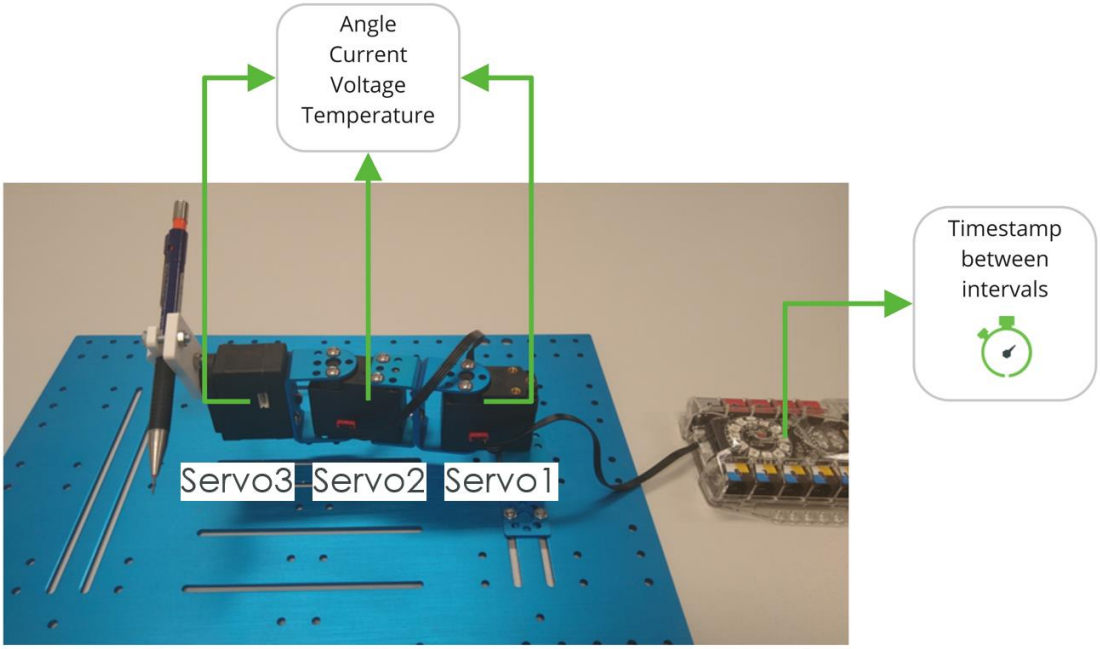
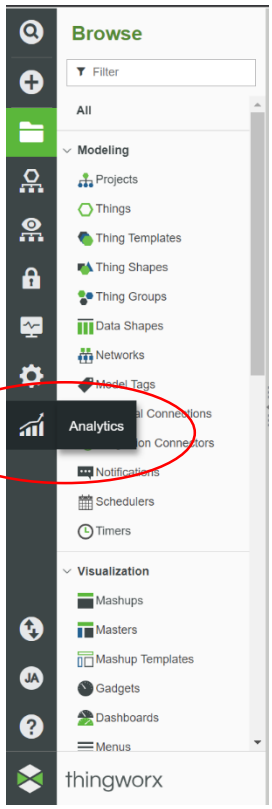
Create an Analytics model and use it with a Thing

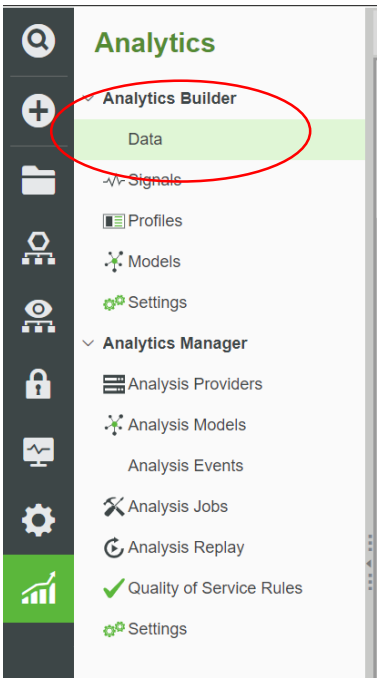
1. Demonstrator



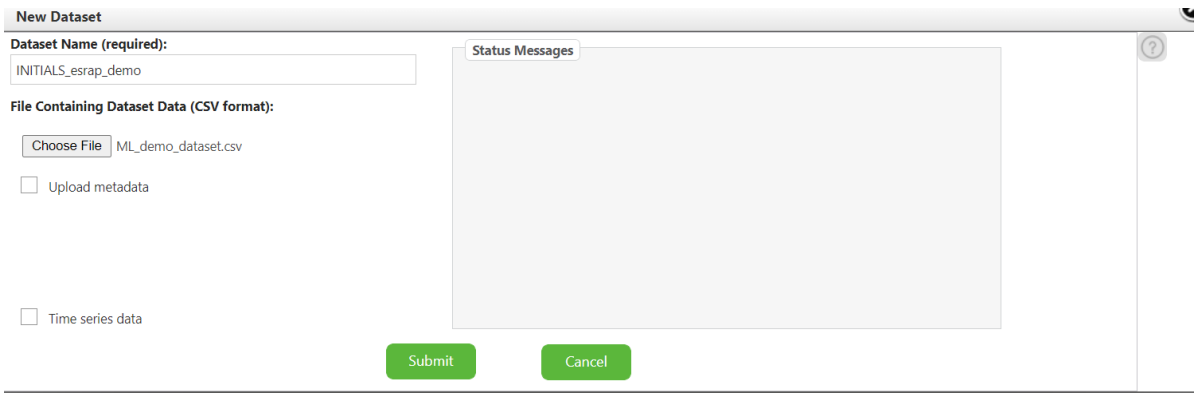
   a. Dataset: ML_demo_dataset.csv
2. In this exercise the position of Servo3 shall change if an additional force is applied to Servo1 (see video in the README.md file)
3. Navigate to Analytics on the menu bar on the left side of composer:



4. Select Data

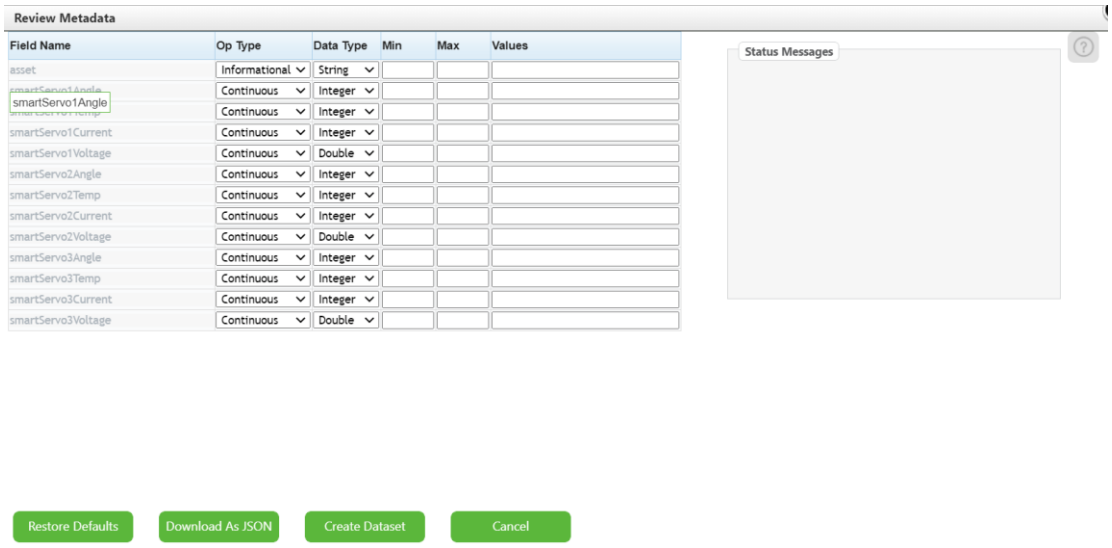


5. Create New dataset
    a. Name: INITIALS_esrap_demo
    b. Choose File -> navigate to the ML_demo_dataset.csv file on your PC
    c. Uncheck the boxes (see screenshot below)
    d. Submit

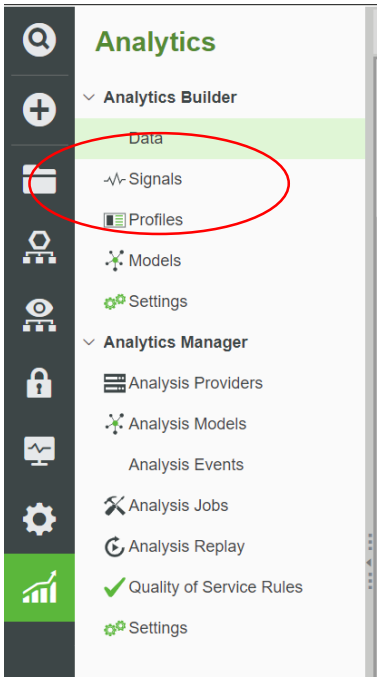


6. Add Metadata information
    a. Op Type:
        i. Asset = Informational
        ii. Everything else Continuous
    b. Data Type:
        i. Asset = String
        ii. Voltage values = Double
        iii. All other values = Integer

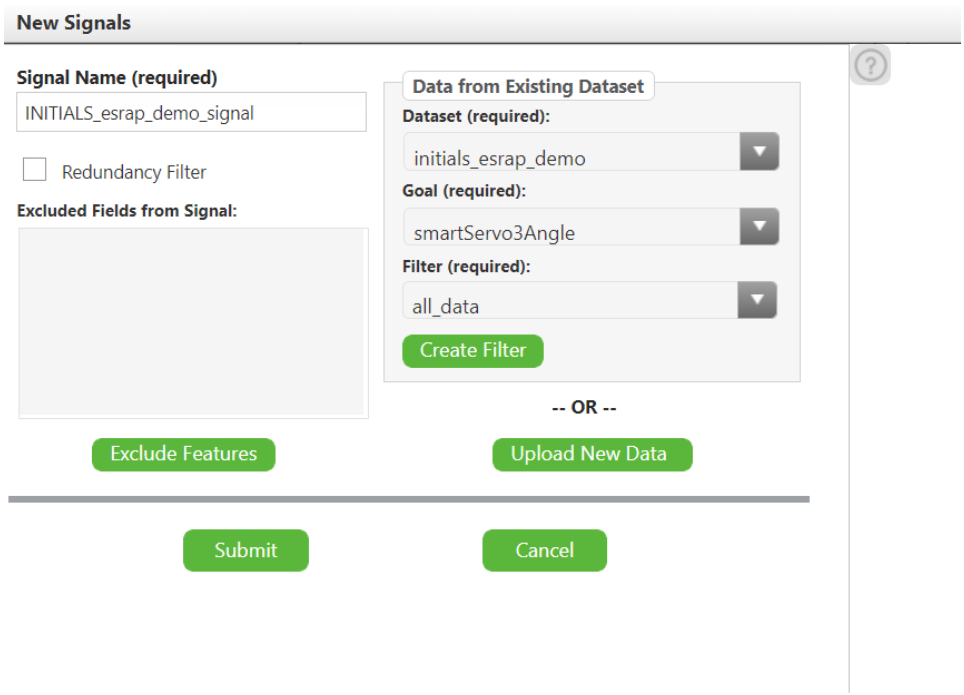
c. Download As JSON (If you want to recreate the dataset, you do not have to supply the Metadata information manually. You can use the JSON as source)
d. Compare with the image below
e. Create Dataset



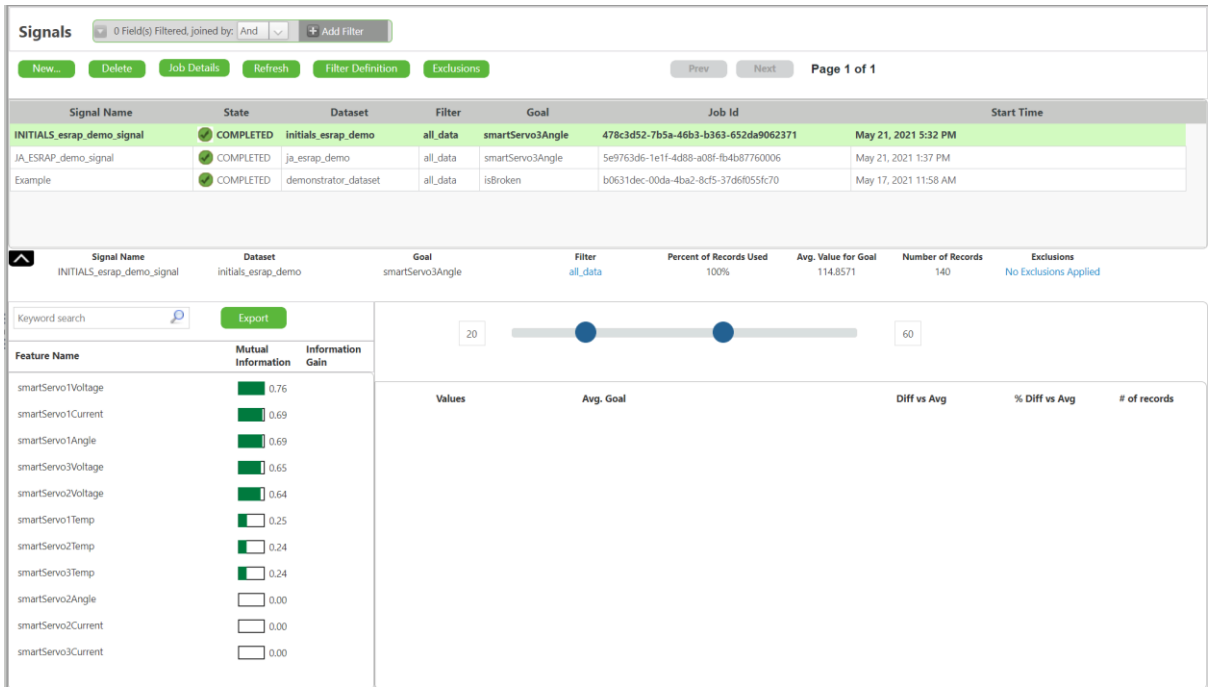
7. Navigate to Signals
    a. Create a new signal



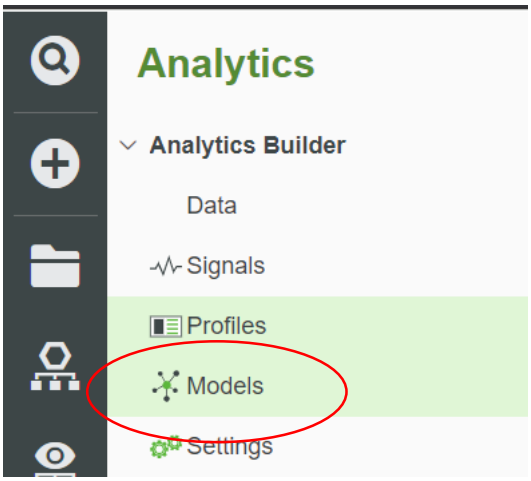
8. Enter Signal settings
    a. Name: INITIALS_esrap_demo_signal
    b. Dataset: Select your previously created Dataset
    c. Goal: Select SmartServo3Angle
    d. Filter: all_data
    e. -> Submit



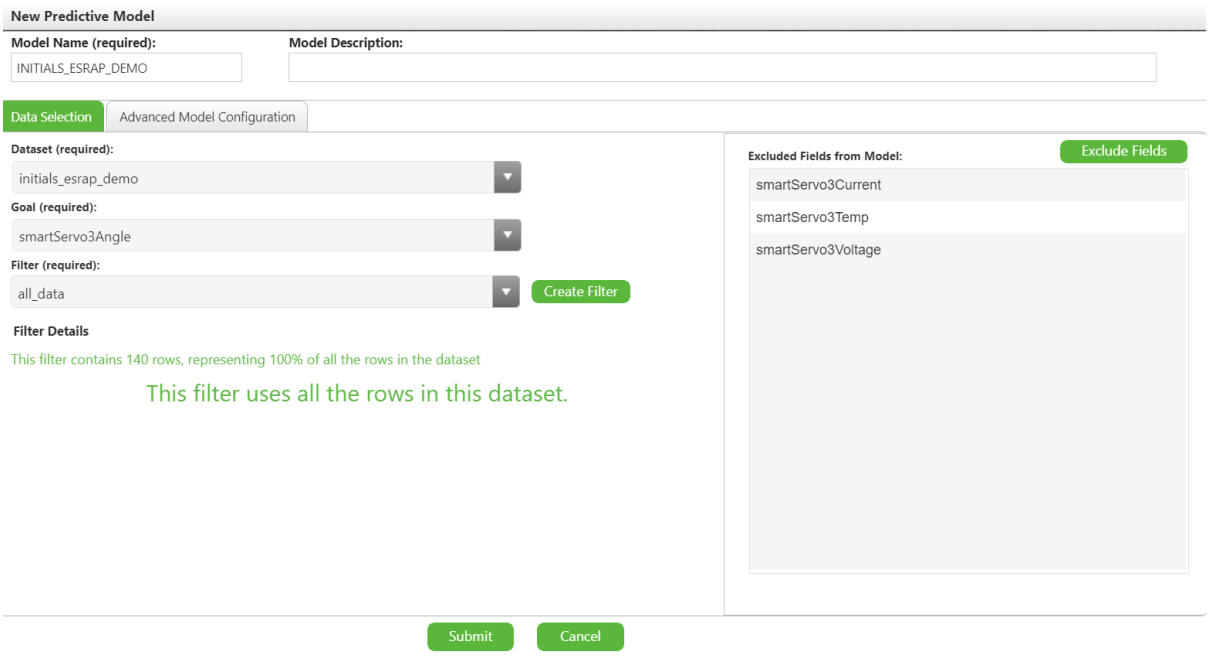
9. Review Signal results
    a. On the left side you can see the Feature Name and how much information it provides for the value of SmartServo3Angle (from 0 to 1), 0 means no mutual information



10. Navigate to Models



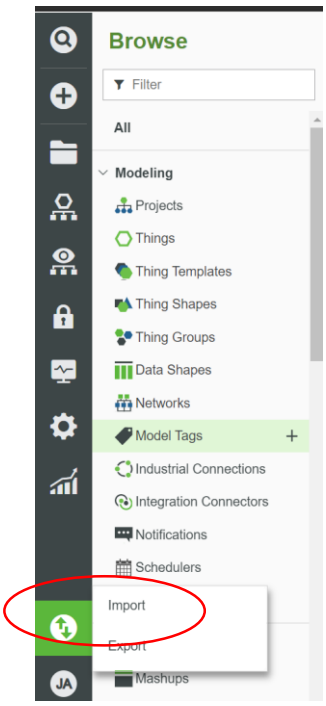
11. Create new Model
    a. Name: INITIALS_ESRAP_DEMO
    b. Dataset: name of the previously created dataset
    c. Goal: smartServo3Angle
    d. Exclude all Fields except: (right side of the window)
        i. smartServo1Current
        (if real hardware is used, you can also add other features with a high Mutual Information)
    e. Submit



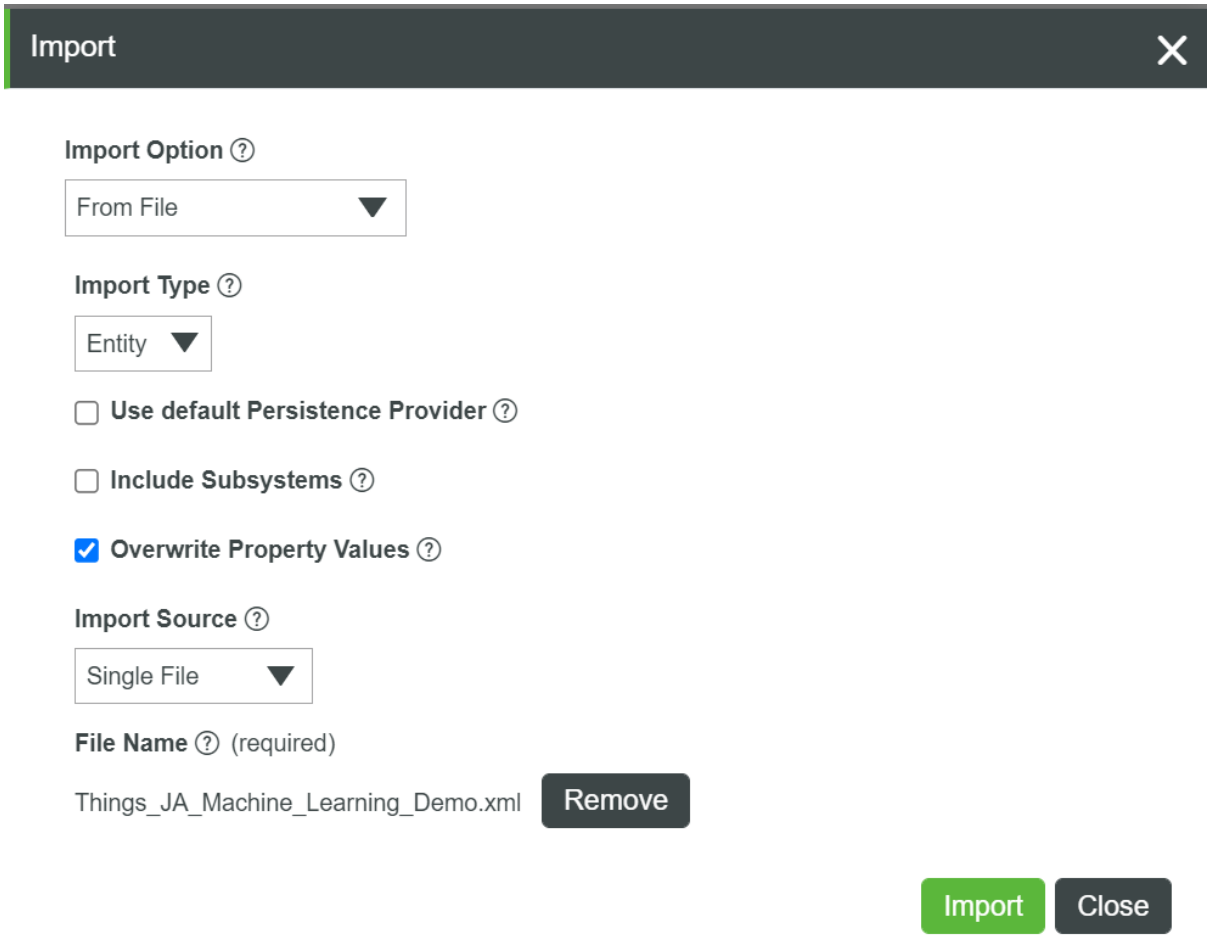
12. Wait until model state changes to Completed
13. Double click on created model

a. Copy Model Job Id

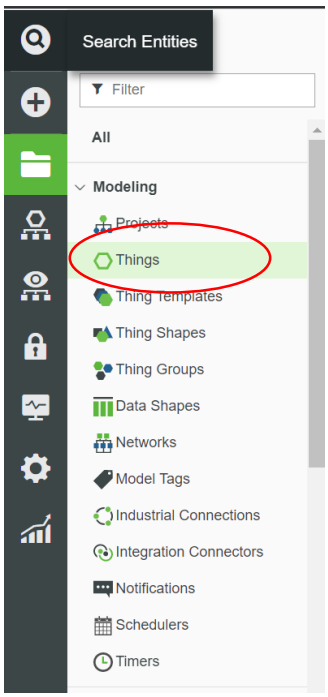
14. Import Things_JA_Machine_Learning_Demo.xml
    a. Navigate to import



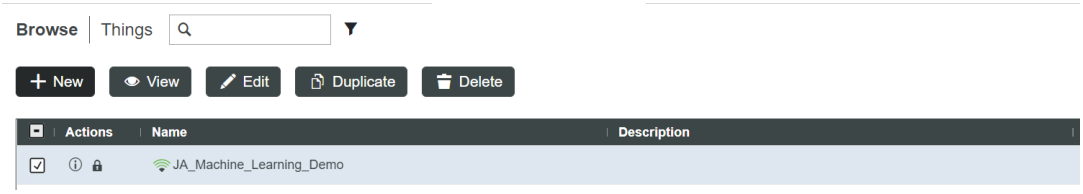
    b. Select Browse -> navigate to the Things_JA_Machine_Learning_Demo.xml file
    c. Compare your settings with the image below
    d. If the settings are the same, click on import



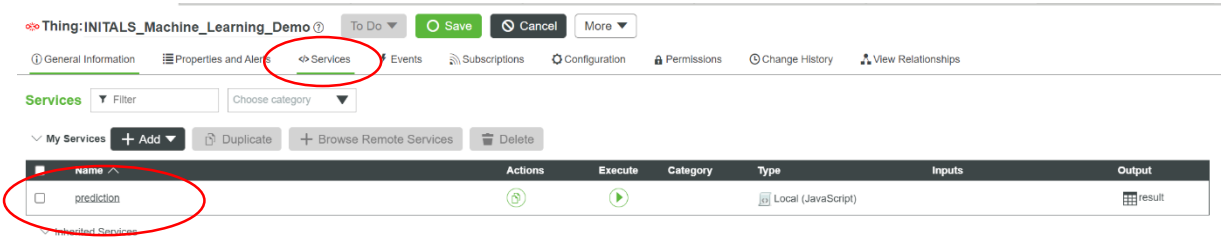
15. Navigate to Browse -> Things



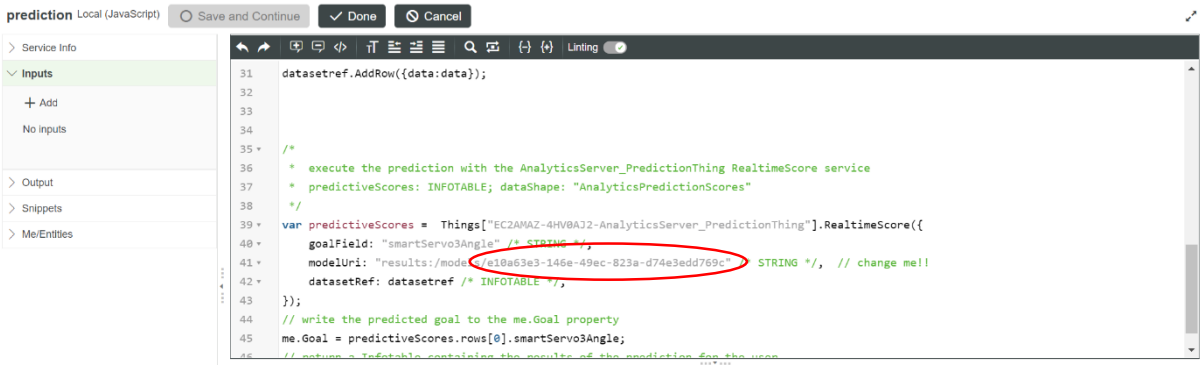
16. Select JA_Machine_Learning_Demo



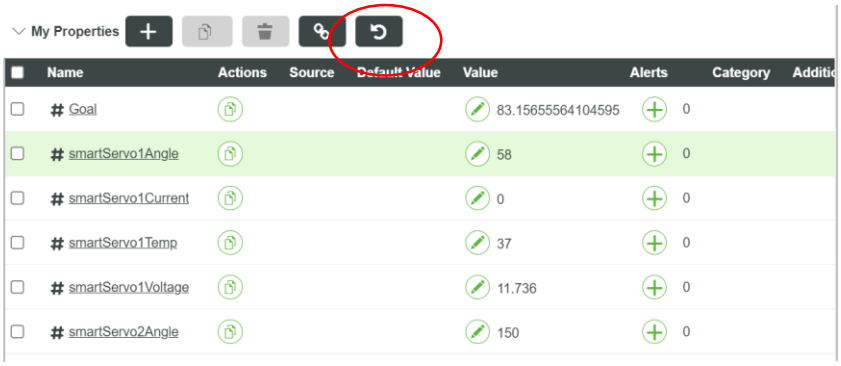
17. Navigate to Services and edit the prediction service:



18. On line 41 change the model URI to "results:/models/your URI here", to find the URI go back to point 12 in the guide



19. Under the Subscriptions a subscriber checks if any of the property values changed. If a change occurs, the prediction service gets executed and a new value for the Goal property will be predicted.
    a. If you use the hardware you can run the ML_demo_live.py script
    Make sure to start the Arduino and connect the Arduino to the PC/Raspi before you execute the python script!
        i. Change the port in line 13 to the one you use
        ii. Change the URL in line 15 to the instance you are using
        iii. In line 16 insert your Thingworx Application Key (if you do not have one, create one in the Thingworx Composer: Browse->Application Key -> Add)
    b. If you do not use the hardware you can change the value of the “smartServo1Angle” property. Have a look at “ML_demo_dataset.csv” which angle values were used and try it out. After you updated the property value in Composer, you need to click the Refresh Button to see the new Goal value.


My Properties
Name
Actions
Source
Default Value
Value
Alerts
Category
Goal
83.15655564104595
0
smartServo1Angle
58
0
smartServo1Current
0
0
smartServo1Temp
37
0
smartServo1Voltage
11.736
0
smartServo2Angle
150
0