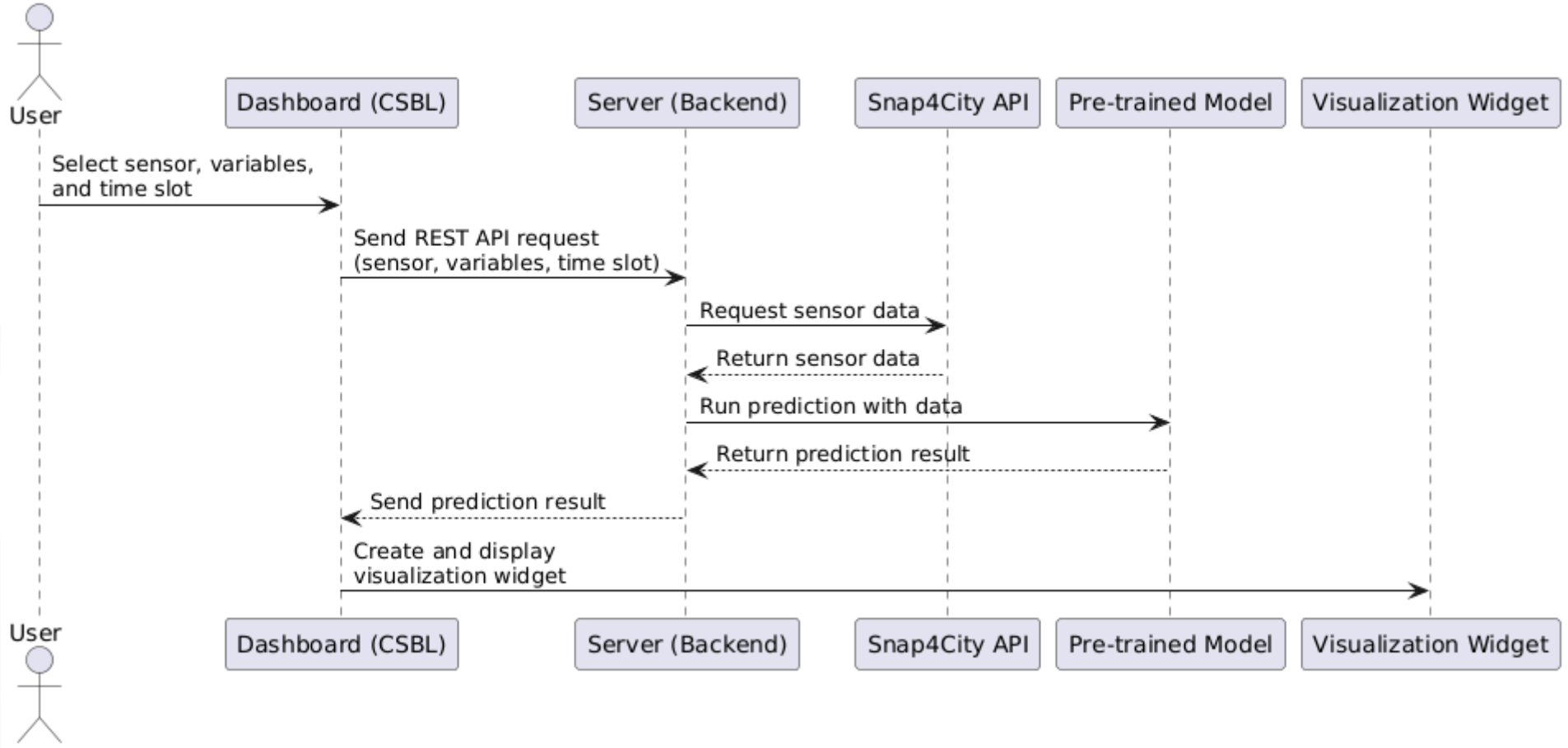
```python
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Ciao, Flask multi-thread!"

if __name__ == "__main__":
    app.run(debug=True, threaded=True)
```

threaded=True

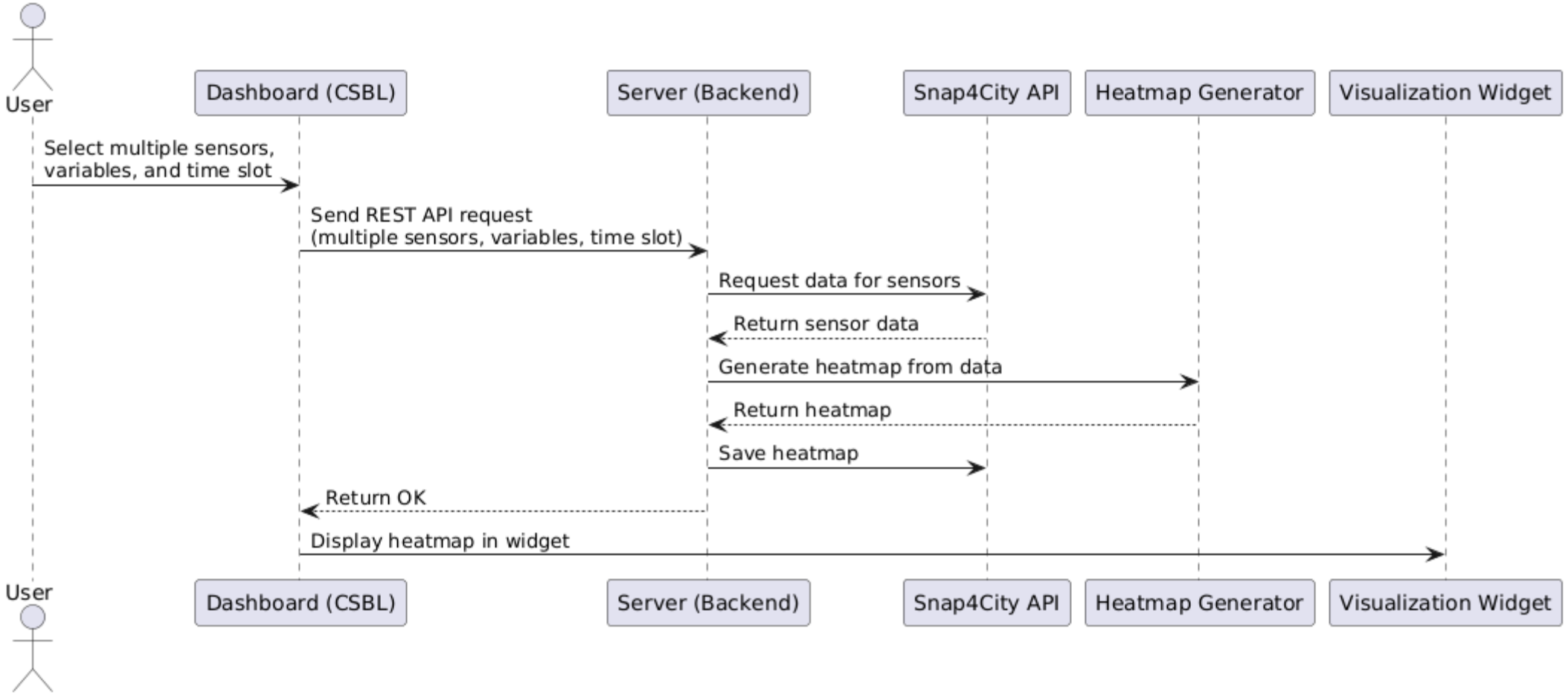
Enable multi-threading:
Each request is handled in a separate thread.

Alternative: processes=n for multiprocessing management, but less common for Flask in development.

Prediction


User
Dashboard (CSBL)
Server (Backend)
Snap4City API
Pre-trained Model
Visualization Widget
Select sensor, variables, and time slot
Send REST API request (sensor, variables, time slot)
Request sensor data
Return sensor data
Run prediction with data
Return prediction result
Send prediction result
Create and display visualization widget
User
Dashboard (CSBL)
Server (Backend)
Snap4City API
Pre-trained Model
Visualization Widget

Heatmap


User
Dashboard (CSBL)
Server (Backend)
Snap4City API
Heatmap Generator
Visualization Widget
Select multiple sensors, variables, and time slot
Send REST API request (multiple sensors, variables, time slot)
Request data for sensors
Return sensor data
Generate heatmap from data
Return heatmap
Save heatmap
Return OK
Display heatmap in widget
User
Dashboard (CSBL)
Server (Backend)
Snap4City API
Heatmap Generator
Visualization Widget

- API Heatmap:
  - https://github.com/disit/snap4city/blob/master/Computing/predictions/heatmap_service.py
- Funzione Heatmap:
  - https://github.com/disit/snap4city/blob/master/Computing/predictions/heatmap.py
- API prediction:
  - https://github.com/disit/snap4city/blob/master/Computing/predictions/predictions_service.py