How to Operationalize a Time Series Model

About

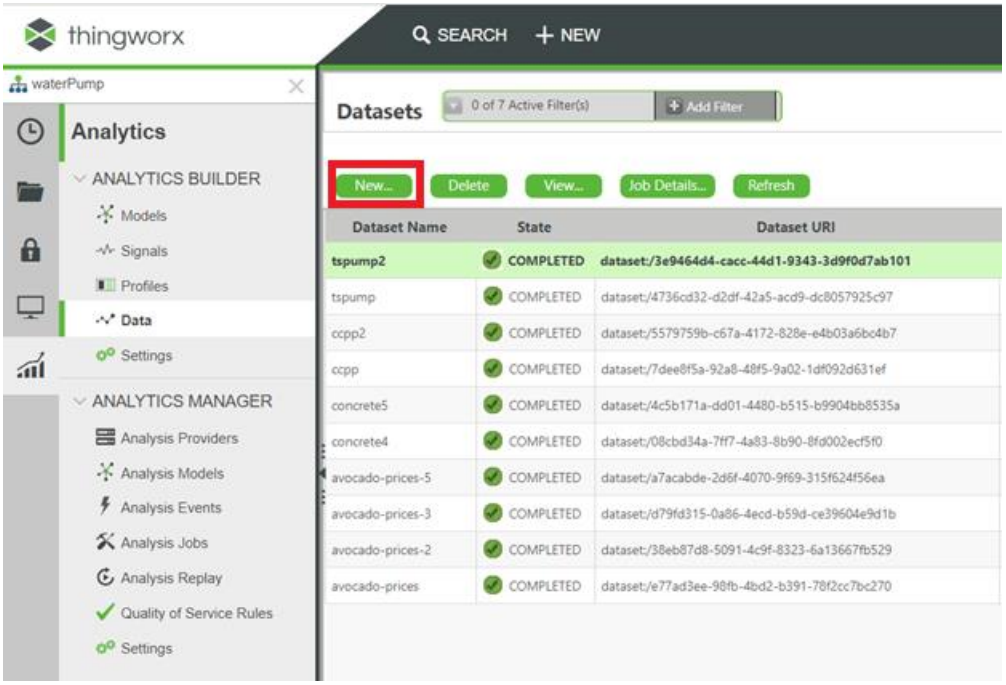
This document explains how to import a time series dataset, generate a prediction model on it, and then score some test records against that model. The main difference when scoring against a time series model is that multiple rows will need to be sent to the model for scoring. This is because you will need previous rows to lookback, and future rows to lookahead. This differs from point-in-time (i.e. non-time-series) models which are only able to score 1 record at a time.

Prerequisites

- Good technical & scripting knowledge of Thingworx (TWX) and Thingworx Analytics (TWA)
- Access to a combined TWX / TWA environment (tested with version 8.3.2)
- Water Pump resource files called waterPump-Resources.zip

Create the Dataset

To upload the dataset, go to Analytics Builder > Analytics Manager > Data > Datasets > New as shown below:



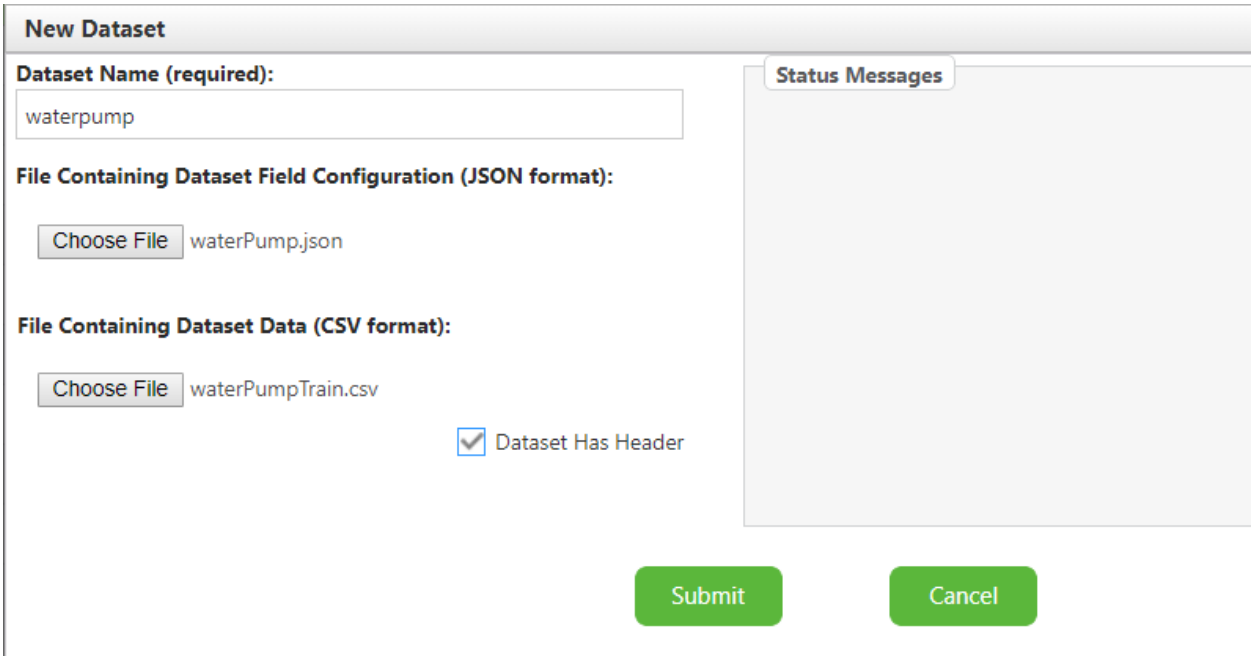
Now input a name for the Dataset.

Set JSON file = waterPump.json

Set CSV file = waterPumpTrain.csv

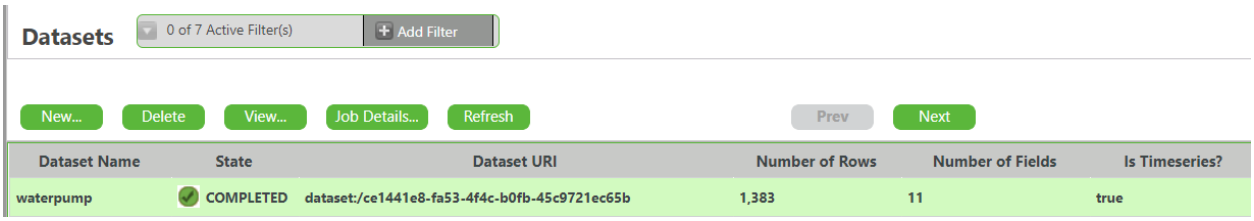
Dataset Has Header = TRUE

The referenced files are contained in the zipfile in the Prerequisites section above. See also the screenshot below:



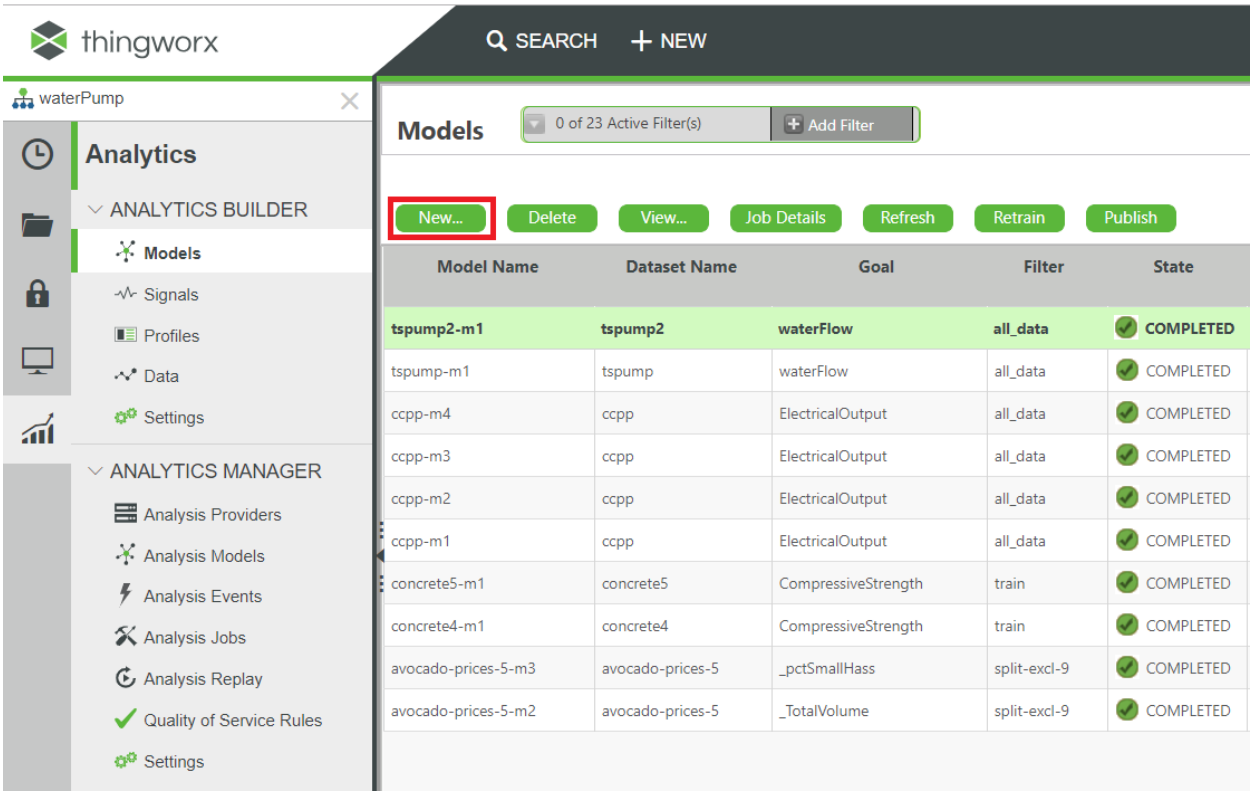
Click on Submit.

You should get a message showing that the upload has completed, and that the dataset is of type TimeSeries:



Create the Model

Go to Analytics > Analytics Builder > Models New, see also the screenshot below:



Set …

Model Name = waterpump-m1 (or any other name, excluding whitespace or special characters)

Dataset = waterpump (or the name you used in the previous section above)

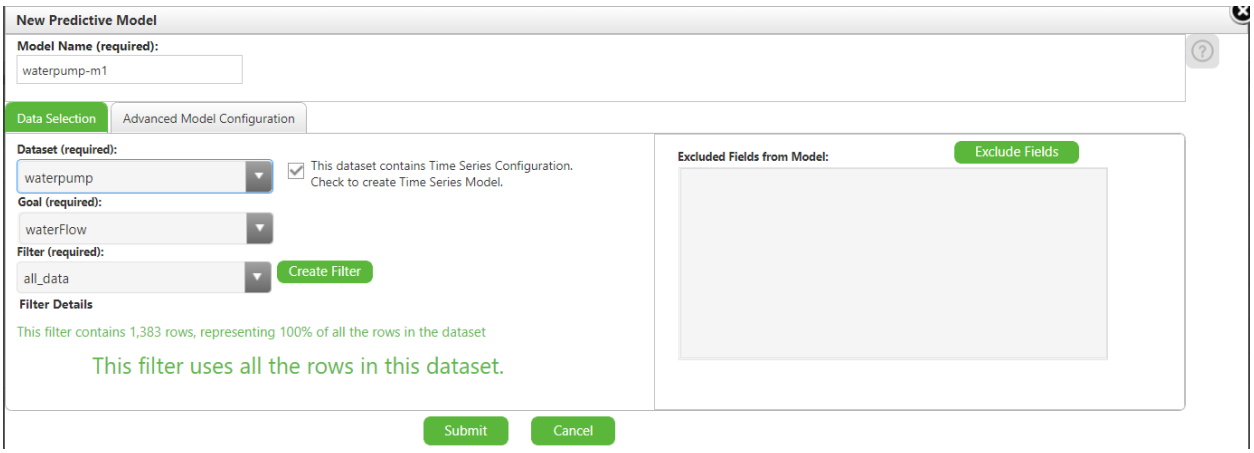
Goal = waterFlow

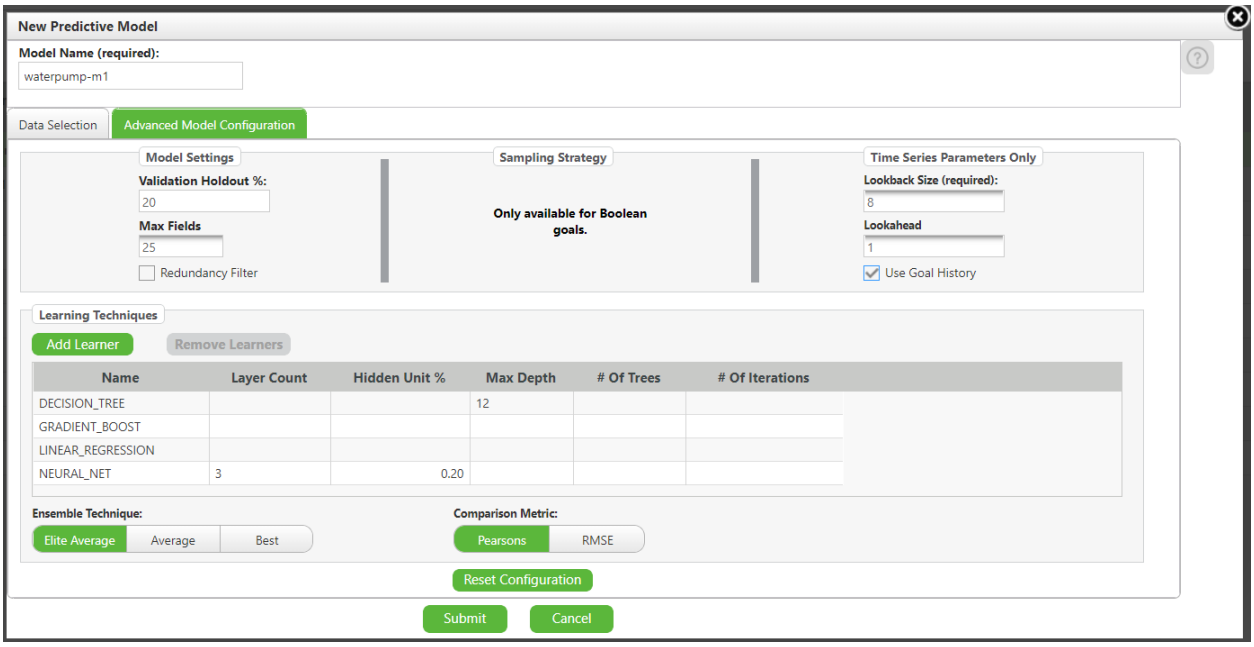
Filter = all_data

Excluded Fields = <none>

Advanced Model Configuration > Lookback Size = 8 (Note that if left at 0, TWA will automatically determine the optimal number of lookback rows. However for simplicity we are hard-coding the number of rows here. When we later score records, we will also provide exactly 8 rows).

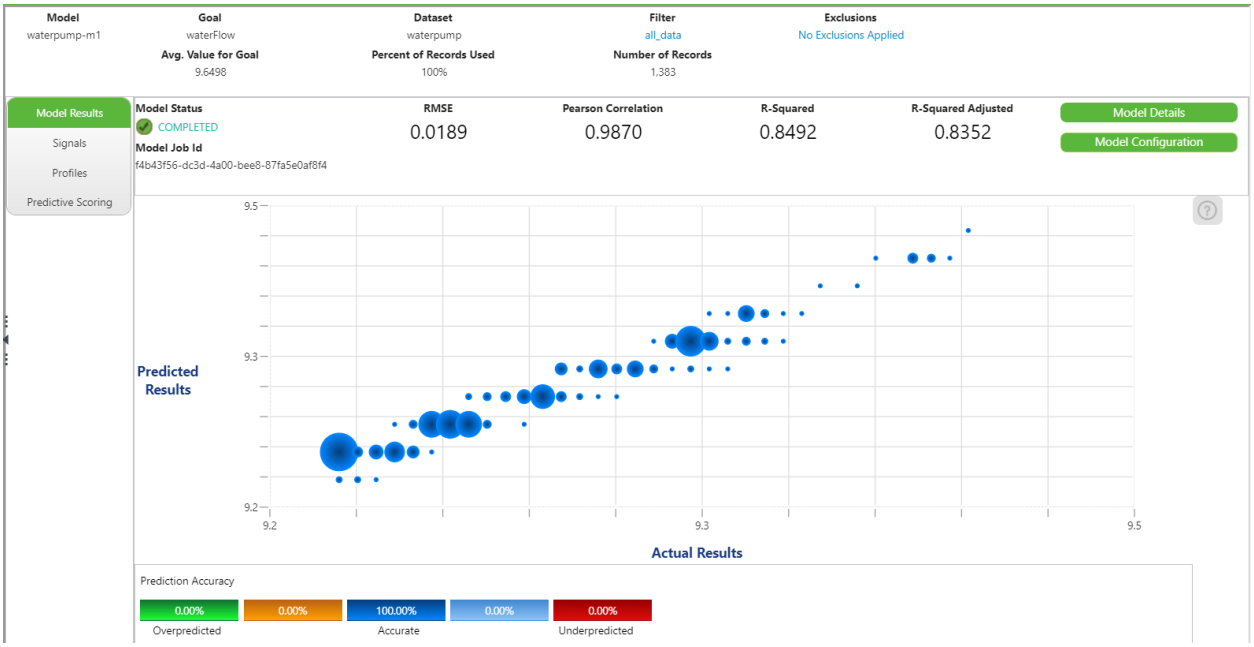
See also the screenshots below:



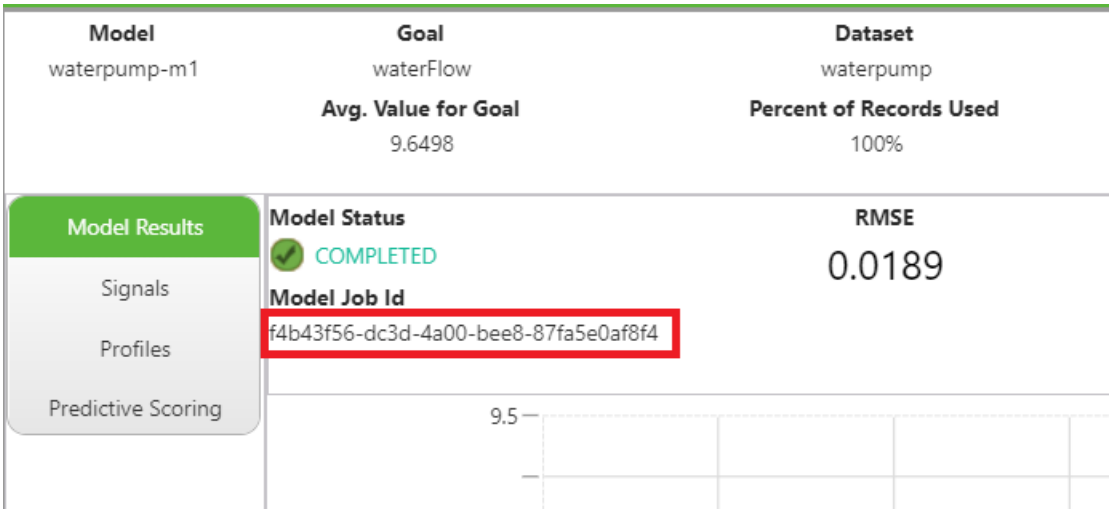


Click on Submit.

After around 1 minute (depending on your system performance) you should see that the model job has completed. You can click on the model and select View… to see some details as shown below:



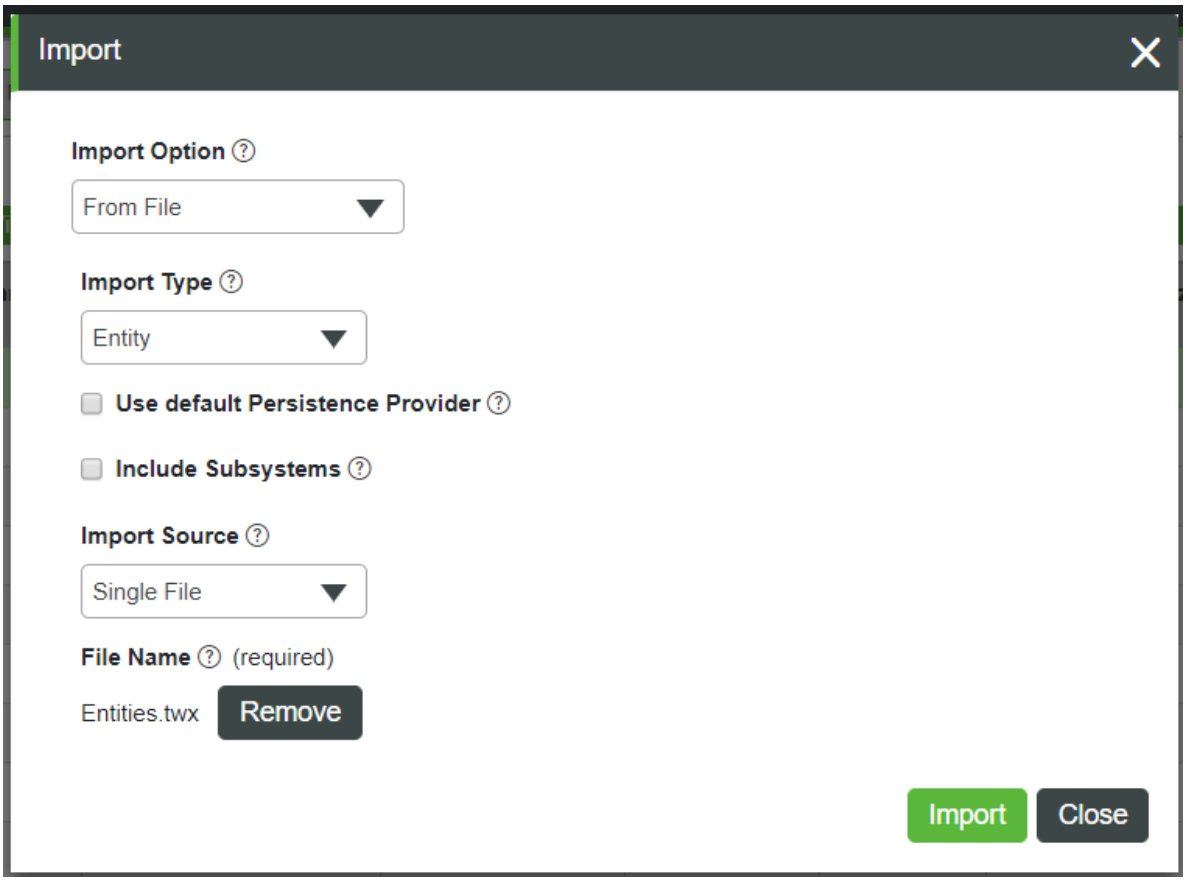
You can see the Model Job Id in the top left corner of the page as shown below:



Select and copy the Model Job Id.

Import the Entities

From the menu options on the top right hand corner of Thingworx Composer, select Import. Select to import from Single File Entities.twx as shown below:



Press Import. You should get a message that the import completed successfully, and you should be able to see the following new entities in your environment:

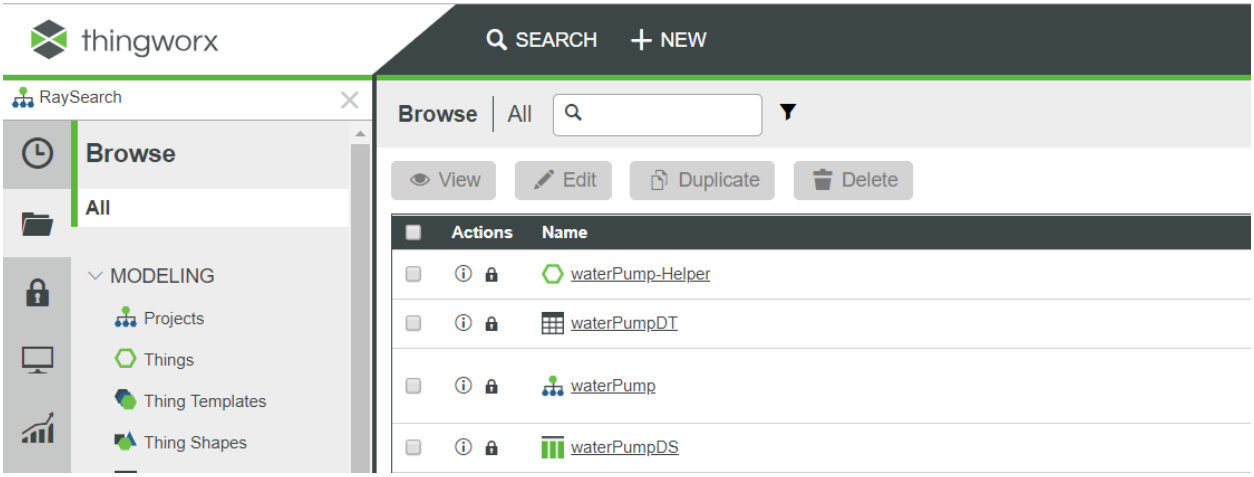
waterPump-Helper (Thing)

waterPumpDT (Data Table)

waterPump (Project)

waterPumpDS (Data Shape)

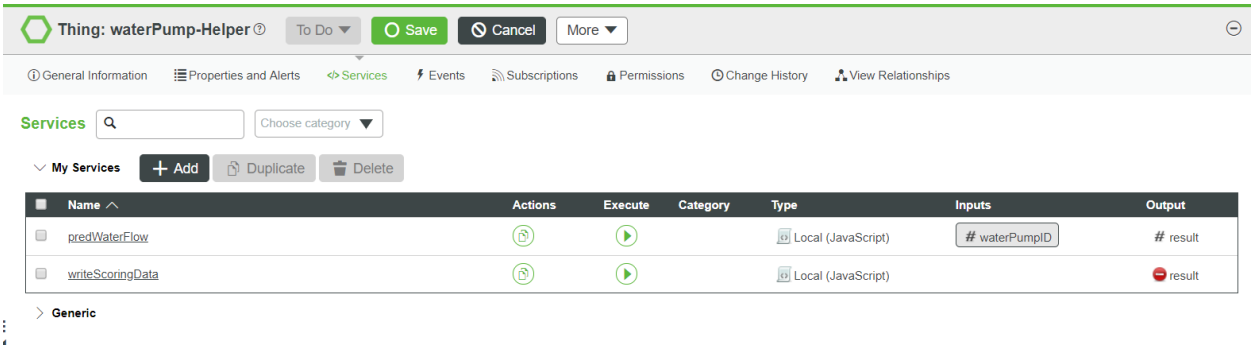
See also the screenshot below:



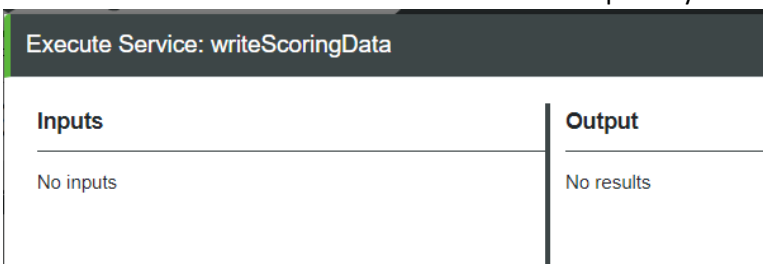
Write Scoring Data to Table

In this section we will execute a service that writes some test data to a Data Table called waterPumpDT. We will later score this test data using the model we created.

In Composer, open the services for Thing waterPump-Helper as shown below:



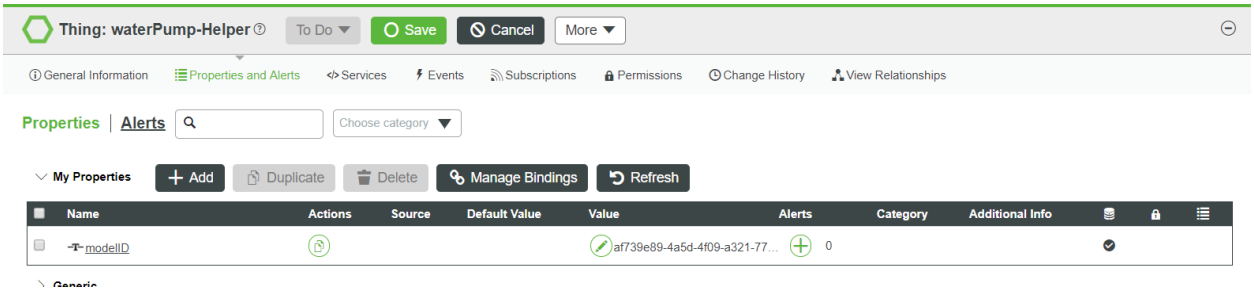
Now execute the service. Note that it does not output any results.



If you wish, you can verify that data was successfully written to Data Table waterPumpDT by executing service Things["waterPumpDT"].GetDataTableEntries with no input parameters. You should see that there are 80 data rows. These rows are grouped by waterPumpID. For each waterPumpID there is a time-series of data which can be sent to the scoring service.

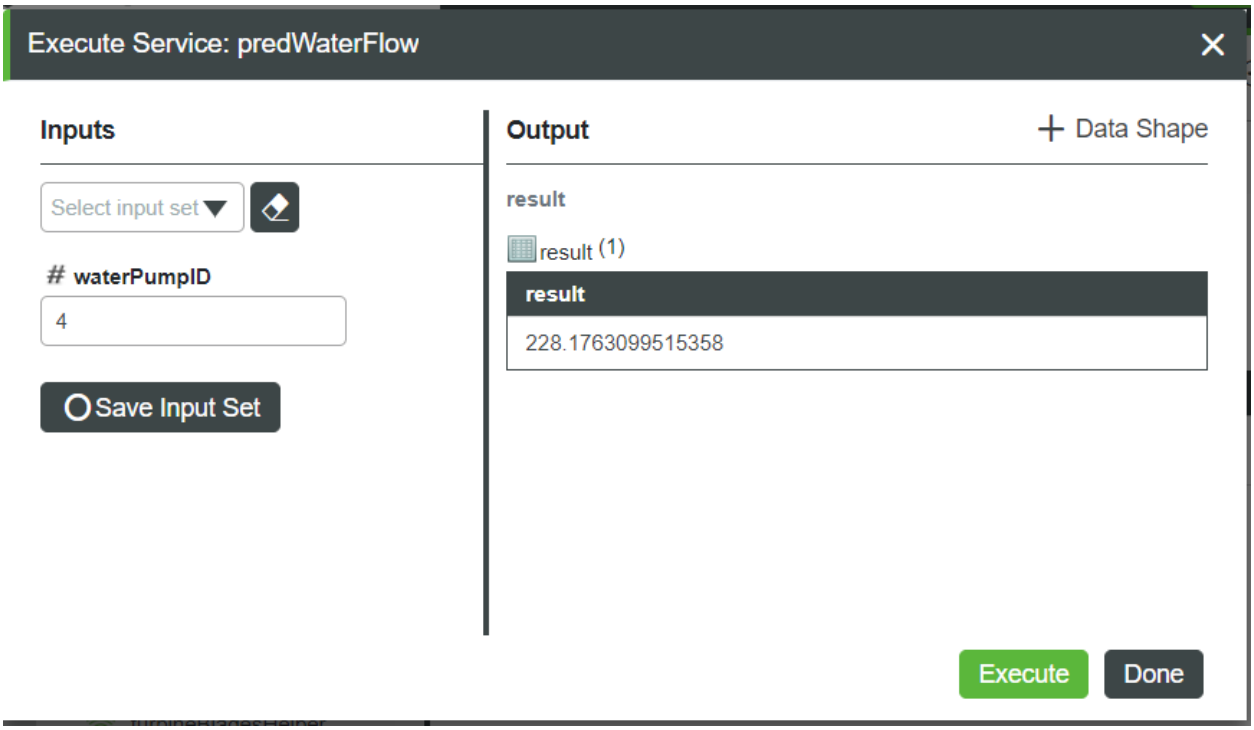
Storing the Model ID

Open Thing waterPumpHelper, Properties and Alerts. Set the value of property modelID to the value you noted in the section “Create the Model” above.

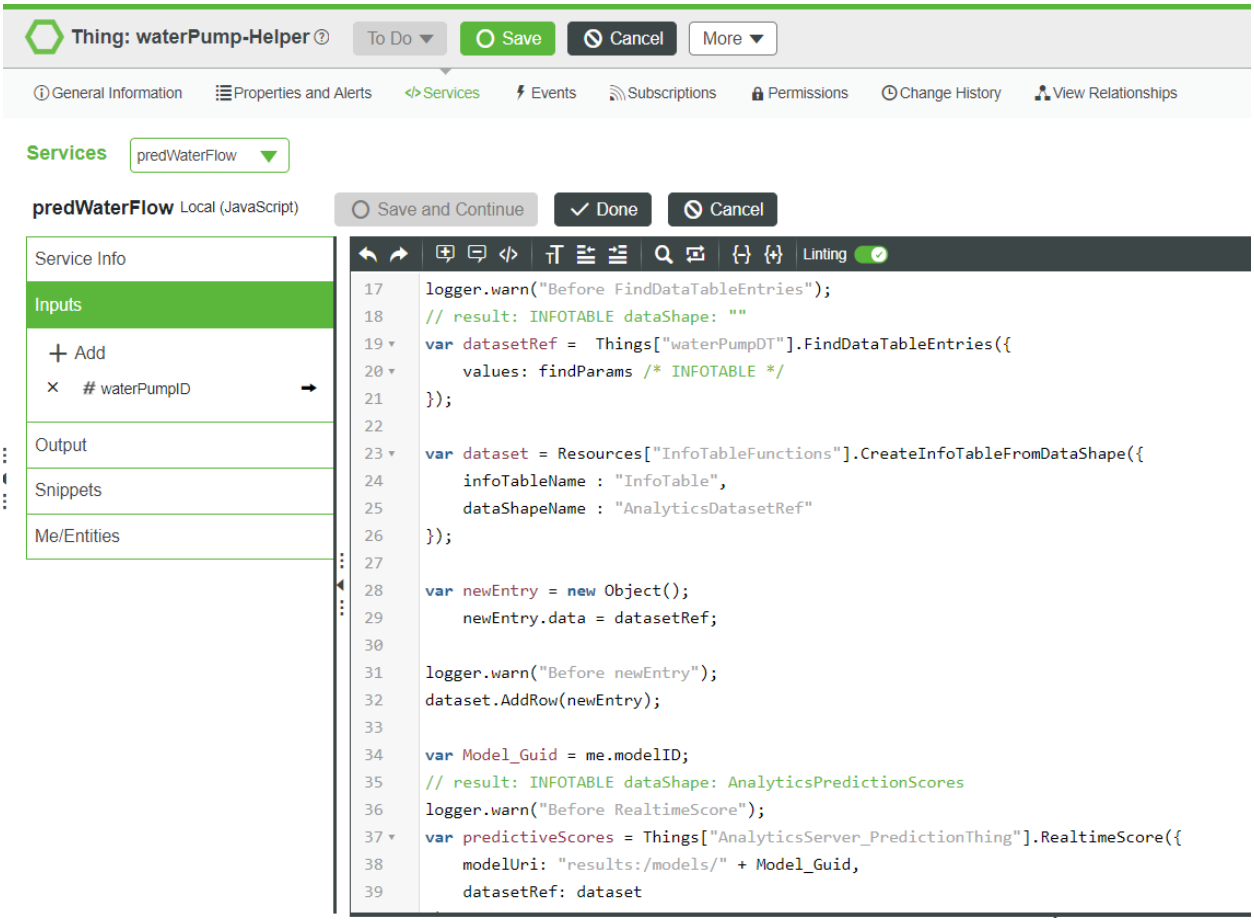


Scoring Against the Time Series Model

Open Thing waterPump-Helper, Services tab. Input a waterPumpID from 1 to 10 and click Execute. You should get a prediction of the water flow for the given waterPumpID based on the test time series data.



Now review the service code for predWaterFlow to understand how to execute method RealtimeScore on the PredictionThing (exact name depends on your system) for a time series model. In this example the method FindDataTableEntries on data table waterPumpDT is returning the 8 rows needed to pass through to the model in method RealtimeScore.



Conclusion

This document has walked through the steps for scoring a set of records against a time series analytics model to get a prediction result. The service code for Things[“waterPump-Helper”].predWaterFlow shows how the scoring is done based on the method RealtimeScore.