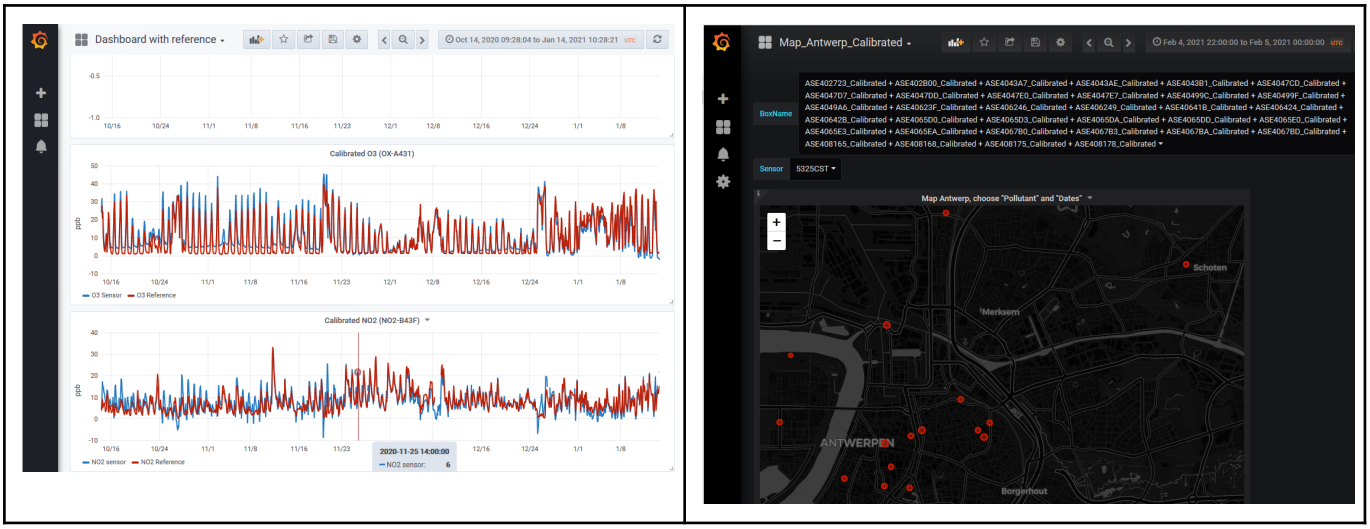
AirSensEUR: Setting Automatic Calibration of sensors and displaying with InfluxDB/Grafana



Yatkin, S., Gerboles, M., Signorini M. and Ferretti F.

Version 2021-05-17

1. Installing the necessary software **3**

2. List of AirSensEUR boxes for this guide **3**

3. Configuration of AirSensEUR boxes **3**

4. Calibration of AirSensEUR box (424A56) **7**

5. New AirSensEUR boxes configuration by cloning an existing one **7**

5.1. Using the ASE_App.R 8

5.2. Run Script Cloning_ASE_Boxes.R if you want to create a batch of new ASE boxes 9

6. Download, predict data and upload of calibrated data **9**

6.1. Updating PROXY flag 9

6.2. Using script Update_Influx.R 9

7. Automatic update of calibrated data in InfluxDB **11**

7.1. Using a Cron job under a Linux server 11

7.2. Using task scheduler under windows 11

8. Display data with grafana **13**

8.1. Dashboard with time-series data 13

8.2. Maps 16

The scripts needed in this document can be found at https://github.com/ec-jrc/airsenseur-calibration

1. Installing the necessary software

To install R, Rstudio and all needed scripts, follow instructions of section 3 of the ASE_App manual at https://docs.google.com/document/d/1EYMYecUhdWfg0tKaOhZRFMb2Rbk-SQUPXWqXbw3BuII/edit?usp=sharing

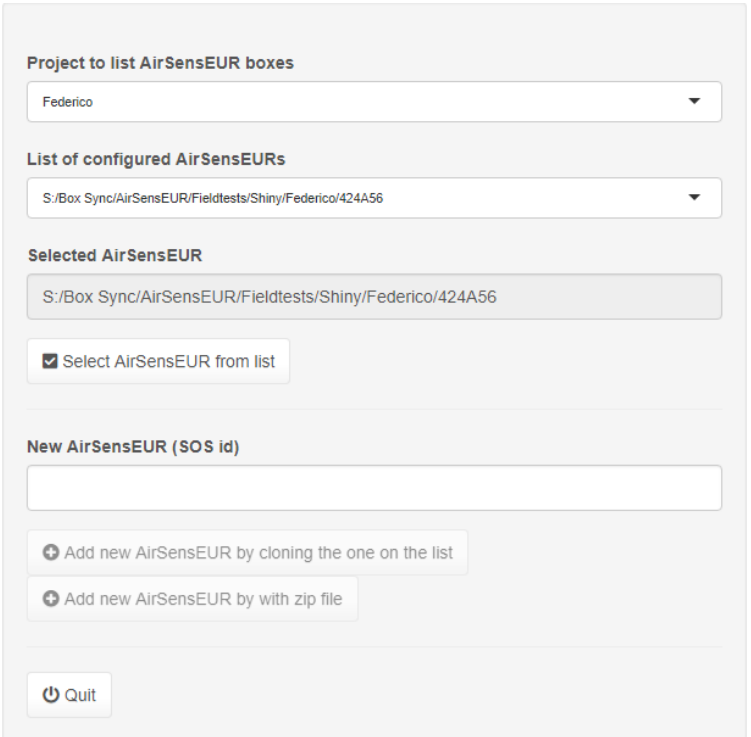
2. List of AirSensEUR boxes for this guide

The list of AirSensEUR boxes (ASE boxes) used for this example includes: 424A56, 42A57, 42D501, 425D0A, 425D0B and 4278FD

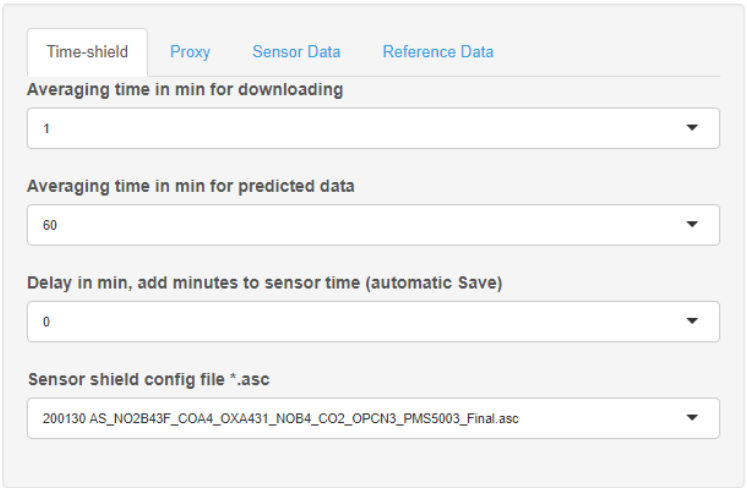
3. Configuration of AirSensEUR boxes

The configuration files of at least one box among the 6 ASE boxes to set up shall be available. The simplest way to configure an ASE box is to use the ASE_App (app.R) available in the github using the configuration of a box already configured. In the following example, the ASE box 424A56 is used. However, it would be possible to manually create all the necessary config files.

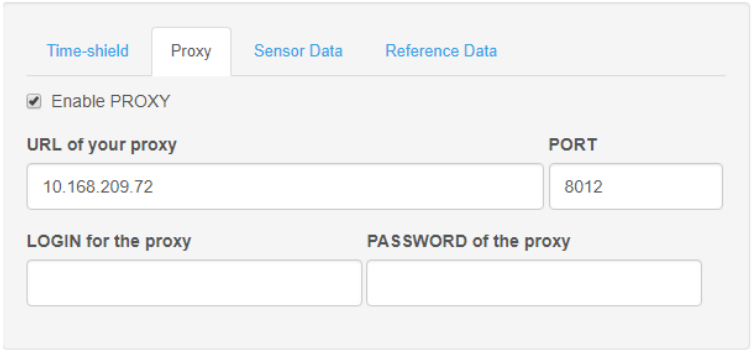
1. Under the ASE_App directory (~/Shiny) create a subdirectory where the ASE boxes configuration and data will be stored, for example "Federico". Be sure to install a previously configured ASE boxes with similar sensor configuration with which you can duplicate the configuration files of the new ASE boxes configuration under "Federico"
2. Run RStudio
3. Open file file app.R, run the App.
4. The ASE App_open in a browser or directly under RStudio.
5. If needed follows instruction in 5.1. Using the ASE_App.R to clone a new ASE box from an existing one.
6. Select the project (e.g. Federico), ASE box (e.g. 424A56) and click on "Select AirSensEUR for List".



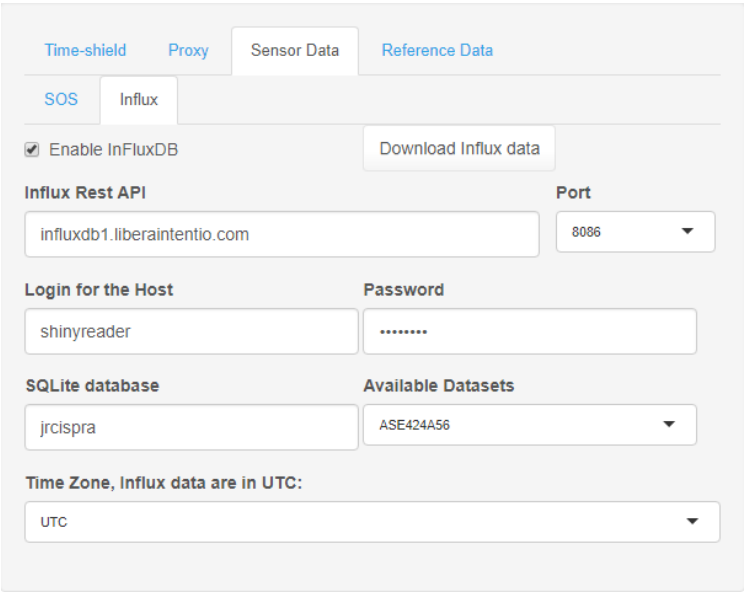
7. Under NavBarMenu "GetData", Tab "Time.shield" set parameters as below to get minute data from InfluxDB and hourly predicted data :



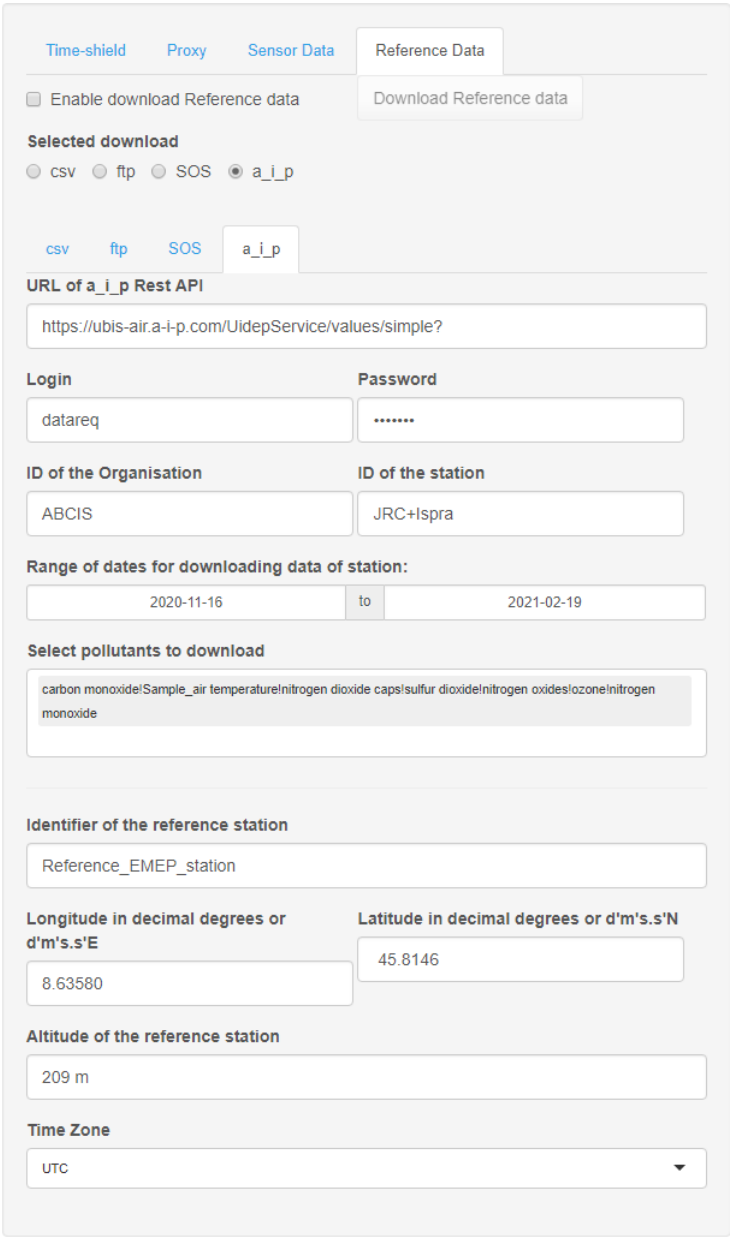
8. Defined a PROXY if needed under Tab “Proxy” set parameter for your Proxy if needed



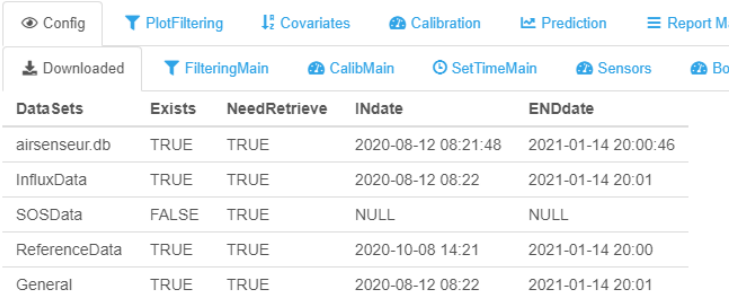
9. Under Tab “Sensor Data”, tab “Influx”, set the parameters for InfluxDB download (url of Rest Api, port - 8086 or 3030, login, password, name of SQLite database and datasets name - ASE+AirSensEUR box ID). Click on “Enable InfluxDB”. Wait message “Server is up”. All data are downloaded when clicking on “Download Influx Data”.



10. Uncheck “Enable InfluxDB”
11. Under Tab “Reference Data” set the downloading of Reference data. See ASE_App manual.

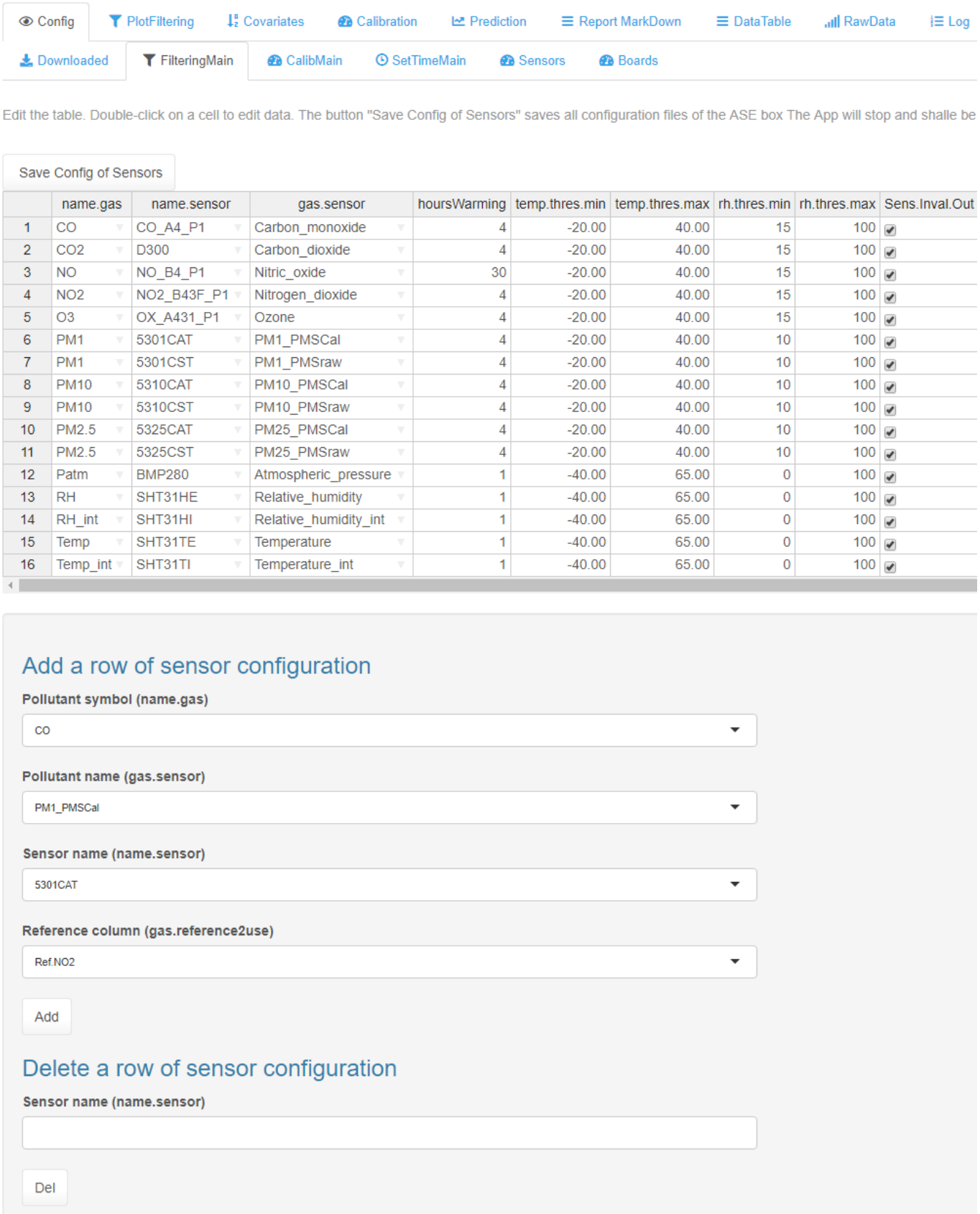


12. Click on NavBarMenu “dataTreatment”. Wait for filtering calibration and prediction of sensor data
13. Observe your available data under Config|Downloaded



DataSets	Exists	NeedRetrieve	INdate	ENDdate
airsenseur.db	TRUE	TRUE	2020-08-12 08:21:48	2021-01-14 20:00:46
InfluxData	TRUE	TRUE	2020-08-12 08:22	2021-01-14 20:01
SOSData	FALSE	TRUE	NULL	NULL
ReferenceData	TRUE	TRUE	2020-10-08 14:21	2021-01-14 20:00
General	TRUE	TRUE	2020-08-12 08:22	2021-01-14 20:01

14. Under Config|FilteringMain, Config|CalibMain and Config|SetTimeMain make sure that you correctly set the list of sensors you want to configure.



15. Use ASE_App manual to delete sensors.
16. For example to delete sensor OPCN3PM25 select the sensor under “Delete a row of sensor configuration|Sensor mame (Name.Sensor)”, click on button “Del” below and the sensor row is discarded from the table of configured sensors.
17. Use ASE_App manual to add sensors.
18. For example to add sensor OPCN3PM25 select: PM2.5 under “Add a row of sensor configuration|Pollutant symbol (name.gas), Particulate_Matter_25 under “Add a row of sensor configuration|Pollutant name (gas.sensor)”, OPCN3PM25 under “Add a row of sensor

configuration|Sensor name (name.sensor)" and Ref.PM2.5 under "Add a row of sensor configuration|Reference column (gas.reference2use)", click on button "Add" below and the sensor row is added at the bottom of the table of configured sensors. Be sure to correctly set all columns in the 3 tables Config|FilteringMain, Config|CalibMain and Config|SetTimeMain. "Pollutant symbol (name.gas), "|Pollutant name (gas.sensor)" and "Sensor name (name.sensor)" can be found at https://docs.google.com/spreadsheets/d/1e1MJm8ut6s8UXP8lsMwij58CyJhWsvVuz2tdLWMpMpI/edit?usp=sharing

19. In order to update the configuration, click on button "Save Config of Sensors" when all sensors are correctly configured. The ASE_App is automatically closed and shall be restarted to take change into account.
20. If you do not need to change any sensor in Config|FilteringMain , click on button "save" in the sidebar layout of NavBarMenu "Data Treatment" in order to save configuration files and data.

4. Calibration of AirSensEUR box (424A56)

In order to proceed with calibration it is necessary to add reference data within the ASE-App. See the ASE_App manual to find explanation how to insert reference data for an ASE Box (https://docs.google.com/document/d/1EYMYecUhdWfg0tKaOhZRFMb2Rbk-SQUPXWqXbw3BuII/edit?usp=sharing).

The aim of the calibration is to establish calibration models, save them in files and register these calibration in the configuration files 424A56.cfg for the sensors to calibrate. Calibration is carried out using script Compare_Model.R. The script includes default calibration options per sensor stored in variable Cal.Param. It is suggested not to change this variable unless you do understand what you are doing. (see https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration).

Please follow the following steps:

1. Update variable "Possible.Dir" making sure that the the ASE_App directory (~/Shiny) is included
2. Update variable "Project", the sub directory of "Possible.Dir" where the ASE boxes are stored
3. Update variable "List .ASE", the IDs of the ASE boxes to calibrate
4. Update variable "Cal.Sensors", the name of the sensors that are going to be calibrated (for example: c("CO_A4_P1", "NO_B4_P1", "NO2_B43F_P1", "OX_A431_P1") .
5. Update variable "Place", the name of the place where the calibration took place with colocation of sensor and reference data
6. Update variable "Cal_Interval", the time interval for calibration
7. Update variable "Predict_Interval", the time interval with colocation of ASE boxes and reference data for testing the calibration
8. Run the script

The calibration models are established and saved into the configuration file 424A56.cfg.

In this example only the electrochemical sensors for CO, NO, NO2 and O3 are calibrated, according to the value of variable "Cal.Sensors".

5. New AirSensEUR boxes configuration by cloning an existing one

When new ASE boxes have to be configured, it is necessary to clone an already configured ASE box with already existing calibration models. After cloning a box, all configuration files will be created and the ASE boxes will be using the same calibration files as the ASE box used for cloning. However, it is possible to establish dedicated calibration for the cloned ASE boxes using the file Compare_Model.R (see 4. Calibration of AirSensEUR box (424A56)) once the ASE boxes configurations are cloned.

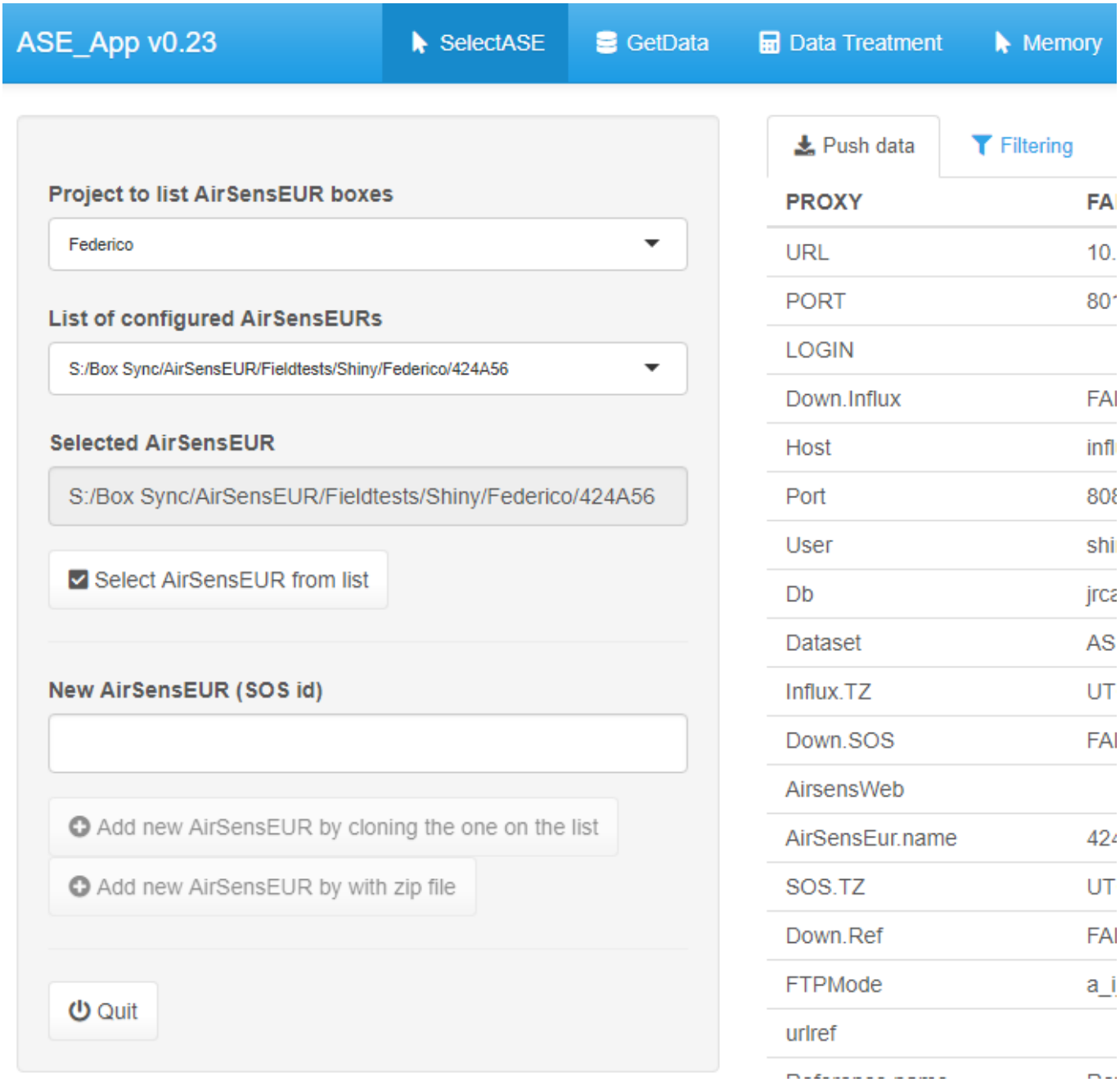
There are two possibilities to configure new ASE Boxes:

- Using the ASE_App.R
- Run Script Cloning_ASE_Boxes.R if you want to create a batch of new ASE boxes

5.1. Using the ASE_App.R

Please follow the following steps:

1. Run the ASE_App in R studio
2. Select the project, the “List of Configured AirSensEURs” will be updated listing all existing ASE boxes.
3. Select the ASE box you want to use for cloning under “List of configured AirSensEURs”
4. Do not click on the button “Select AirSensEUR from List” otherwise it is not possible to clone the selected ASE box to a new ASE box. The “Select AirSensEUR” box shall remain empty.
5. In “New AirSensEUR (SOS ID)” enter the ID of the new ASE box that you want to configure. The correct spelling of the ID is very important because the ID will be used to download sensor data from InfluxDB.
6. Click on button “Add new AirSensEUR by cloning the one on the list”



5.2. Run Script Cloning_ASE_Boxes.R if you want to create a batch of new ASE boxes

The script for cloning ASE boxes, Cloning_ASE_Boxes.R, can be found at https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration
Please follow the following steps:

1. Update variable “Possible.Dir” corresponding to the the ASE_App directory (~/Shiny)
2. Update variable “Project”, the sub directory where the ASE boxes are stored
3. Update variable “Cloning.path”, the file path of the the ASE box used for cloning
4. Update variable “List.ASE”, the list of IDs new ASE boxes to be cloned. The correct spelling of the IDs is very important because the IDs will be used to download sensor data from InfluxDB.
5. Run the script

6. Download, predict data and upload of calibrated data

6.1. Updating PROXY flag

It might be that the PROXY flag in file ./Project/Configuration/ASE_Servers.cfg is set to TRUE. If the server that will run Update_Influx.R has no configured PROXY, it is then necessary to set this flag to FALSE in order to avoid a timeout error.
All ./Project/Configuration/ASE_Servers.cfg can be edited manually to set the flag to FALSE.
Otherwise use the script Set_Proxy_FALSE.R found at https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration
Please follow the following steps:

1. In RStudio Server, open the file Set_Proxy_FALSE.R
2. Update variable “Possible.Dir” corresponding to the the ASE_App directory (~/Shiny)
3. Update variable “Project”, the sub directory where the ASE boxes are stored, e. G. “Federico”
4. Update variable “DIR_Config”, the sub directory where the config file ASE_Server.cfg are stored.
5. Update variable “List.ASE”, the list of IDs of the ASE boxes to be updated, e.g. 424A56, 42A57, 42D501, 425D0A, 425D0B and 4278FD.
6. Run the script

6.2. Using script Update_Influx.R

The script for downaloding and calibrating ASE boxes, Update_Influx.R, can be found at https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration
If the ASE.cfg has a calibration file registered for each sensor of the ASE box, the script Update_Influx.R can be run as it is. The aim of this script is to :

- download sensor data from InfluxDB (and reference data in requested into ASE.cfg),
- filter all data using the ASE configuration files,
- apply the calibration models,
- compute averages according to parameter UserAvgMins in ASE_sever.cfg,
- upload all calibrated data into a new InfluxDB table names ASE424A56_Calibrated on the Influx Server.

Please follow the following steps:

1. Run RStudio
2. Open file Update_Influx.R
3. Update variable “Possible.Dir” making sure that the directory where the R scripts lay is included. This directory corresponds to the ASE_App directory (~/Shiny)
4. Update variable “Project”, the sub directory of the ASE_App directory where the ASE box directory trees are stored
5. Update variable “Dir_General” and “Dir_config”, the sub directory names of the ASE boxes where the data and configuration files of each ASE boxes are stored
6. Update variable “DIR_Results”, the sub directory where the calibrated data will be stored.

7. Update variable “List.ASE”, the list of IDs of the ASE boxes to be updated, e.g. 424A56, 42A57, 42D501, 425D0A, 425D0B and 4278FD.
8. Update variable “Parallel”, that control if the script is run in parallel (several cores processing the same number of ASE boxes as the number of cores) or sequential computing (one core processing one ASE box at a time). Three values are possible: "Future_apply", "ForEach" and " Sequential", the former ones use parallel computing.
    - "ForEach": this is the fastest parallel computing and creates a log file. However, ForEach seems to have problemto release memory under Unix and should be avoided for this Operating System. If the “ForEach” method is chosen, update variable List.ASE.width” that allow working in chunks of ASE boxes.Typically, List.ASE.width should be equal to the number of cores.
    - "Future_apply": it is slower than the "ForEach" method and does not allow saving a log file, it only print a few messages when computing ends. However, this method is more robust and should be preferred under Linux.
    - "Sequential": no parallel computing, only sequential. Extremely slow but save a log file which is sorted by ASE boxes
9. If this script is run in parallel computing, update the variables “Max.cores” and “Free.cores” the maximum number of cores dedicated to this script and minimum number of unused cores.
10. Update logical argument “Add.Ref” in calling function influx.downloadAndPredict(), if TRUE reference data are added in the list to the list of calibrated sensor data to upload to the Influx sensor. Please take care if reference data are available for each ASE box in List.ASE. i is the index of ASE box into List.ASE. Requesting to upload reference data if they are not available will result in a crash of the script.
11. Update user and pass in Upload2Influx function call. A user with write credentials in InfluxDB.
12. Update the names of reference analysers in the argument Ref_analysers of the Upload2Influx function call if needed.
13. Run the script

Except for the “Future_apply” method, thes script creates a log file, named Log_Parallel.txt, in ~/Shiny with console messages in order to observe info, warning and error messages.

In directory DIR_Results, a csv file with calibrated data is created for each ASE box listed in List.ASE. The name of the file is ASE424A56_Calibrated.csv. It is a sequential file (long format) with the following headers:

- Time, datetime with UTC time zone in all rows
- ID_ASE, ID of ASE box in all rows
- Latitude, in decimal degrees if available
- Longitude, in decimal degrees if available
- Variable, name of the pollutant in all rows (see table below)
- Name.sensor, brand model of the sensor in all rows (see table below)
- Serial, part number of the sensor as stored into the ASE box in all rows
- Value, numeric pollutant concentration level in all row
- Uncertainty, not yet implemented
- Sens.unit, sensor unit for value and uncertainty in all rows, it can be ppb, ppm, µg.m-3, mg.m-3, hPa, percent or Celsius

Name.sensor	Variable (sensor)	Name.sensor	Variable (reference)
CO_A4_P1	Carbon_monoxide	As set in Ref_Analysers	Ref.CO_ppm
D300	Carbon_dioxide	As set in Ref_Analysers	Ref.CO2
NO_B4_P1	Nitric_oxide	As set in Ref_Analysers	Ref.NO

NO2_B43F_P1	Nitrogen_dioxide	As set in Ref_Analysers	Ref.No2
OX_A431_P1	Ozone	As set in Ref_Analysers	Ref.O3
5301CAT	PM1_PMSCal	As set in Ref_Analysers	Ref.PM1
5301CST	PM1_PMSraw	As set in Ref_Analysers	Ref.PM1
OPCN3PM1	Particulate_Matter_1	As set in Ref_Analysers	Ref.PM1
5310CAT	PM10_PMSCal	As set in Ref_Analysers	Ref.PM10
5310CST	PM10_PMSraw	As set in Ref_Analysers	Ref.PM10
OPCN3PM10	Particulate_Matter_10	As set in Ref_Analysers	Ref.PM10
5325CAT	PM25_PMSCal	As set in Ref_Analysers	Ref.PM2.5
5325CST	PM25_PMSraw	As set in Ref_Analysers	Ref.PM2.5
OPCN3PM25	Particulate_Matter_25	As set in Ref_Analysers	Ref.PM2.5
BMP280	Atmospheric_pressure	As set in Ref_Analysers	Ref.Patm
SHT31HE	Relative_humidity	As set in Ref_Analysers	Ref.RH
SHT31TE	Temperature	As set in Ref_Analysers	Ref.Temp

This script creates a subdirectory as set by variable “Dir_results” under subdirectory “Project” with the calibrated data in csv format for each ASE box. Here below is an example file:

```
2020-08-17T03:00:00Z,424A56,,,Carbon_monoxide,CO_A4_P1,132770002,0.205839302283235,ppm
2020-08-17T04:00:00Z,424A56,,,Carbon_monoxide,CO_A4_P1,132770002,0.185832485787055,ppm
2020-08-17T05:00:00Z,424A56,,,Carbon_monoxide,CO_A4_P1,132770002,0.187809807450509,ppm
2020-08-17T06:00:00Z,424A56,,,Carbon_monoxide,CO_A4_P1,132770002,0.184110836413108,ppm
2020-08-17T07:00:00Z,424A56,45.8028491646689,8.87780811838155,Carbon_monoxide,CO_A4_P1,132770002,0.191265981781329,ppm
2020-08-17T08:00:00Z,424A56,45.8027878113959,8.87771035674331,Carbon_monoxide,CO_A4_P1,132770002,0.226326194085613,ppm
2020-08-17T09:00:00Z,424A56,45.8027952070114,8.87772618428675,Carbon_monoxide,CO_A4_P1,132770002,0.19089075050752,ppm
2020-08-17T10:00:00Z,424A56,45.8027970838767,8.87765639656857,Carbon_monoxide,CO_A4_P1,132770002,0.179253062912472,ppm
2020-08-17T11:00:00Z,424A56,45.80277316058,8.87766291271688,Carbon_monoxide,CO_A4_P1,132770002,0.210342077568943,ppm
2020-08-17T12:00:00Z,424A56,45.8027625952042,8.87766927757061,Carbon_monoxide,CO_A4_P1,132770002,0.194424178335888,ppm
2020-08-17T13:00:00Z,424A56,45.8027898293956,8.87770149511484,Carbon_monoxide,CO_A4_P1,132770002,0.155698103636009,ppm
2020-08-17T14:00:00Z,424A56,45.8028100035291,8.87770130602524,Carbon_monoxide,CO_A4_P1,132770002,0.167505013178069,ppm
2020-08-17T15:00:00Z,424A56,45.8028094830806,8.87770274949484,Carbon_monoxide,CO_A4_P1,132770002,0.0963343088766343,ppm
2020-08-17T16:00:00Z,424A56,45.8028505170434,8.87765018341248,Carbon_monoxide,CO_A4_P1,132770002,0.063821622622475,ppm
2020-08-17T17:00:00Z,424A56,45.8028260736606,8.87768216972167,Carbon_monoxide,CO_A4_P1,132770002,0.155644761935321,ppm
2020-08-17T18:00:00Z,424A56,45.8028041932017,8.8776803068755,Carbon_monoxide,CO_A4_P1,132770002,0.275444703574812,ppm
2020-08-17T19:00:00Z,424A56,45.802813142357,8.87757329618056,Carbon_monoxide,CO_A4_P1,132770002,0.247171762840897,ppm
2020-08-17T20:00:00Z,424A56,45.8027549146813,8.87762735542141,Carbon_monoxide,CO_A4_P1,132770002,0.252035054448501,ppm
2020-08-17T21:00:00Z,424A56,45.8027856085028,8.87769112066101,Carbon_monoxide,CO_A4_P1,132770002,0.251400472147206,ppm
2020-08-17T22:00:00Z,424A56,45.8027844952341,8.87761807315286,Carbon_monoxide,CO_A4_P1,132770002,0.241732748739656,ppm
2020-08-17T23:00:00Z,424A56,45.8027640579584,8.87766851260264,Carbon_monoxide,CO_A4_P1,132770002,0.242043602098841,ppm
```

7. Automatic update of calibrated data in InfluxDB

7.1. Using a Cron job under a Linux server

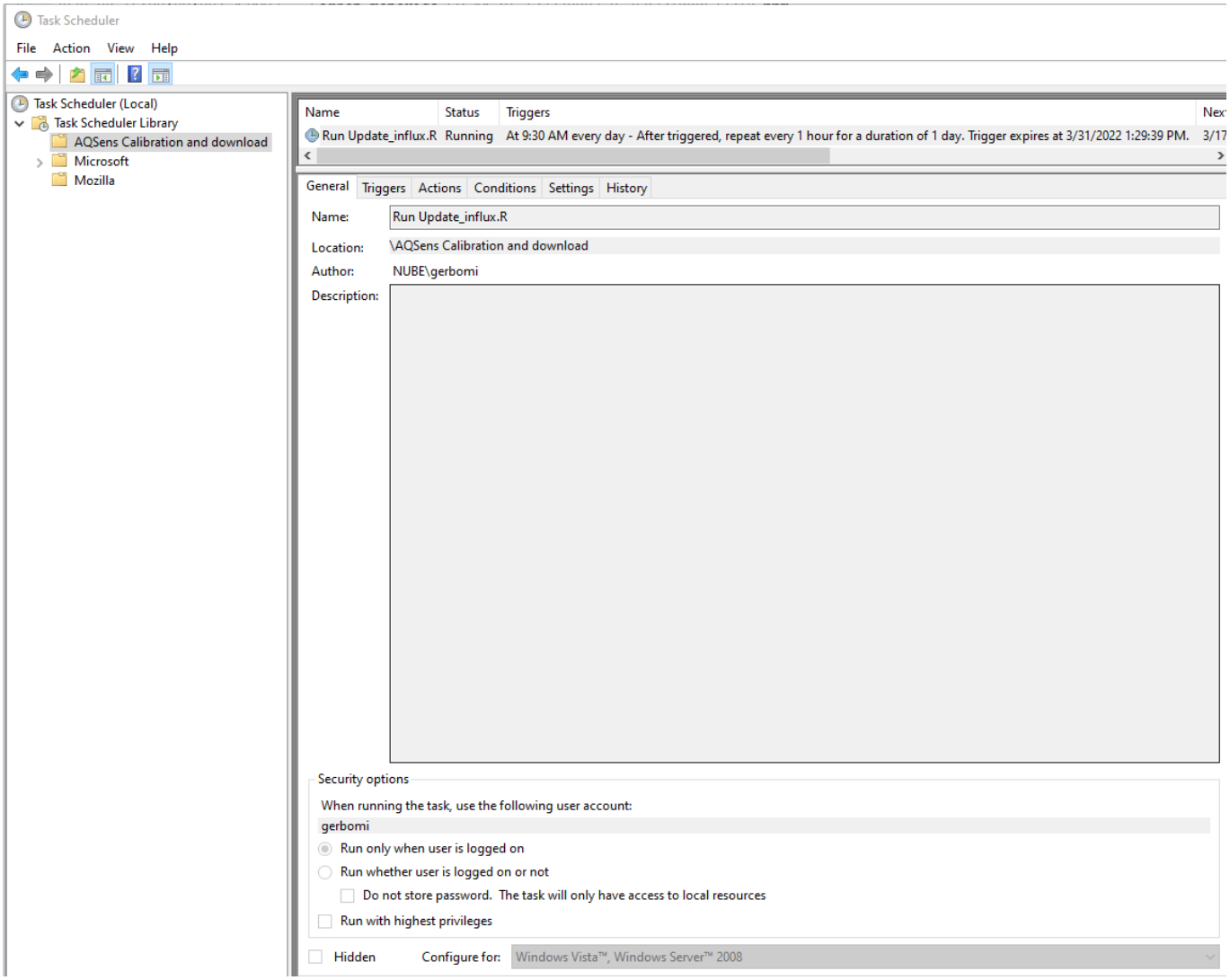
The automatic update of calibrated data is performed setting a CRON job file.
Create a text file (Cronjob.txt) with the following text assuming that the file path of the script is/home/shinyadmin/App and that a log of the process is saed and named `Download_Predict.log`:

```
30 * * * * cd /home/shinyadmin/App; Rscript Update_Influx.R >/home/shinyadmin/App/Download_Predict.log 2>&1
```

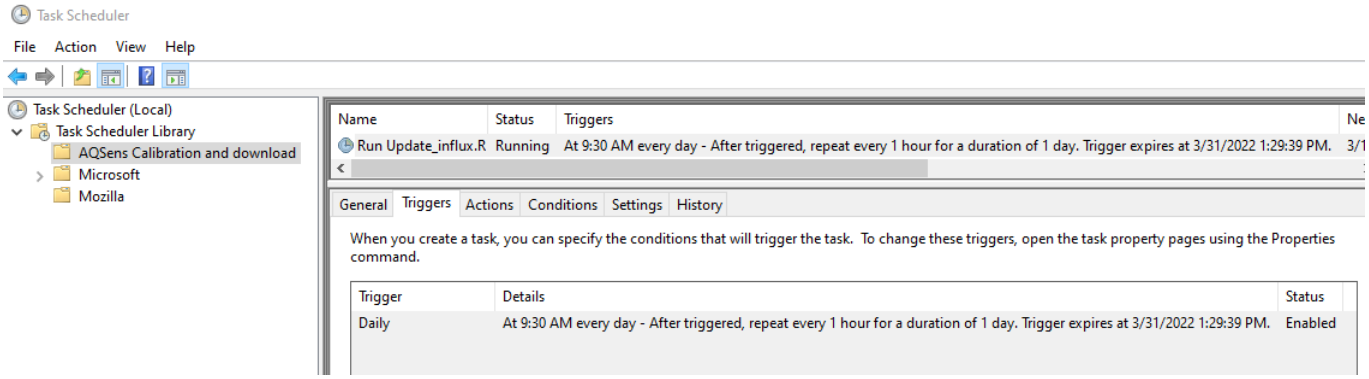
Add it into your crontab file. You can open the file by using command crontab –e

7.2. Using task scheduler under windows

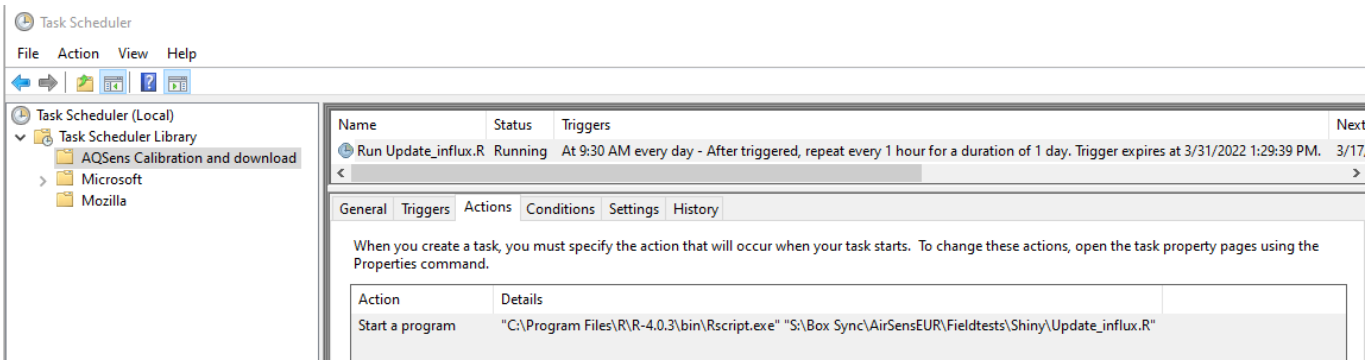
Open task scheduler under windows and create a job like:
General:



Triggers:



Action:

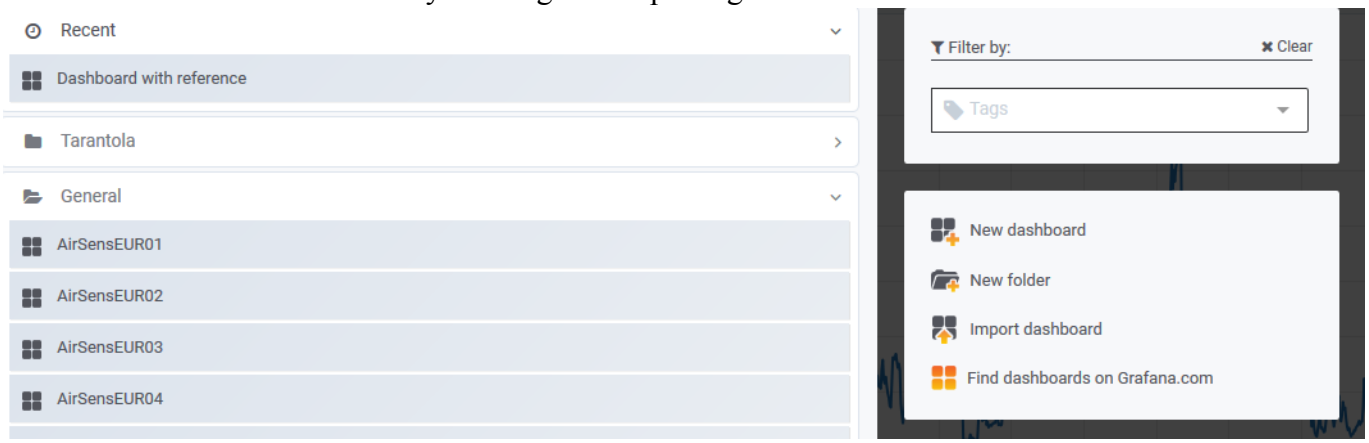


8. Display data with grafana

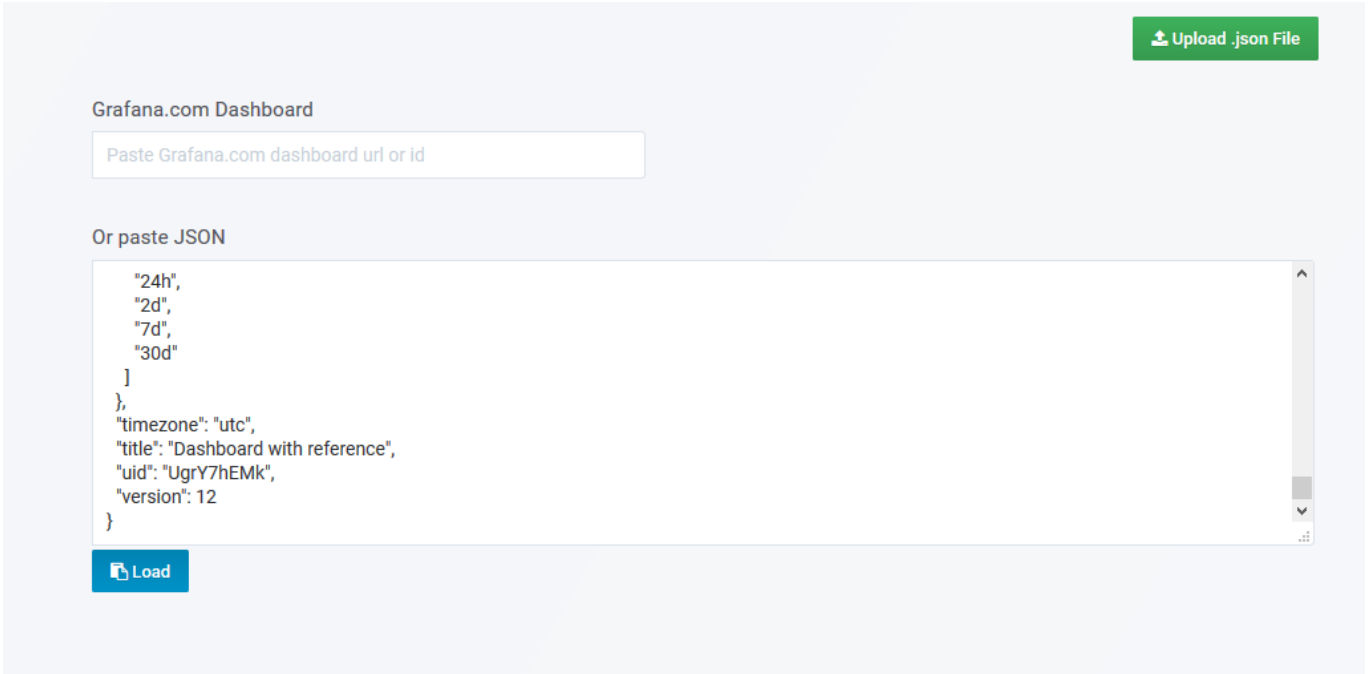
8.1. Dashboard with time-series data

Please follow the following steps:

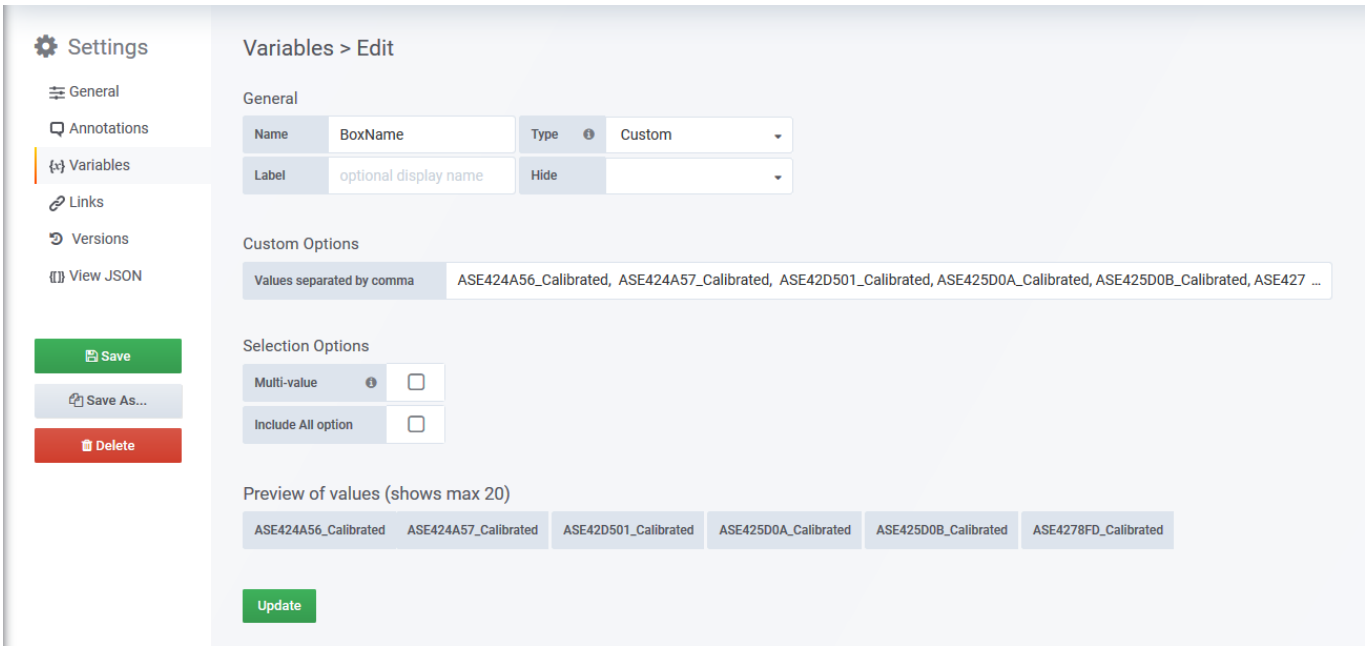
1. Create a new dashboard by clicking on “Importing dashboard”



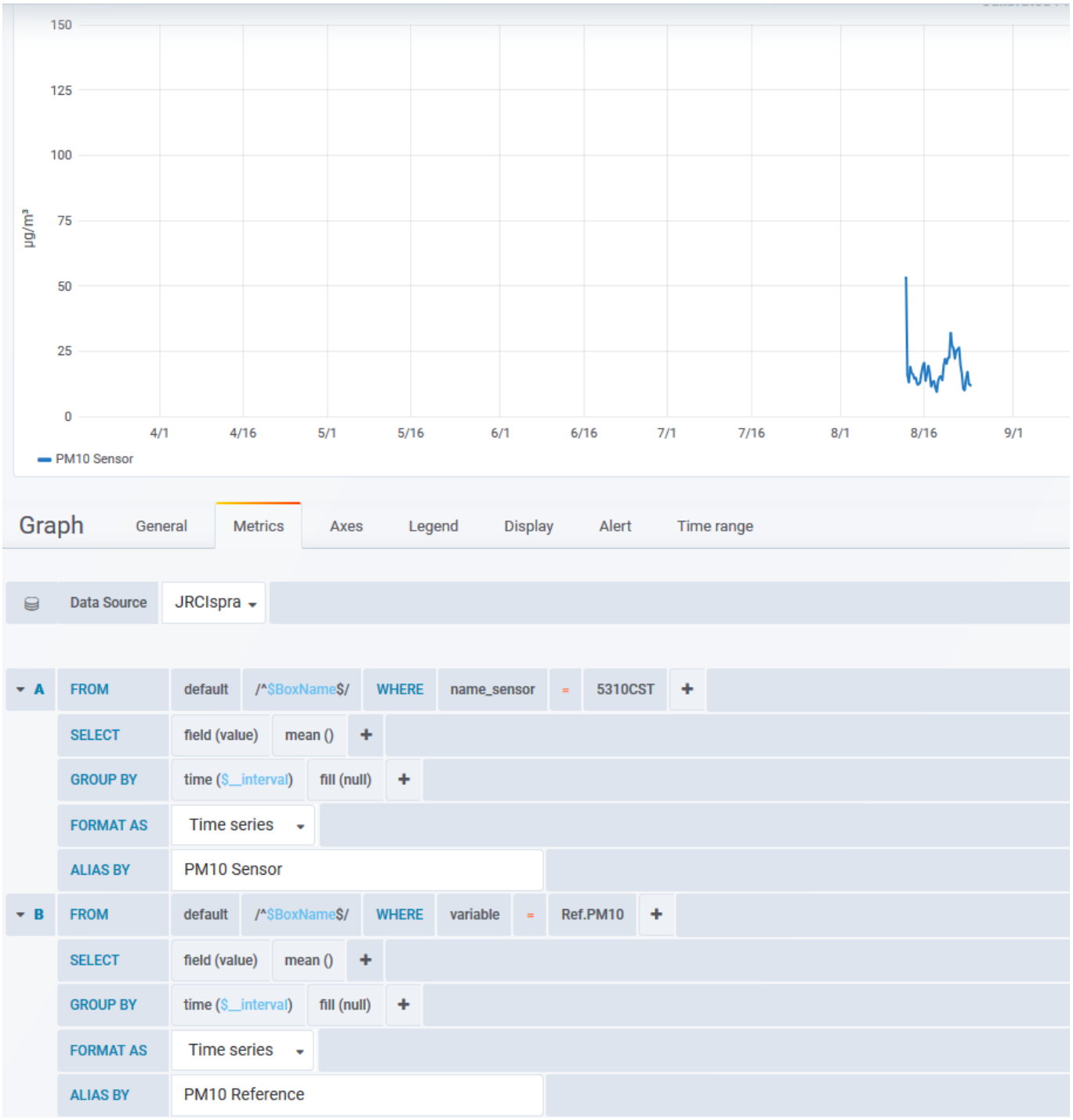
2. Paste the JSON code from JSON template “Dashboard with Reference”(https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration/Grafana) , click on load and set name



3. In settings|Variables, variable “BoxName”, update Values separated by comma with List.ASE, e.g. ASE424A56_Calibrated, ASE424A57_Calibrated, ASE42D501_Calibrated, ASE425D0A_Calibrated, ASE425D0B_Calibrated, ASE4278FD_Calibrated. Click on “Save”.

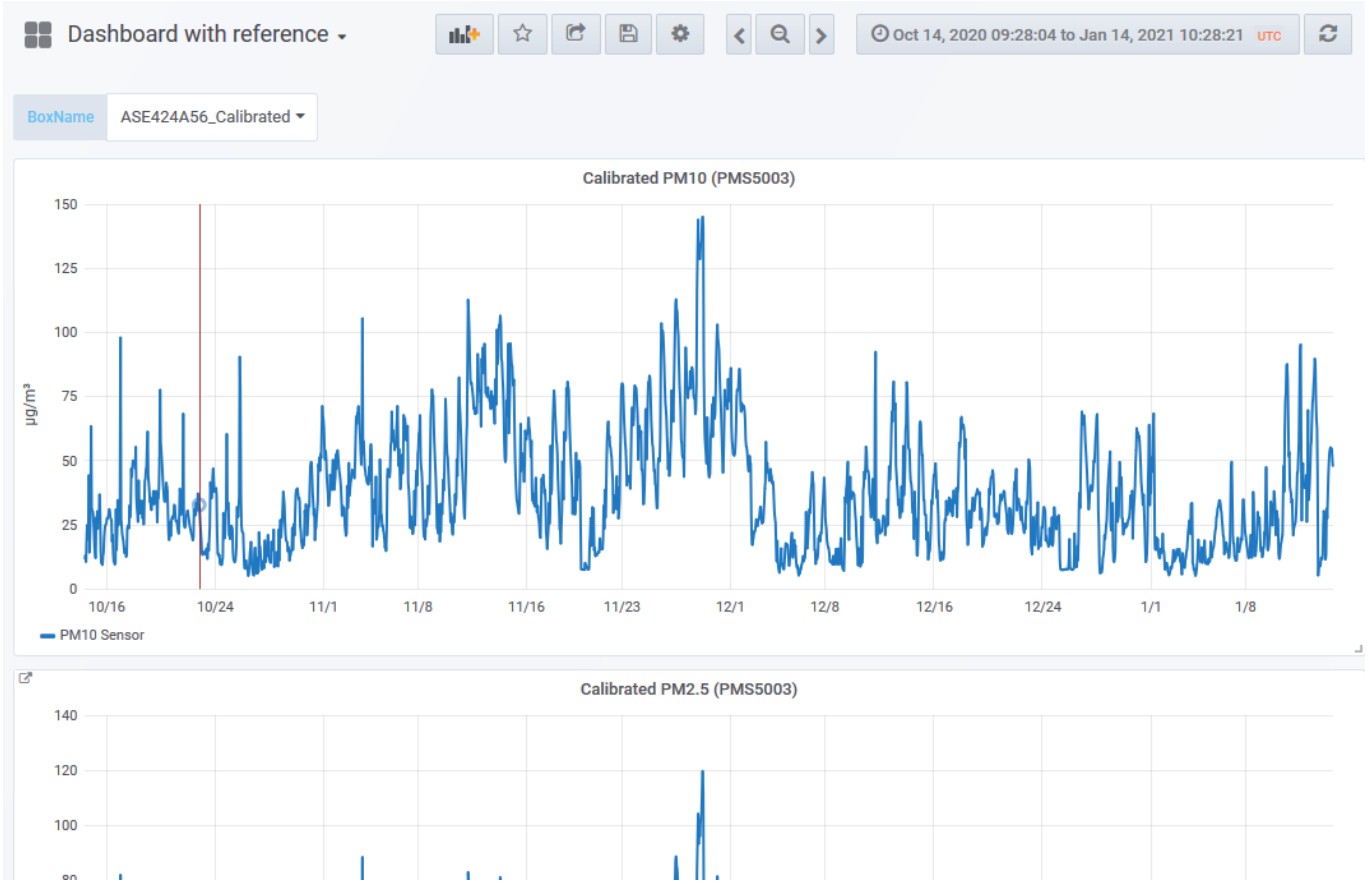


4. Update source: edit all plots, under tab “Metrics” set “Data Source” to the InfuxDB database name



5. In settings|Variables Update Values separated by comma with List.ASE, e.g. ASE424A56_Calibrated, ASE424A57_Calibrated, ASE42D501_Calibrated, ASE425D0A_Calibrated, ASE425D0B_Calibrated, ASE4278FD_Calibrated. Then click on button “Save”.

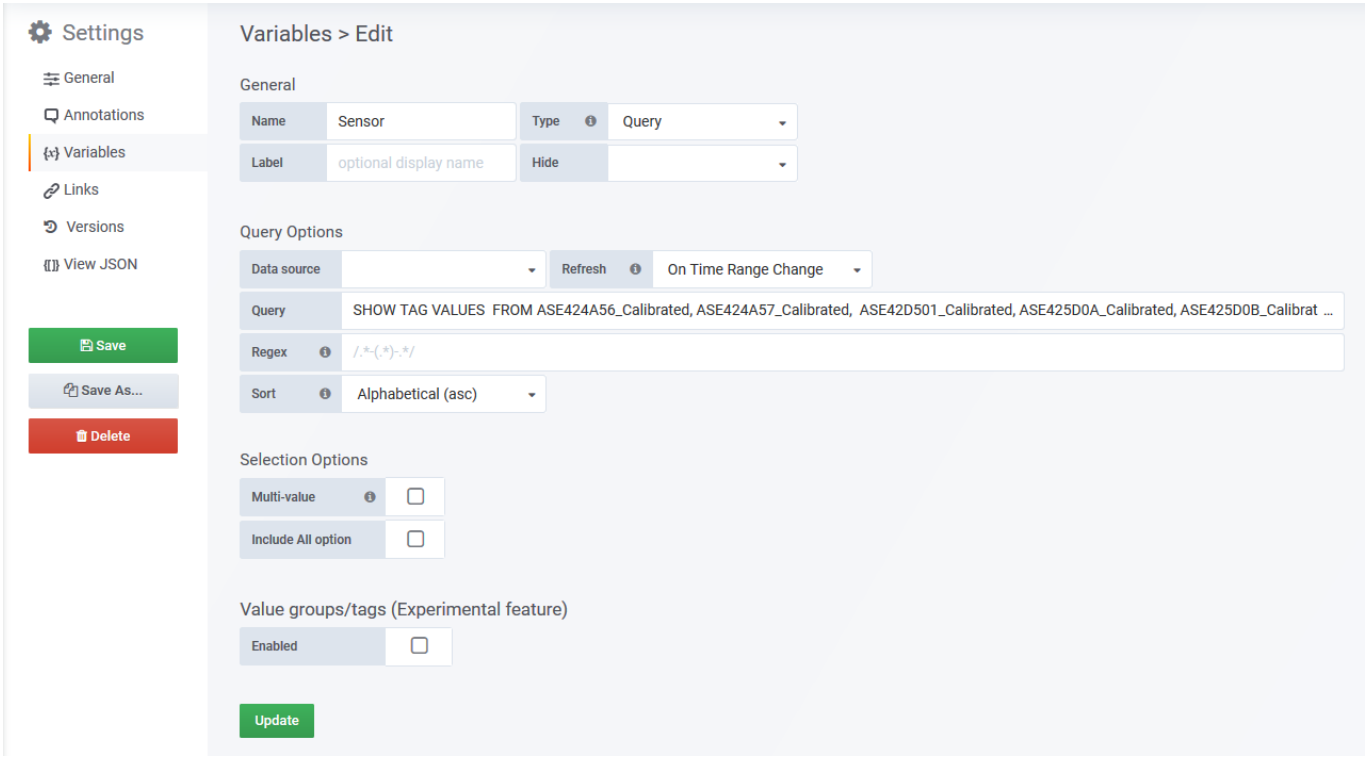
In the dashboard, it is possible to select which ASE box to display with calibrated sensor data (blue line) and reference data in red (if available) using the “BoxName” list and adapt the date interval on the upper left corner to display the desired time interval.



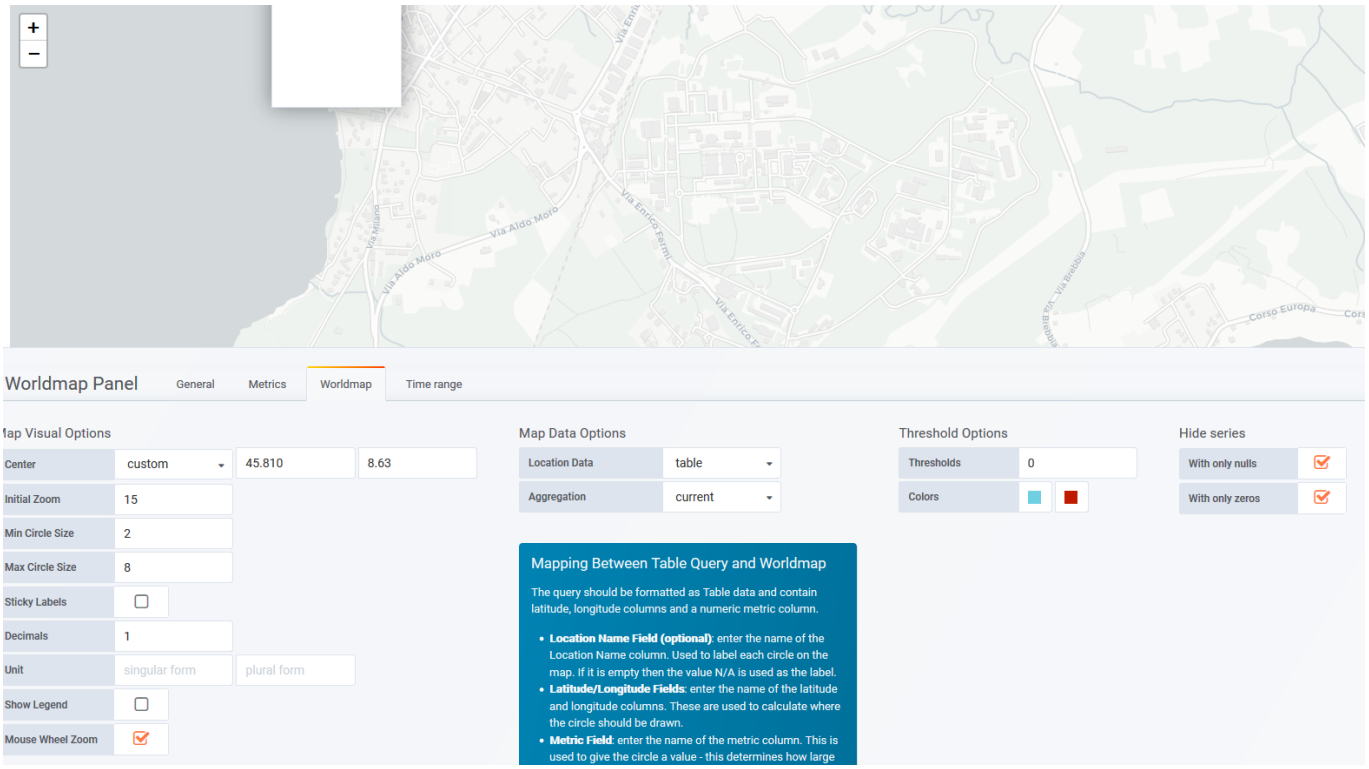
8.2. Maps

Please follow the following steps:

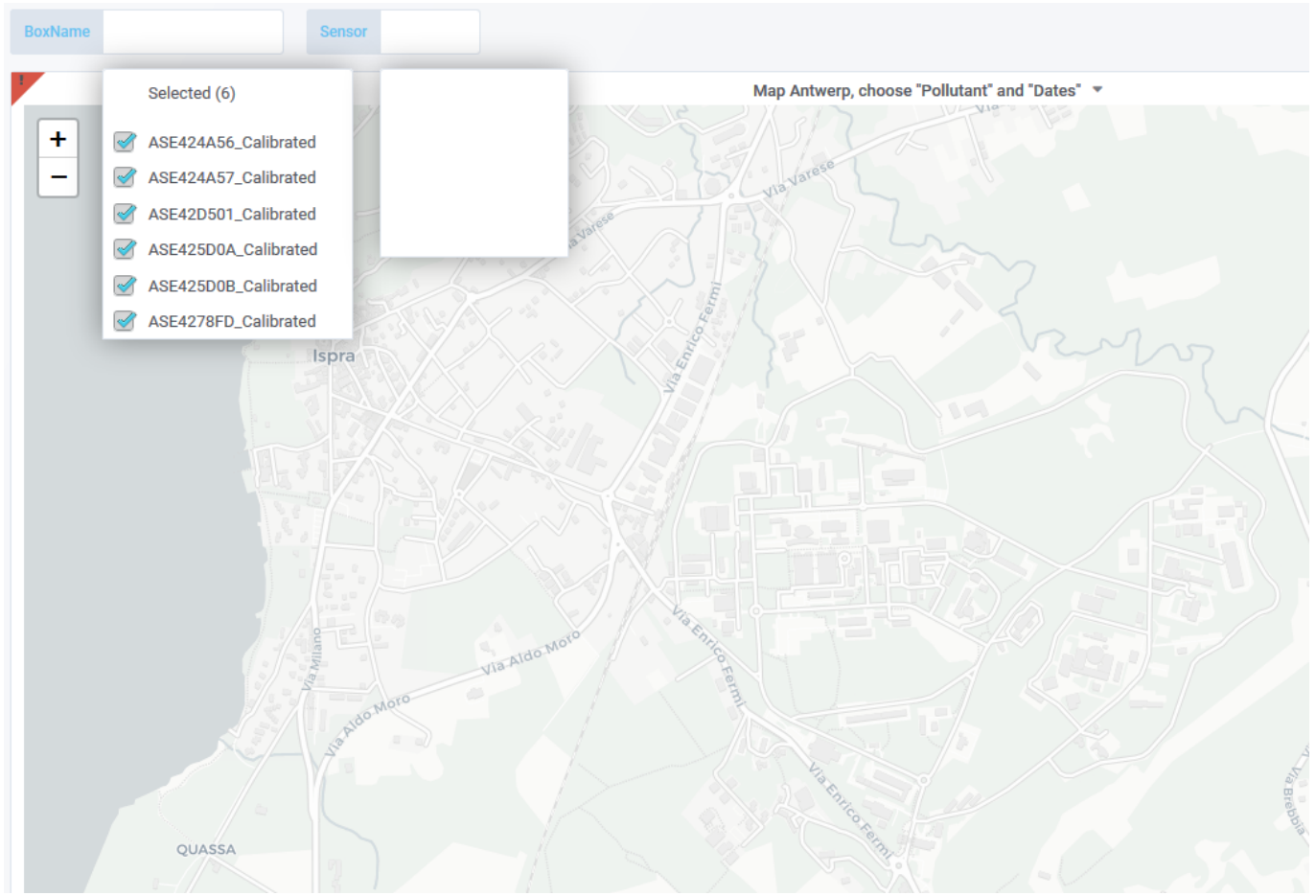
1. Repeat steps 1 to 3 of 8.1. Dashboard using JSON template Map_Calibrated (https://github.com/ec-jrc/airsenseur-calibration/tree/master/Auto_Calibration/Grafana)
2. In settings|Variables, variable “Sensor”, update “Data Source” to the name of the InfluxDB database and update “Query” with List.ASE IDs, e.g. SHOW TAG VALUES FROM ASE424A56_Calibrated, ASE424A57_Calibrated, ASE42D501_Calibrated, ASE425D0A_Calibrated, ASE425D0B_Calibrated, ASE4278FD_Calibrated WITH KEY = "name_sensor". Click on “Save”.



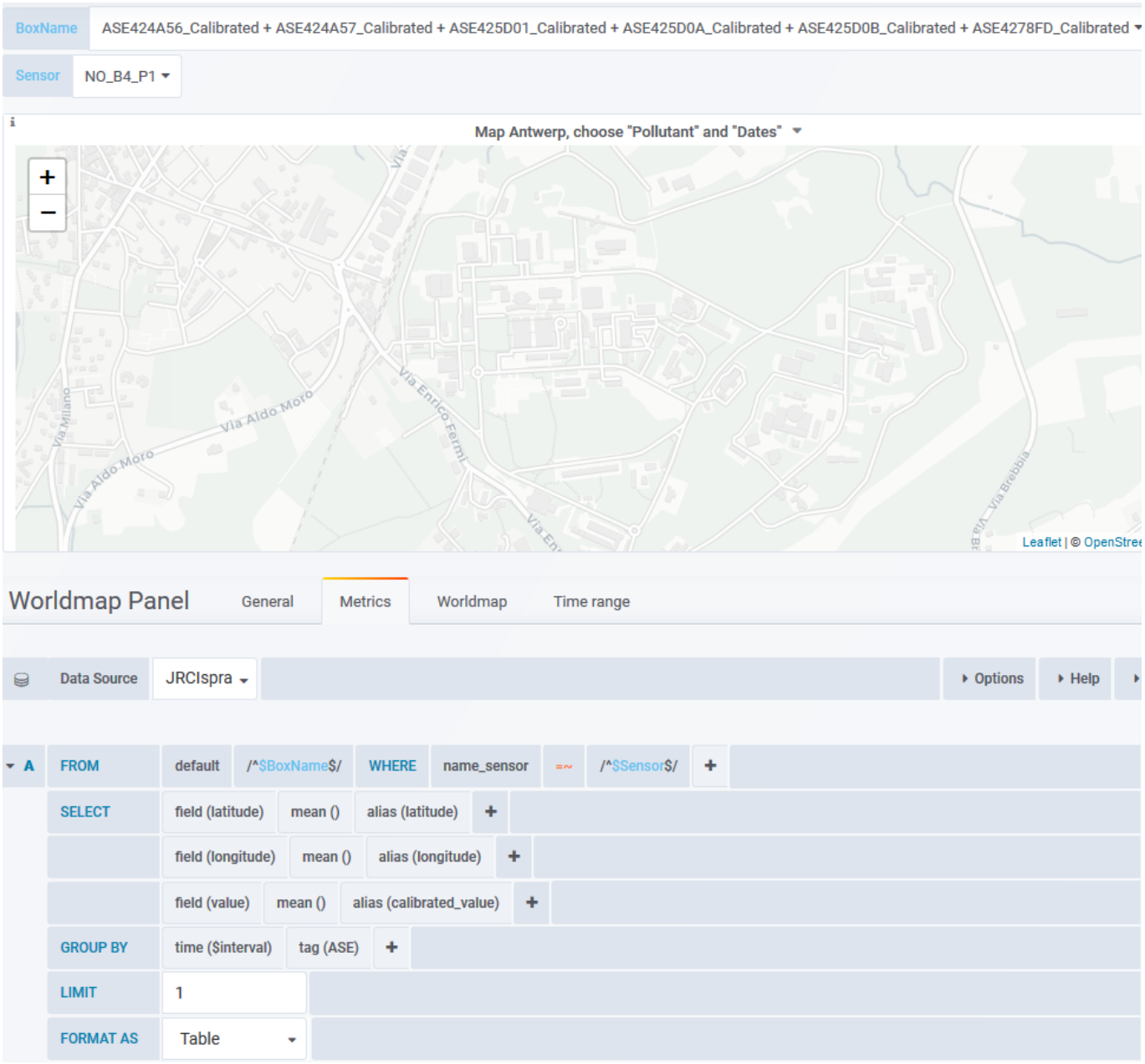
3. Update source: edit the map, under tab “Metrics” set “Data Source” to the InfuxDB database name
4. Update map coordinates: edit the map, under tab “Worldmap” set “Center” coordinates and “initial zoom”



5. In the Dashboard under BoxName”, select all ASE boxes.



6. Update source: edit the map in the dashboard, under tab “Metrics” set “Data Source” to the InfuxDB database nameData Source” to the InfuxDB database name



7. Make sure all ASE boxes are selected in the “BoxName”. Choose sensor in variable list “Sensor”

In the map, it is possible to select which sensor data to display using the “BoxName” list and adapt the date interval on the upper left corner to display the desired time interval. Sensor data are displayed with red dots whose size is proportional to the value of the pollutant levels at each ASE box location. When hovering the red dots a window opens giving the ASE ID and pollutant value.
The map may not display the pollutant values if the time interval is too wide (more than a month?). In this case an error message will be displayed.

9. Scripts to help in a network of ASE boxes

9.1. Editing Configuration files

Use file Updating_cfg.R to change any parameters in a list of ASE boxes. Follow the following points:

1. Set Dir.shiny, the file path where the script lays.
2. Set Project, the sub directory of shiny including all ASE box configuration files to edit
3. Set List.ASE.Config, the list of ASE boxes included into Project to be edited
4. Set List.Sensor, the sensor to be edited

5. Set Param2Change, the parameter of file the cfg file to be updated for sensor List.Sensor
6. Set, the Param2Change new value
7. Run the for loop to update the cfg files
8. If necessary set list.files2Delete to delete filtering index RDS files in order to update the List.ASE data.