

# *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	<a href="https://openweathermap.org/api">https://openweathermap.org/api</a>	30 mins	<ul style="list-style-type: none"><li>• airHumidity</li><li>• airTemperature</li><li>• cloudCoverPerc</li><li>• feelsLike</li><li>• groundLevel</li><li>• maxTemperature</li><li>• minTemperature</li><li>• pressure</li><li>• seaLevel</li><li>• sunrise</li><li>• sunset</li><li>• visibility</li><li>• windDirection</li><li>• windSpeed</li></ul>

# Data Ingestion Example

Data Source	Data Source Main Address	Periodicity for Data ingestion	Variables
Sea Conditions	<a href="https://open-meteo.com/">https://open-meteo.com/</a>	60 mins	<ul style="list-style-type: none"><li>• oceanCurrentDirection</li><li>• oceanCurrentVelocity</li><li>• swellWaveDirection</li><li>• swellWaveHeight</li><li>• swellWavePeakPeriod</li><li>• swellWavePeriod</li><li>• waveDirection</li><li>• waveHeight</li><li>• wavePeriod</li><li>• windWaveDirection</li><li>• windWaveHeight</li><li>• windWavePeakPeriod</li><li>• windWavePeriod</li></ul>

# Data Ingestion Example

Data Source	Data Source Main Address	Periodicity for Data ingestion	Variables
Open Pollution	<a href="https://openweathermap.org/api/air-pollution">https://openweathermap.org/api/air-pollution</a>	30 mins	<ul style="list-style-type: none"><li>• CO</li><li>• NO</li><li>• NO2</li><li>• NH3</li><li>• O3</li><li>• SO2</li><li>• PM2.5</li><li>• PM10</li></ul>

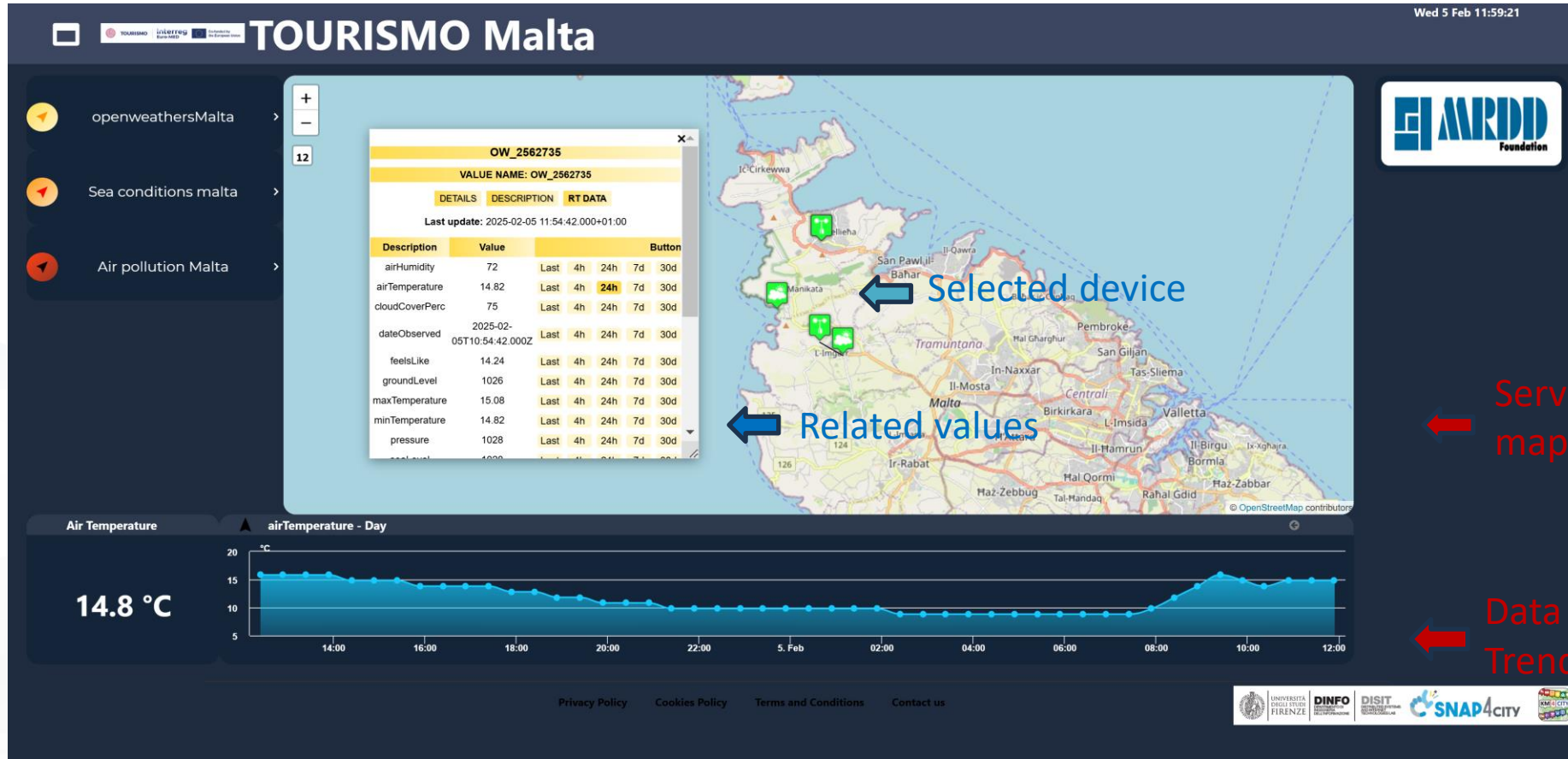
Dashboard:

<https://www.snap4city.org/dashboardSmartCity/view/newTheme.php?iddashboard=NDMzNQ==>



# Data Visualization in a Dashboard

Devices  
selector



Partner  
link



Service  
map



Data  
Trend



# HOW TO Create an IOT Device Model

## Edit Model - SirSensors

General Info

IoT Broker

Static Attributes

Values

SirSensors	model for Sir Sensors data
Name	Description
weather	Sensor
Device Type	Kind
SIR	900
Producer	Frequency
Refresh Rate	300
Healthiness Criteria	Healthiness Value
Automatically generated	Edge-Gateway Type
Key Generation	

Save as

Cancel

Confirm

*General Info*

## Edit Model - SirSensors

General Info

IoT Broker

Static Attributes

Values

☐ Device in Mobility

Subnature

Weather Sensor (Environment)

Add Attribute

Save as

*Static Attributes*

## Edit Model - SirSensors

dateObserved	Timestamp (timestamp)	timestamp in millisecond	string
Value Name	Value Type	Value Unit	Data Type
Refresh rate	900		Remove Value
Healthiness Criteria	Healthiness Value		

Value Name	Value Type	Value Unit	Data Type
Refresh rate	900	<input type="checkbox"/> Real Time	Remove Value
Healthiness Criteria	Healthiness Value		

temperature	Temperature (temperat	Celsius (°C)	float
Value Name	Value Type	Value Unit	Data Type
Refresh rate	900	<input type="checkbox"/> Real Time	Remove Value
Healthiness Criteria	Healthiness Value		

rainDelta15	Rain (rain)	Millimeter (mm)	float
Value Name	Value Type	Value Unit	Data Type

*Values*

## Edit Model - SirSensors

General Info

IoT Broker

Static Attributes

*Broker*

orionUNIFI	ngsi
ContextBroker	Protocol
json	
Format	
Service/Tenant	ServicePath
only ngsi w/MultiService supports Service/Tenant selection	only ngsi w/MultiService supports ServicePath

Save as

Cancel

Confirm

# HOW TO Create an IOT Device instance

iotdirectory-new-device-from-model

### Edit device - SIRSensor\_TOS30355400

Info	IoT Broker	Position	Static Attributes	Values	Status
SIRSensor_TOS30355400					
Device Identifier			Model		
weather			Mac Address		
Device Type Ok					
Edge-Gateway Type			Edge-Gateway URI		
SIR			MyOwnPublic		
Producer			Ownership		
900			http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT		
Frequency			Service URI		
6ab2b7c1-00a8-4977-938c-11e994518163			031b1802-0fbc-4341-b52c-a932ba6afef2		
KEY1			KEY2		

Generate Keys

Save as Cancel Confirm

### Edit device - SIRSensor\_TOS30355400

Info	IoT Broker	Position	Static Attributes	Values	Status
44.171		10.2081			
Latitude		Longitude			
Ok		Ok			

Save as

### Edit device - SIRSensor\_TOS30355400

Info	IoT Broker	Position	Static Attributes	Values	Status
<input type="checkbox"/> Device in Mobility					
Subnature Weather Sensor (Environment)					
Locality	Minucciano	Remove			
Region	LU	Remove			
Is in road	http://www.disit.org/km4city/resoun	Remove			
Altitude	666	Remove			
River name	Bagnone	Remove			

Add Attribute

Save as Cancel Confirm



# HOW TO Create an IOT Device instance

The screenshot displays the Node-RED web interface. On the left, a sidebar shows the user profile 'User: envdatacollection, Org: DISIT' and a menu with categories like 'Dashboards (Public)', 'Dashboards of My Organization', and 'Processing Logics / IOT App'. The main workspace shows a flow titled 'Sistemazione sensori pubblici' with two subflows, 'Flow 1' and 'Flow 2'. The flow starts with a 'Tutti timestamp' node, followed by a series of 'function' nodes. These nodes are connected to a 'split' node, which then branches into two parallel paths. The top path includes an 'http request' node, a 'json' node, and a 'function' node. The bottom path includes a 'timestamp' node, a 'function' node, an 'http request' node, and a 'json' node. Both paths converge at a 'msg payload' node. The right sidebar shows the 'Node Help' section for Node-RED v2.2.2, with a list of subflows and a search bar.

# HOW TO Create an IOT Application for Data Ingestion

The screenshot displays the Snap4City Node-RED interface. On the left, a sidebar shows the user profile (envdatacollection, Org: DISIT) and a navigation menu with options like 'Processing Logics / IOT App'. The main workspace, titled 'SIR', shows a flow diagram with several 'function' nodes. A red circle highlights a 'limit 1 msg/s' node, which is connected to a 'msg.payload' node and a 'lucianolessandro.ipsaropalesi@unifi.it' node. The right sidebar contains a 'help' section for Node-RED v2.2.2.

# HOW TO Create an IOT Application for Data Ingestion

**Snap4City**

Switch To New Layout (Beta)

User: envdatacollection, Org: DISIT  
Role: AreaManager, Level: 0

Logout

- My Snap4City
- 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
- Entity Directory and Devices
  - My IOT Sensors and Actuators
  - IOT Sensors and Actuators
  - Entity Instances, IoT Devices**
  - IOT Brokers
  - FIWARE Smart Data Models
  - Entity Models/IoT Devices
  - IOT Devices Bulk Registration
  - Doc: IOT Directory and Devices
  - Create an IOT Device Instance
  - Create an IOT Device Model
  - Add an IOT Device into Snap4City
- Resource Manager
- Development Tools
- Management
- Decision Support Systems
- Deploy and Installation
- Help and Contacts

## Entity Instances, IoT Devices

Search:

Import New Device

Location	View
	<a href="#">VIEW</a>
	<a href="#">VIEW</a>
	<a href="#">VIEW</a>
	<a href="#">VIEW</a>
	<a href="#">VIEW</a>
	<a href="#">VIEW</a>

```
{
  "id": "SIRSensor_TOS30355400",
  "type": "weather",
  "dateObserved": {
    "type": "string",
    "value": "2024-05-30T07:45:00.000Z"
  },
  "humidity": {
    "type": "float",
    "value": ""
  },
  "rainDelta15": {
    "type": "float",
    "value": 0
  },
  "temperature": {
    "type": "float",
    "value": ""
  },
  "windDirection": {
    "type": "float",
    "value": ""
  },
  "windGust": {
    "type": "float",
    "value": ""
  },
  "windSpeed": {
    "type": "float",
    "value": ""
  }
}
```

Model: SIRSensor

Longitude: 10.2081

Device Uri: [http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/SIRSensor\\_TOS30355400](http://www.disit.org/km4city/resource/iot/orionUNIFI/DISIT/SIRSensor_TOS30355400)

Organization: DISIT

Owner: undefined

[PAYLOAD NGSI v1](#)

K1: 6ab2b7c1-00a8-4977-938c-11e994518163

Created on: 2022-10-28 11:47:02

Producer: SIR

Latitude: 44.171

[VIEW IN SERVICE MAP](#)

[NEW DATA IN SIRSensor\\_TOS30355400](#)

[EXPORT JSON](#)

[PAYLOAD NGSI v2](#)

K2: 031b1802-0fbc-4741-b32c-a932ba6afef2

+	TestArpat	orionUNIFI	air_quality	ArpatSensor	MYOWNPRIVATE	active	<a href="#">EDIT</a>	<a href="#">DELETE</a>		<a href="#">VIEW</a>

Showing 461 to 467 of 467 entries

Previous

1 ... 43 44 45 46 **47** Next

# 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

The screenshot displays the Snap4City IoTApp interface. On the left is a sidebar with navigation options. The main workspace shows a Node-RED flow for 'S4CIoT'. A blue box highlights the 'delegate-my-device' node in the left sidebar and its details in the right sidebar. A red circle highlights the 'Deploy' button in the top right corner of the Node-RED interface.

**Left Sidebar (Navigation):**

- 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

**Node-RED Flow (S4CIoT):**

- Input: kpdata
- Function: save - my - kpdata - values
- Function: timestamp
- Function: anemo
- Function: termo
- Function: igro
- Function: pluvio
- Function: fware orion out api v2
- Function: split
- Function: limit 1 msg/s
- Function: msg payload
- Function: lucianolessandro.ipsaropalesi@unifi.it
- Function: delegate-my-device

**delegate-my-device Node Details:**

It allows to delegate a device.

**Inputs**

A JSON with these parameters:

- id** (string): the nome of the kpi device (you MUST have the ownership of the device)
- kind** (string): Kind of delegation. You can choose between READ\_ACCESS, READ\_WRITE and MODIFY.
- usernameDelegated** (string): Username of the person to be delegated to view the device
- groupDelegated** (string): Group to be delegated to view the device

**An example of the JSON array filled with correct data:**

```
{
  "id": "nameDevice",
  "usernameDelegated": "username",
  "kind": "READ_ACCESS"
}
```

**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



# CSBL Tutorial







# Computing Predictions and Heatmaps

Computing Predictions And Heatmaps - Cloned Last

Wed 28 May 16:06:18

Scenario Editor

Air quality Sensors

Weather Sensors

Traffic Sensors

OpenWeather

Traffic Flow

Selector - Map

MAPS

View

Edit

☒ Show Road graph
 ☒ Show Traffic Sensors
 ☐ Draw sub-area
 

Filter by road types

Scenario

Load Scenario: ☐ Init ☒ Acc

Scenarios waiting to be processed: AirQuality

Scenario version: 2024-07-30 14:46:12

Load Scenario Clean

Form Invio

Compute Predictions

Compute Heatmaps

Show Heatmaps

Heat Map Model

heatMapModel\_traffic

Heat Map

AlessandroTryHeatmap4\_vehicleFlow

Heatmap Update

Show Heatmap

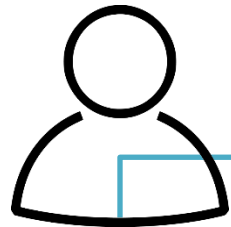
Remove Heat Map

DISIT:OrionUNIFI:METRO1059 - VehicleFlow

9m

Selected Trend And Predictions

13m



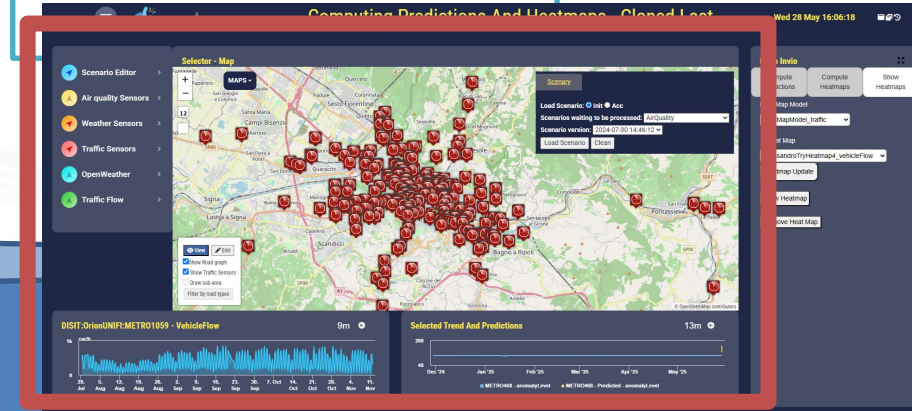
CSBL

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!"

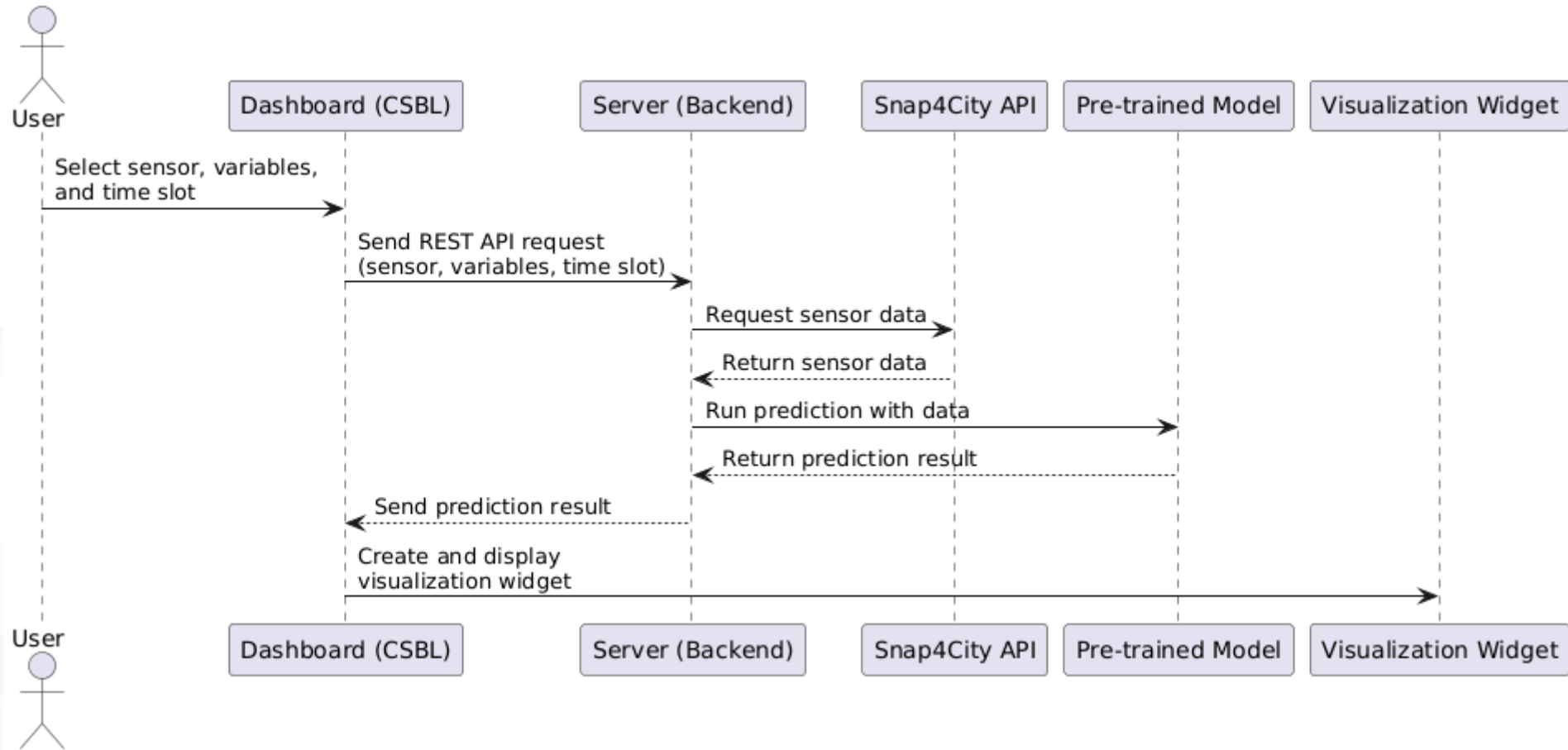
if __name__ == "__main__":
    app.run(debug=True, threaded=True)
```

**threaded=True**

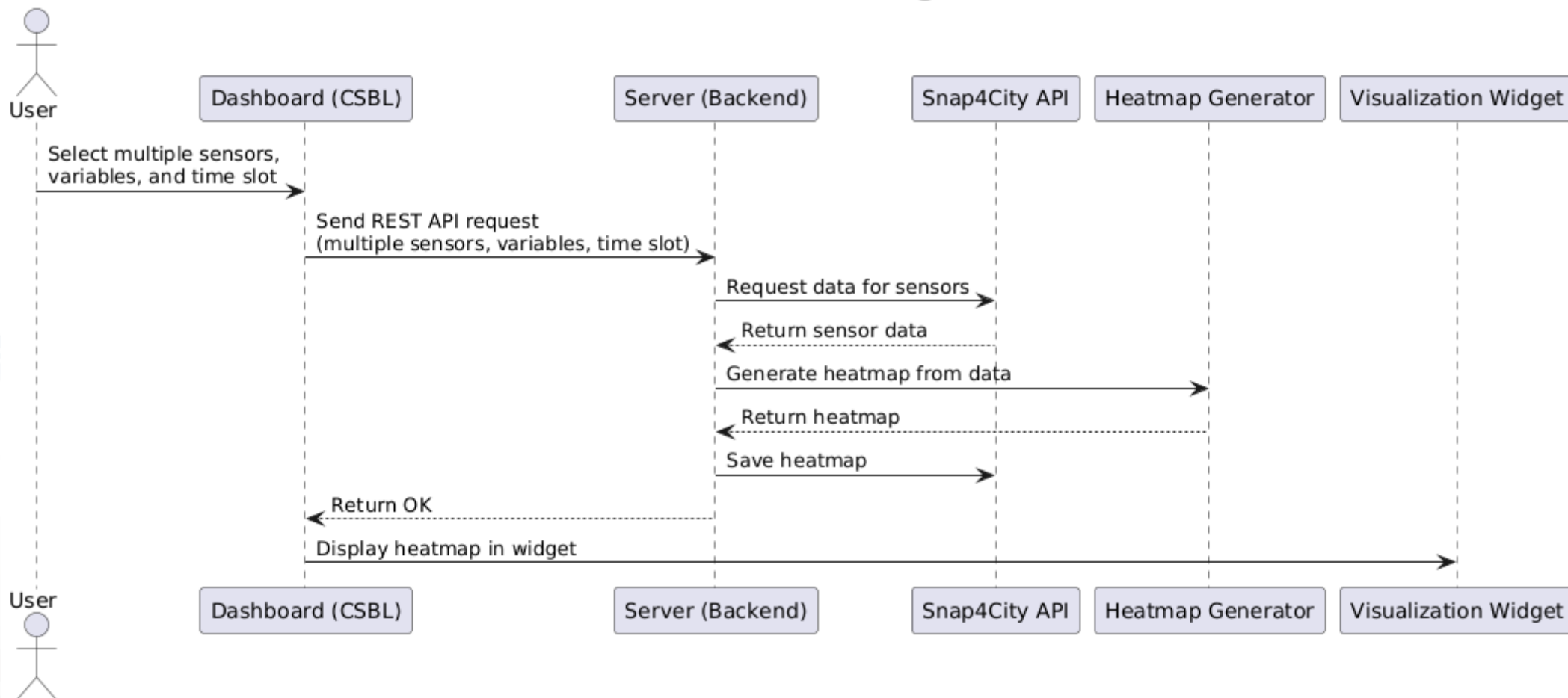
Enable multi-threading:  
Each request is handled in a separate thread.

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

# Prediction



# Heatmap



- API Heatmap:
  - [https://github.com/disit/snap4city/blob/master/Computing/predictions/heatmap\\_service.py](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](https://github.com/disit/snap4city/blob/master/Computing/predictions/predictions_service.py)