KARABÜK ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ

STAJ DEFTERİ

Ad Soyad : **Onur Saltaş**

Bölüm : **Bilgisayar Mühendisliği**

Staj Türü : **Endüstri Stajı I**

KARABÜK-2020

İŞ YAPRAĞI	Staj Tarihi:	Yapılan İş:	Sayfa No:	1
	01.07.2020	Dating and Distribution of Tasks	Yetkili İmza	

Firstly, Met with people at the office. They explained some concepts after They gave technical training set. The training kit was teaching the topic of thingworx. thingworx is a program used for iot applications. Internet of Things (IOT) , each assigned specifically to the identifiers (UID) that is associated with each other; It is a system consisting of computers, mechanical and digital machines, objects, animals or people and capable of transferring data from a human to human or human to computer without requiring interaction.Thingworx important for ınternet of things. The platform leading industrial innovation platform provides flexibility to rapidly build and implement IoT applications that bring your vision to life. The thingworx components are :

Connect : In the world of IoT application development, connectivity refers to the infrastructure and protocols which connect devices to the cloud or network .

Analyze : The AI and Machine Learning technologies used in ThingWorx Analytics automate much of the complex analytical processes involved in creating data-driven insights for your IIoT application.

Build : Build your industrial IoT application using ThingWorx's drag-and-drop GUI development environment, model-based development platform.

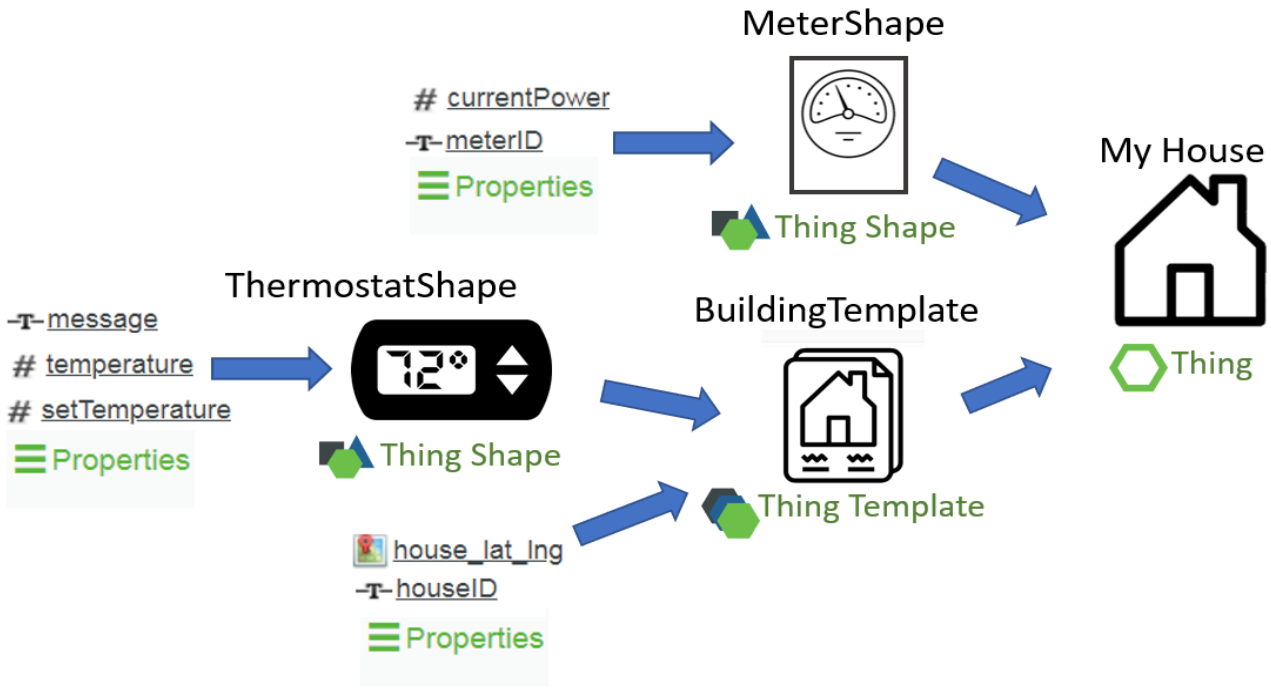
Manage : Efficiently manage your assets with visibility and control over your IoT solution. Install, configure and troubleshoot your application, while monitoring performance and communication with devices.

Experience : Use the industry's first purpose built IoT application development environment to design engaging experiences for web and mobile applications.

Secure : ThingWorx is secure by design and offers multiple authentication options to increase the security of your IoT application. From TLS-encrypted communication and role-based access controls to the distribution of security patches, ThingWorx integrates a range of security features that you can leverage in your development process.

Learned about the general steps of thingworx.This called data model also. Model-based design with reusable building blocks makes your applications scalable and flexible. A ThingWorx application is built from Things, each based on a Thing Template that defines the common Properties (characteristics) and Services (behaviors) for a set of entities. Once a Thing Template is created, you can easily instantiate multiple Things without duplicating effort.

In this tutorial, Develop an application for a house including a thermostat, an electrical meter, and a sensor data simulator. We will demonstrate how to capture, store, analyze, and visualize data using the ThingWorx Foundation Server.It will create Thing Shapes that model both a thermostat and an electric meter. It will then create a Thing Template that represents a house based on these shapes and other Properties.This is thingworx general model :



Before creating entities in the platform, recommend create Model Tags to organize your work so it can be easily exported and saved. Model tags are terms used to categorize and group items throughout the platform. Tags are also used for version-control, searching, and migrating entities between servers. Model Tags consist of a "vocabulary" and a "term" (Vocabulary:Term). For example, Color: Red or Department: Econmy.

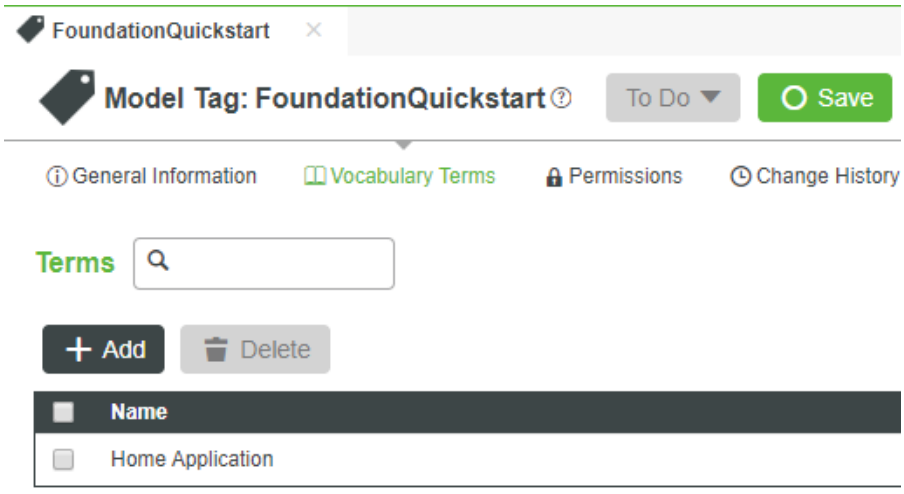
It is done step by step like this:

For vocebulary :

1) Start on the Browse, folder icon tab of ThingWorx Composer.
2) In the Modeling section of the left-hand navigation, hover your mouse over Model Tags and click the + button.
3) Enter whatever name into the Name text box after enter save.

For term :

1) Open the before said whatever name vocabulary and click Edit.
2) Click Vocabulary Terms.
3) Click the + Add button
4) Enter general name in the text box
5) Click ✓ button.
6) Click Save.



Thing Shapes are components that contain Properties and Services. In Java programming terms, they are similar to an interface.

For Thing Shapes Steps :

Start on the Browse, folder icon tab of ThingWorx Composer.

Under the Modeling section of the left-hand navigation panel hover over Thing Shapes, then click the + button.

Type whatever_in the Name field

In the Tags entry box, click the + and select the FoundationTutorial:After application tag.

Click Save.

Click Properties and Alerts tab at the top of your shape

Click + Add.

Enter the property name from the first row of the table below into the Name field of the Thing Shape

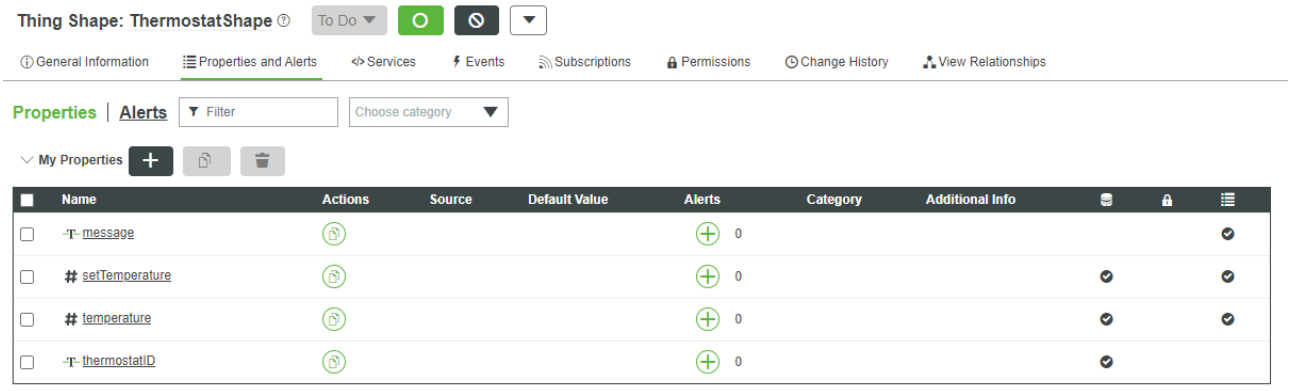
Select the appropriate Base Type from the drop-down menu

Check Persistent and/or Logged according to the table.

Click the done ✓ Button. You'll see that these Properties have been created for the Thing Shape.

Note :

When Persistent is selected, the property value will be retained during a system restart. Properties that are not persisted will be reset to the default during a system restart. When Logged is selected, every property value change will be automatically logged to a specified Value Stream.



İŞ YAPRAĞI	Staj Tarihi:	Yapılan İş:	Sayfa No:	5
	07.07.2020	Create Thing Template	Yetkili İmza	

You can create reusable building blocks called Thing Templates in ThingWorx to maintain scalability and flexibility of your application development. With Thing Templates you define a set of similar objects by specifying the Properties (characteristics) and Services (behaviors) that are common for all the objects.

In Java programming terms, a Thing Template is like an abstract class and can be created by extending other Thing Templates. Once a Thing Template is defined and saved in ThingWorx Foundation Server, you can replicate multiple Things to model a complete set without duplicating effort.

Steps for thing template :

Start on the Browse, folder icon tab on the far left of ThingWorx Composer

Under the Modeling section of the left-hand navigation panel, hover over Thing Templates and click the + button

Enter whatever in the Name field

In the Tags entry box, click the + and select the FoundationQuickstart:HomeApplication tag.

In the Base Thing Template box, click + to choose GenericThing as the Template

In the Implemented Shapes field, click the + to select the MeterShape Thing Shape.

After Properties features need to be added here. In this step, you will specify the Properties that represent the characteristics of a building. Some Properties like the building location may never change (static), while other properties like power and temperature information may change every few seconds

Steps for properties

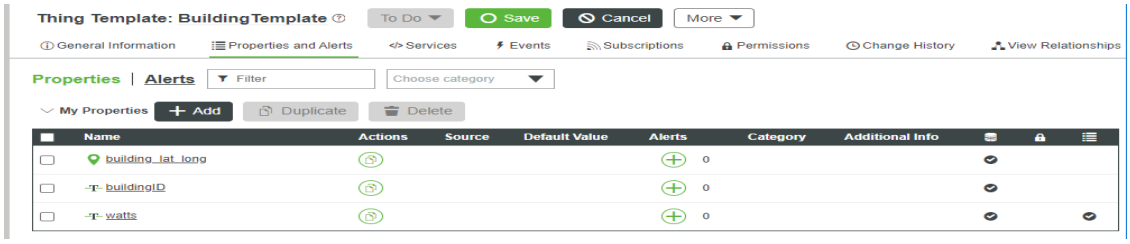
Select the Properties and Alerts tab under Thing Template: BuildingTemplate

Click the Edit button if the Template is not already open for editing, then click + Add next to My Properties.

Enter the property name in the Name field copied from a row of the table below,

Select the Base Type of the property from the drop down menu.

Check the appropriate Persistent and Logged check box



A Thing is used to digitally represent a specific component of your application in ThingWorx. In Java programming terms, a Thing is similar to an instance of a class.

In this step, you will create a Thing that represents an individual house using the Thing Template we created in the previous day. Using a Thing Template allows you to increase development velocity by creating multiple Things without re-entering the same information each time.

Start on the Browse, folder icon tab on the far left of ThingWorx
Composer.foundation_QS_addThing

Under the Modeling tab, hover over Things then click the + button.

Type MyHouse in the Name field.

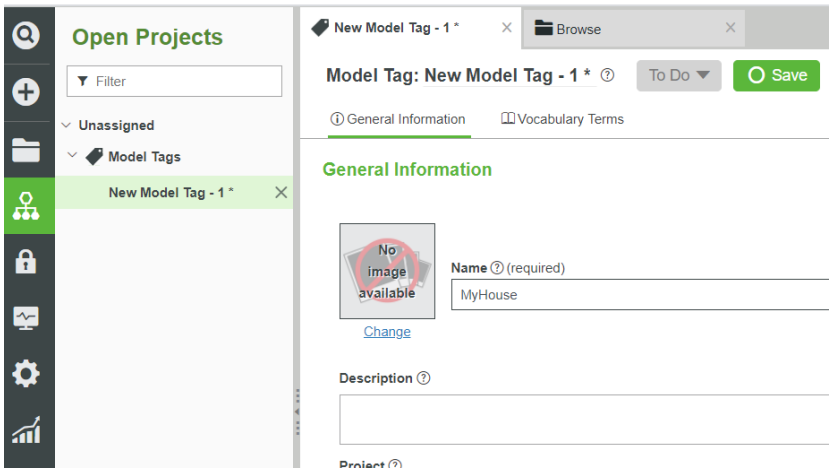
NOTE: This name, with matching capitalization, is required for the data simulator which will be imported in a later step.

In the Base Thing Template text box, click the + and select the recently created BuildingTemplate.foundation_QS_addThingInfo

In the Implemented Shapes text box, click the + and select the recently created ThermostatShape.

In the Tags text box, click the wand icon and select the FoundationQuickstart:HomeApplication tag.

Click Save.



you need to create a storage entity to record changing property values. This guide shows ways to store data in ThingWorx Foundation. This exercise uses a Value Stream which is a quick and easy way to save time-series data.

Steps for value stream :

Start on the Browse, folder icon tab on the far left of ThingWorx Composer.

Under the Data Storage section of the left-hand navigation panel, hover over Value Streams and click the + button.

Select the ValueStream template option, then click OK.

Enter Foundation_Quickstart_ValueStream in the Name field.

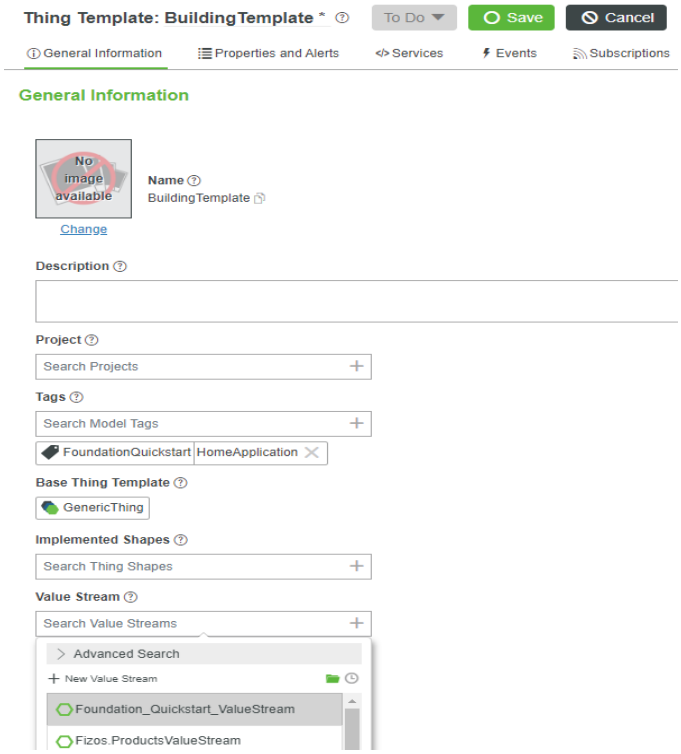
In the Tags field, click the + and select the FoundationQuickstart:Home Application tag.

Steps for Update Thing Template :

Navigate to the BuildingTemplate Thing Template.

Confirm you are on the General Information tab.

Click Edit then, in the Value Stream text entry box, click the + then select Foundation_Quickstart_ValueStream.



The ThingWorx Foundation server provides the ability to create and execute custom Services written in Javascript. Expedite your development with sample code snippets, code-completion, and linting in the Services editor for Things, Thing Templates, and Thing Shapes.foundationTutorial_ServiceOverview

In this section, you will create a custom Service in the Electric Meter Thing Shape that will calculate the current hourly cost of electricity based on both the simulated live data, and the electricity rate saved in your model. You will create a JavaScript that multiplies the current meter reading by the cost per hour and stores it in a property that tracks the current cost.

Steps for service :

Click Thing Shapes under the Modeling tab on the left navigation pane; then click on MeterShape in the list.

Click Services tab, then click + Add and select Local(Javascript).

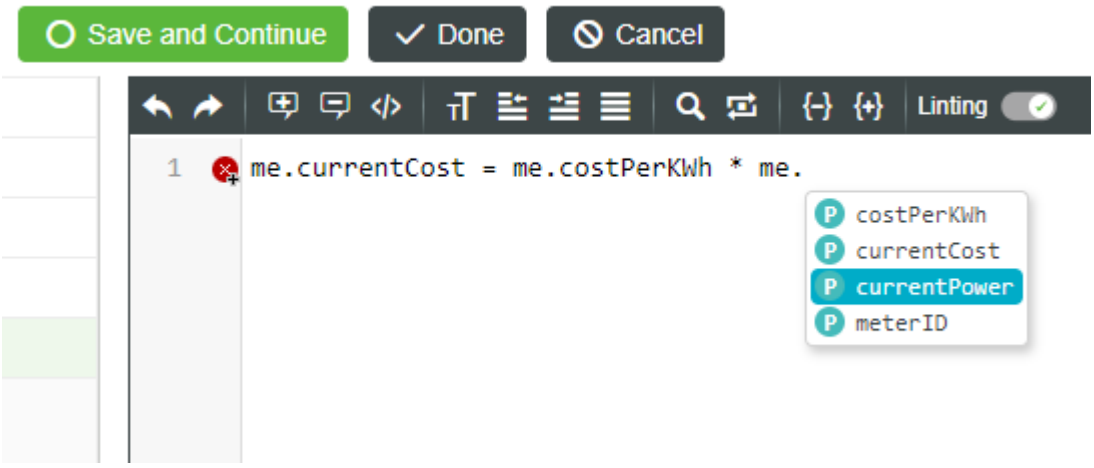
Type calculateCost into the Name field

Click Me/Entities to open the tab.

Click Properties.

This will add the Javascript code to the script box for accessing the currentCost property.

Reproduce the code below by typing in the script box or clicking on the other required properties under the Me tab:



An Event is a custom-defined message published by a Thing, usually when the value of a property changes. A Subscription listens for a specific Event, then executes Javascript code.

For alert steps :

Create an Alert that will be sent when the temperature property falls below 32 degrees.

Click Thing Shapes under the Modeling tab in Composer, then open the ThermostatShape Thing Shape from the list.

Click the temperature property.

Click the green Edit button if not already in edit mode, then click the + in the Alerts column.

Choose Below from the Alert Type drop down list.

Type freezeWarning in the Name field

Enter 32 in the Limit field. Keep all other default settings in place.

Click ✓ button above the new alert panel

For Subscription steps :

Create a Subscription to this event that uses Javascript to record an entry in the error log and update a status message.

Open the MyHouse Thing, then click Subscriptions tab

Type freezeWarningSubscription in the Name field.

After clicking the Inputs tab, click the the Event drop down list, then choose Alert.

In the Property field drop down, choose temperature

Click the Subscription Info tab, then check the Enabled checkbox

Follow the steps below to create code that sets the message property value and writes a Warning message to the ThingWorx log.

Enter the following JavaScript in the Script text box to the right to set the message property.

me.message = "Warning: Below Freezing";

Click the Snippets tab.

NOTE: Snippets provide many built-in code samples and functions you can use.

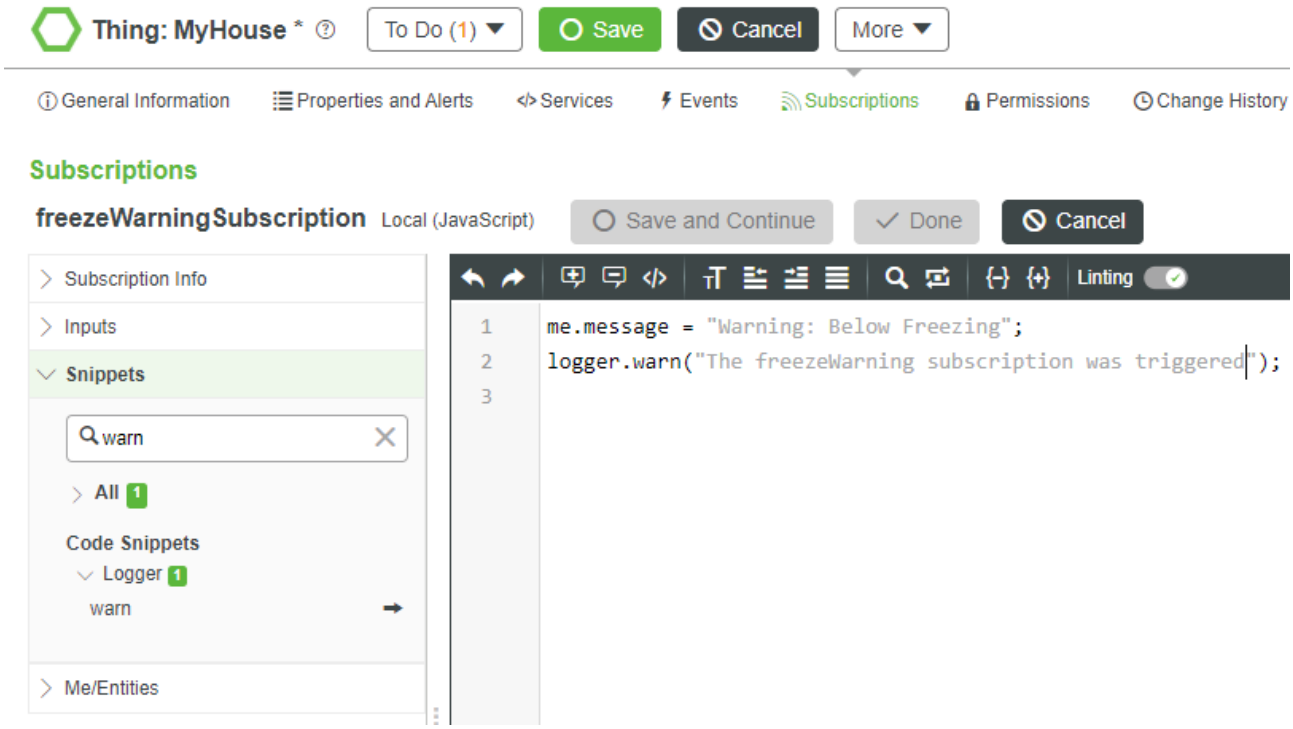
Click inside the Script text box and hit the Enter key to place the cursor on a new line.

Type warn into the snippets filter text box or scroll down to locate the warn Snippet.

Click the arrow and Javascript code will be added to the script window.

Add an error message in between the quotation marks.

logger.warn("The freezeWarning subscription was triggered");



ThingWorx you can create customized web applications that display and interact with data from multiple sources. These web applications are called Mashups and are created using the Mashup Builder.

The Mashup Builder is where you create your web application by dragging and dropping Widgets such as grids, charts, maps, buttons onto a canvas. All of the user interface elements in your application are Widgets.

We will build a web application with three Widgets: a map showing your house's location on an interactive map, a gauge displaying the current value of the watts property, and a graph showing the temperature property value trend over time.

Build Mashup :
Start on the Browse, folder icon tab of ThingWorx Composer.
Select Mashups in the left-hand navigation, then click + New to create a new Mashup.foundation_QS_newMashup
For Mashup Type select Responsive.select_static_mashup
Click OK.
Enter 'widgetMashup' in the Name text field, then click Savemashup_design
Select the Design tab to display Mashup Builder

Organize UI :
On the upper left side of the design workspace, in the Widget panel, be sure the Layout tab is selected, then click Add Bottom to split your UI into two halves.

Click in the bottom half to be sure it is selected before clicking Add Left
Click anywhere inside the lower left container, then scroll down in the Layout panel to select Fixed Size
Enter 200 in the Width text box that appears, then press Tab key of your computer to record your entry.
Click Save

İŞ YAPRAĞI	Staj Tarihi:	Yapılan İş:	Sayfa No:	12
	16.07.2020	Add Widgets	Yetkili İmza	

Click the Widgets tab on the top left of the Widget panel, then scroll down until you see the Gauge Widget

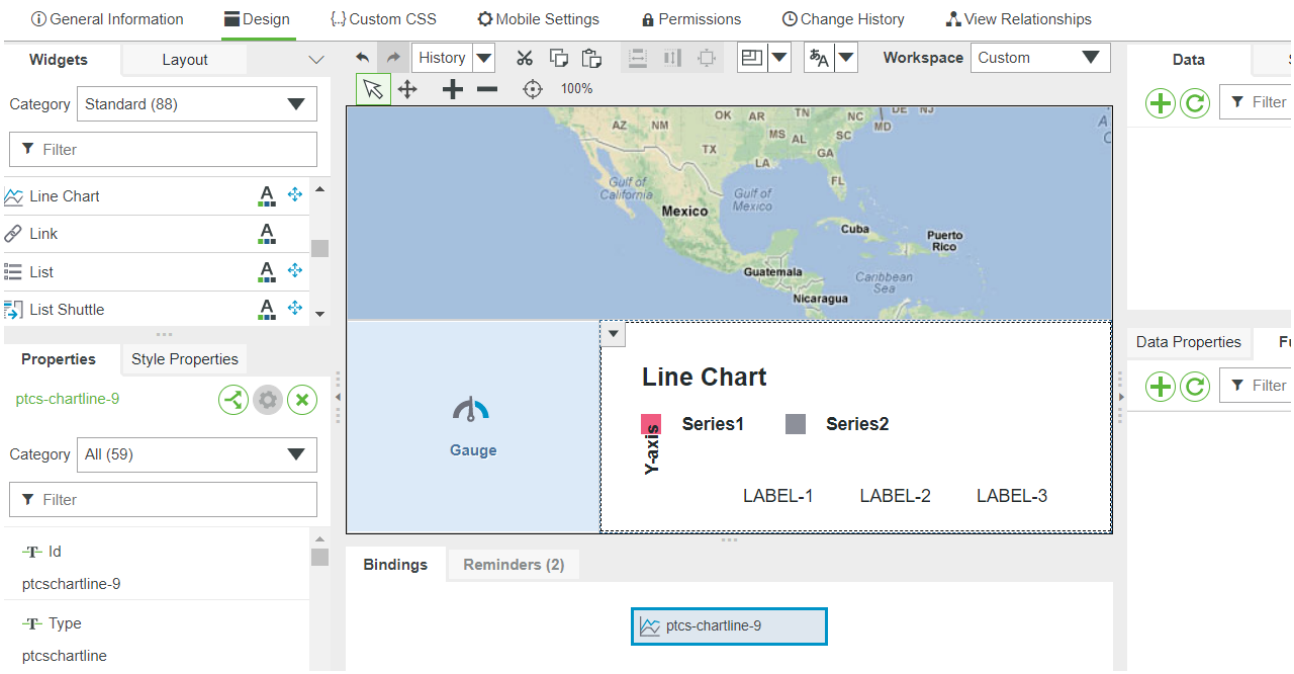
Drag the Gauge widget onto the lower left area of the canvas on the right. This Widget will show the simulated watts in use

Select the Gauge object on the canvas. The bottom left side of the screen contains the Widget properties. Scroll down to the Bindings section. Instead of binding the widget input to a dynamic source, enter Watts for the Legend property value and then press tab.

Click and drag the Google Map Widget onto the top area of the canvas

Click and drag the Line Chart Widget onto the lower right area of the canvas.

Click Save



Now that you have configured the visual part of your application, you need to bind the Widgets in your Mashup to a data source, and enable your application to display data from your connected devices.

Add Services to Mashup

Click Data in the top-right section of the Mashup Builder.

Click on the green + symbol in the Data tab

Type myhouse in the Search Entities textbox.

Click MyHouse.

In the Filter textbox below Services, type GetPropertyValues.

Click the arrow to the right of the GetPropertyValues service to add it.

Select the checkbox under Execute on Load.

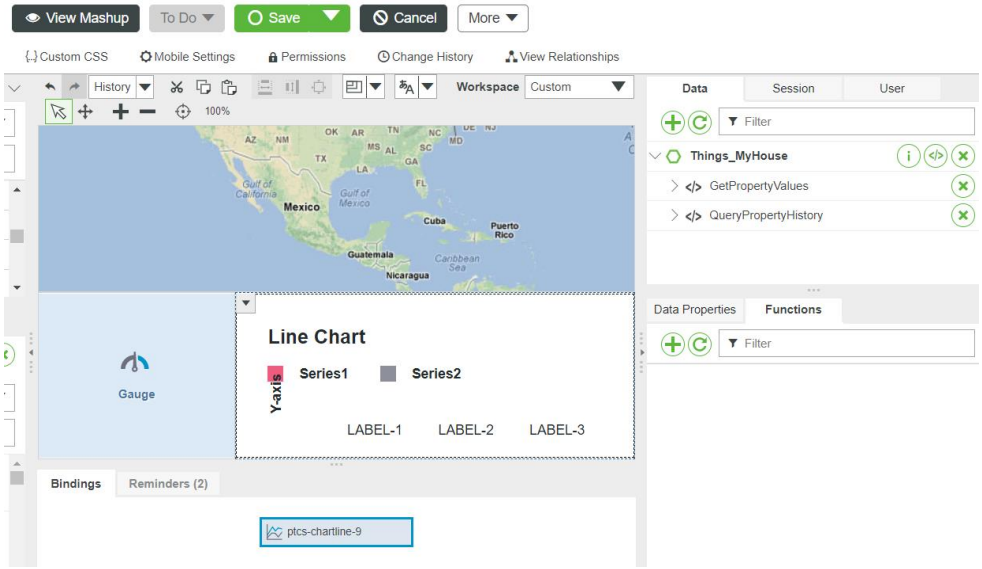
NOTE: If you check the Execute on Load option, the service will execute when the Mashup starts.

In the Filter textbox under Services, type QueryProperty.

Add the QueryPropertyHistory service by clicking the arrow to the right of the service name

Click the checkbox under Execute on Load.

Click Done



We will now connect the Services we added to the Widgets in the Mashup that will display their data.

Gauge :

Configure the Gauge to display the current power value.

Expand the GetPropertyValues Service as well as the Returned Data and All Data sections.

Drag and drop the watts property onto the Gauge Widget.bind_gauge_data

When the Select Binding Target dialogue box appears, select # Data.

Map :

Configure Google Maps to display the location of the home.

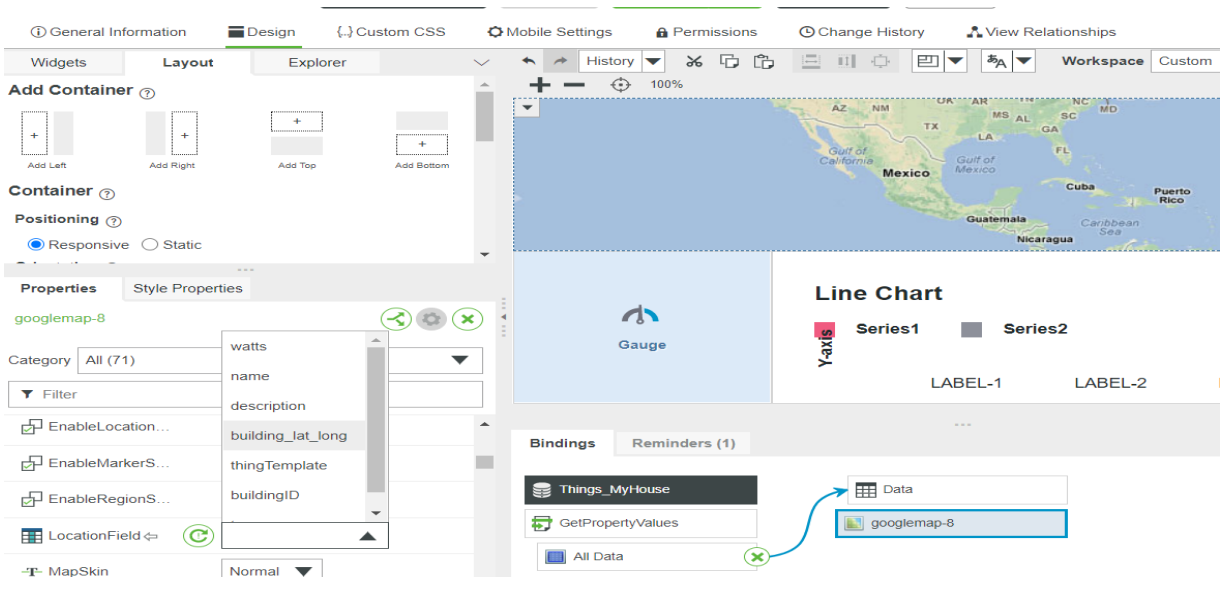
Expand the GetPropertyValues service as well as the Returned Data section.

Drag and drop All Data onto the map widget.

When the Select Binding Target dialogue box appears, select Data.

Selecting the Google Map Widget on the canvas displays properties that can configured in the lower left panel.

Set the LocationField property in the lower left panel by selecting building_lat_lng from the drop-down menu



Chart

Configure the Chart to display property values changing over time.

Expand the QueryPropertyHistory Service as well as the Returned Data section.

Drag and drop All Data onto the Line Chart Widget.

When the Select Binding Target dialogue box appears, select Data.

In the Property panel in the lower left, enter field in Filter Properties text box then select the property value from the drop-down according to the table below:

Property	Value
DataField1	temperature
XAxisField	timestamp

confirm_properties_bound_to_map

Verify Data Bindings

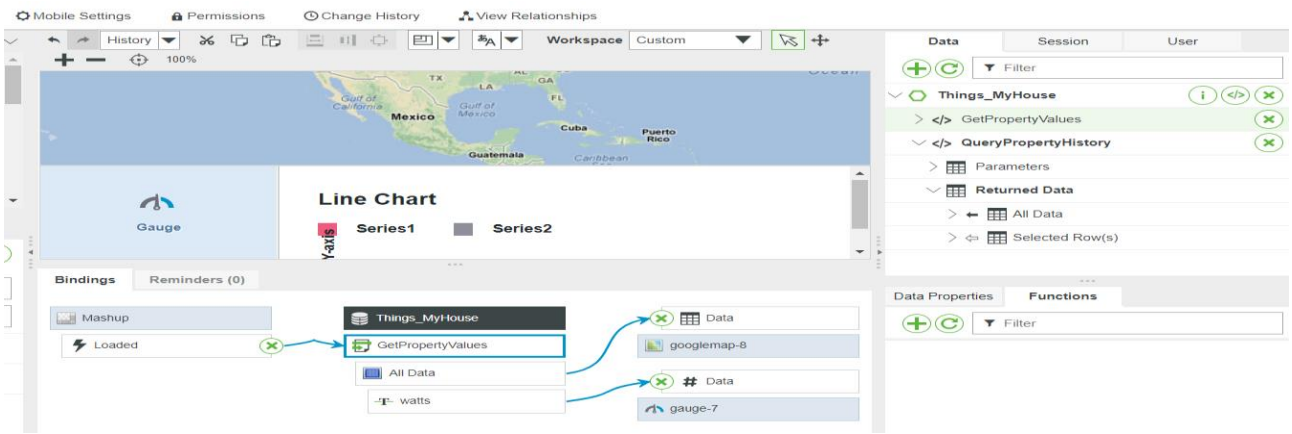
You can see the configuration of data sources bound to widgets displayed in the Connections pane.

Click on GetPropertyValues in the data source panel then check the diagram on the bottom of the screen to confirm a data source is bound to the Gauge and Map

widget.foundationTutorial_Mashup_Add_chartBinding

Click on the QueryPropertyHistory data source and confirm that the diagram shows the Time Series Chart is bound to it

Click Save.



At this point, you have created a Value Stream to store changing property value data and applied it to the BuildingTemplate. This guide does not include connecting edge devices and another guide covers choosing a connectivity method. We will import a pre-made Thing that creates simulated data to represent types of information from a connected home. The imported Thing uses Javascript code saved in a Subscription that updates the power and temperature properties of the MyHouse Thing every time it is triggered by its timer Event.

Import Data Simulation Entities

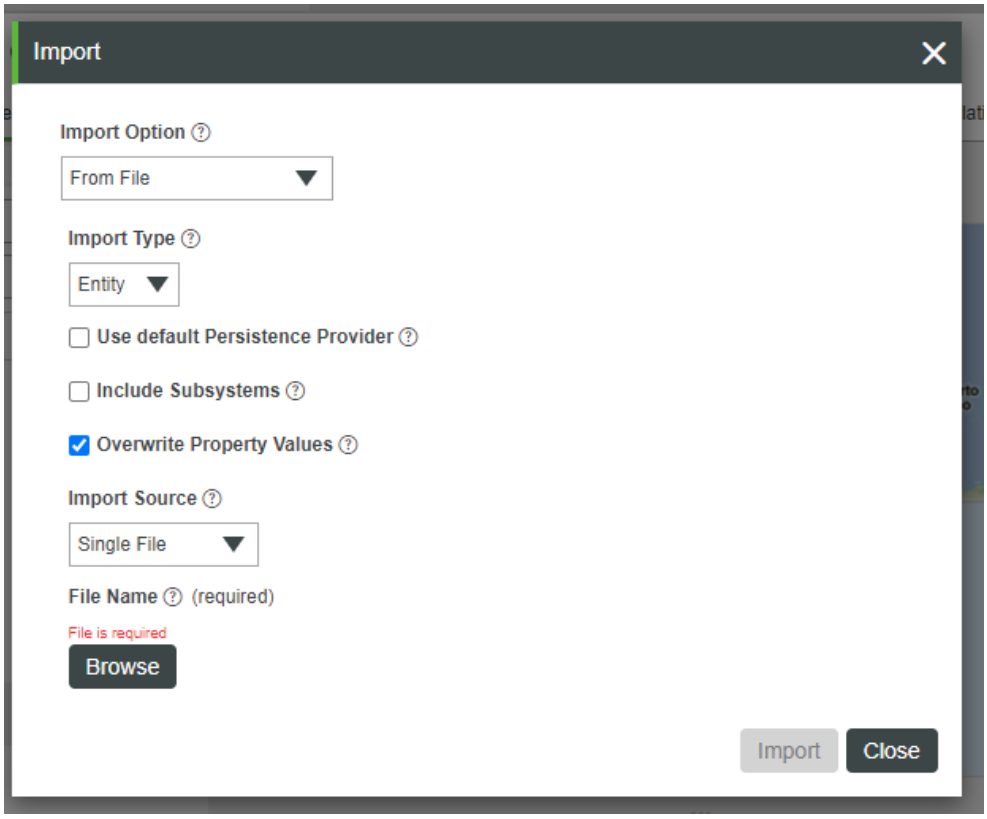
Download the sample by right-clicking MyHouse Simulator Thing and save to your computer by clicking 'Save link as'.

In Composer, click the Import/Export icon at the lower-left of the page.foundation_QS_uploadSimulator

Click Import.foundation_QS_importSimulator

Leave all default values and click Browse to select the Things_House_data_simulator.xml file that you just downloaded.

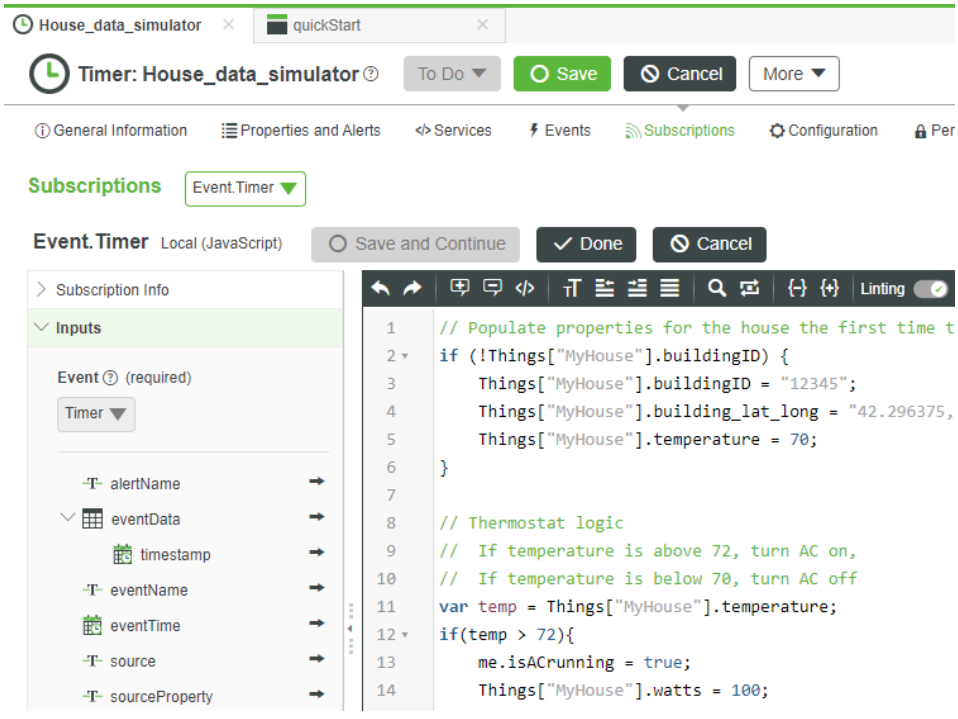
Click Open, then Import, and once you see the success message, click Close.



Navigate to the House_data_simulator Thing by using the search bar at the top of the screen.
Click the Subscriptions tab
Click Event.Timer under Name.
Select the Subscription Info tab

NOTE: Every 30 seconds, the timer event will trigger this subscription and the Javascript code in the Script panel will run. The running script updates the temperature and watts properties of the MyHouse Thing using logic based on both the temperature property from MyHouse and the isACrunning property of the simulator itself.

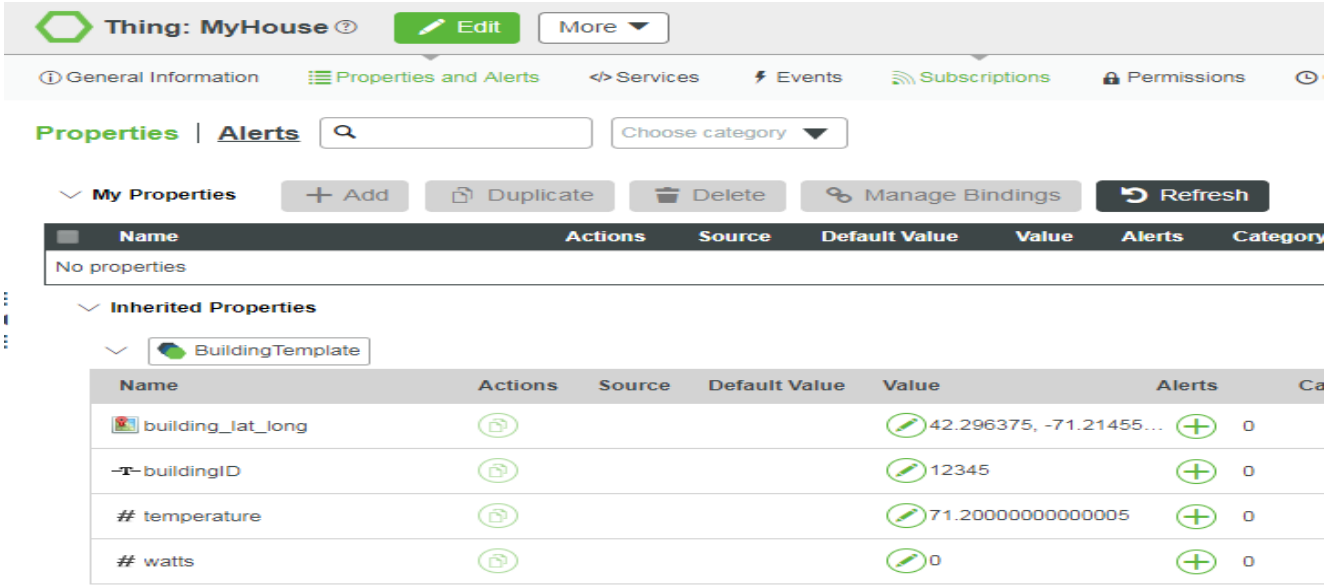
Expand the Subscription Info menu. The simulator will send data when the Enabled checkbox is checked.
Click Done then Save to save any changes.



Open the MyHouse Thing and click Properties an Alerts tab

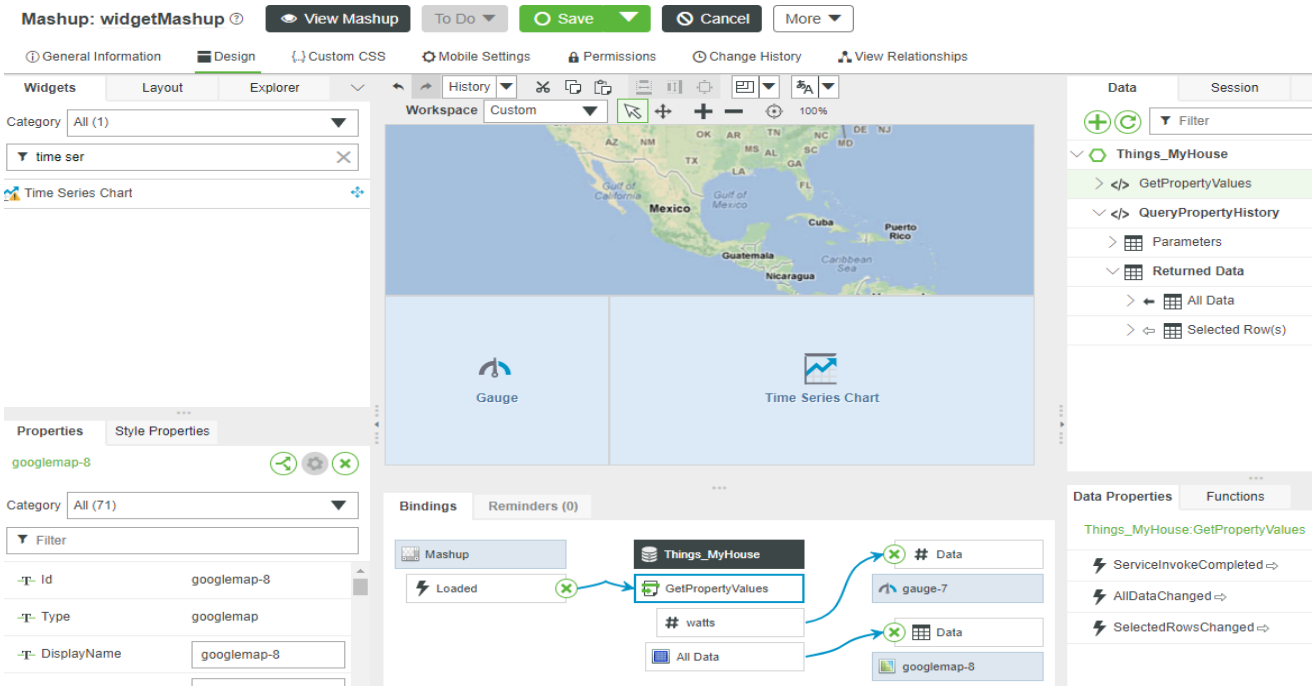
Click the Refresh button above where the current property values are shown

Notice that the temperature property value changes every 30 seconds when the triggered service runs. The watts property value is 100 when the temperature exceeds 72 to simulate an air conditioner turning on.



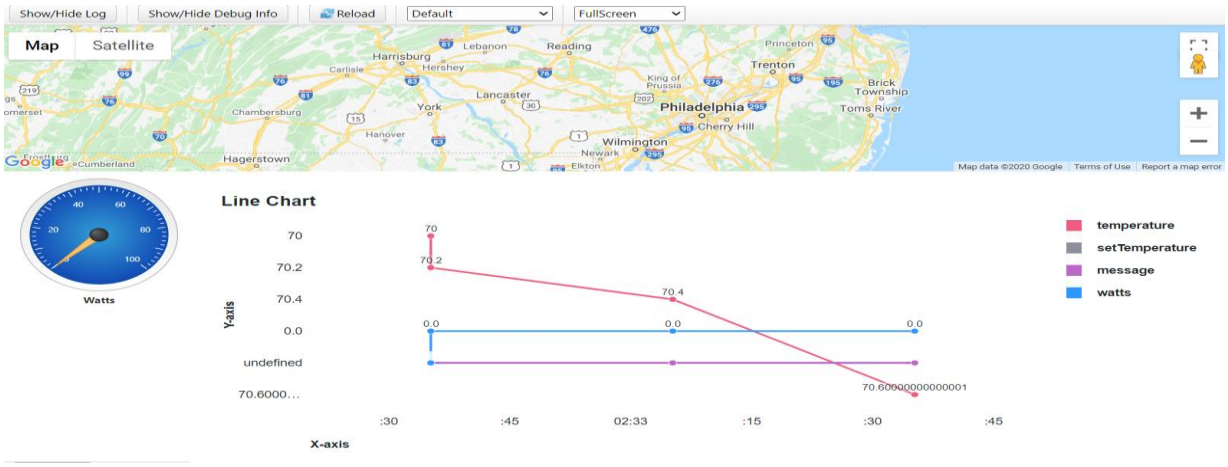
Test Application

Browse to your Mashup and click View Mashup to launch the application.



Browse to your Mashup and click View Mashup to launch the application

Confirm that data is being displayed in each of the sections



Test Alert

Open MyHouse Thing

Click the Properties and Subscriptions Tab.

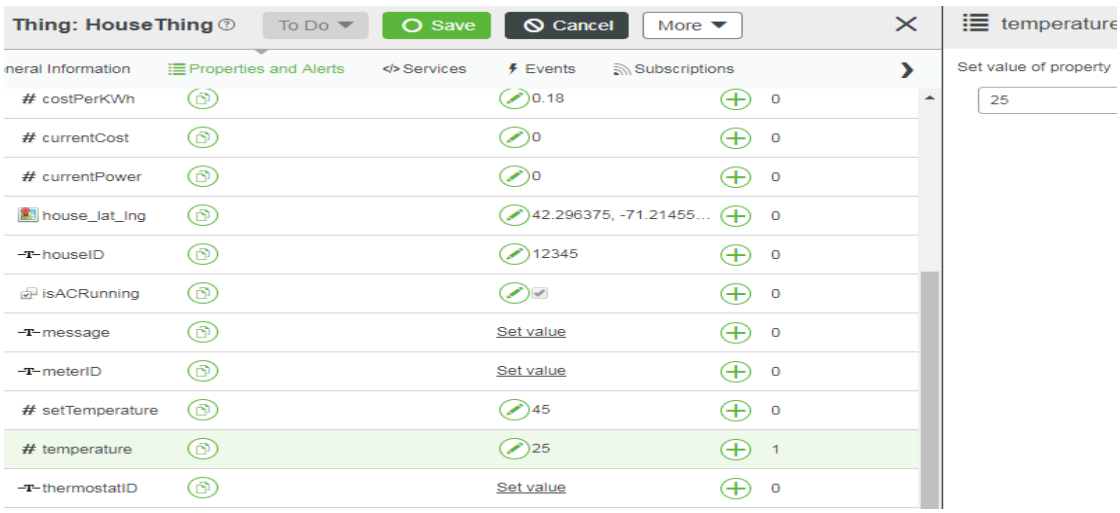
Find the temperature Property and click on pencil icon in the Value column.

Enter the temperature property of 29 in the upper right panel.

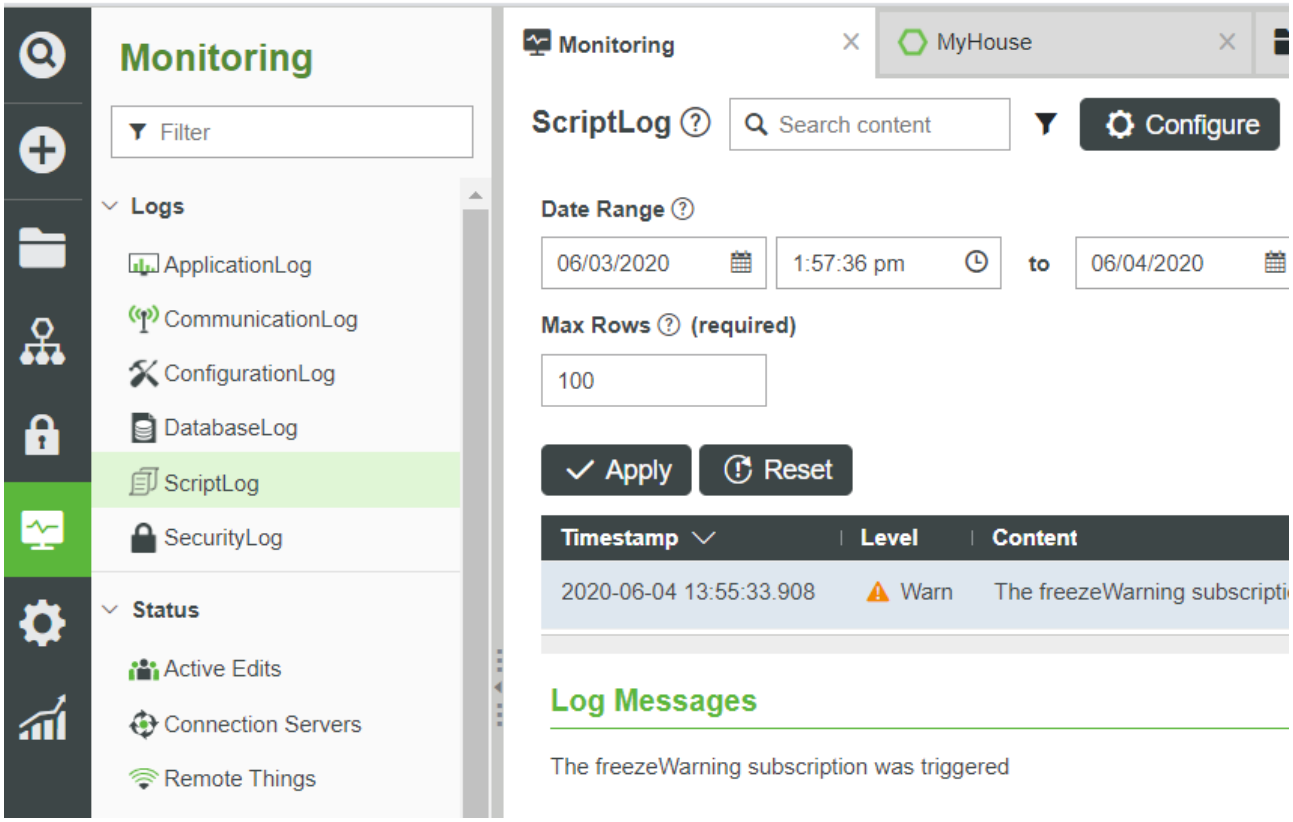
Click Check mark icon to save value. This will trigger the freezeWarning alert

Click Refresh to see the value of the message property automatically set

Click the the Monitoring icon on the left, then click ScriptLog to see your message written to the script log

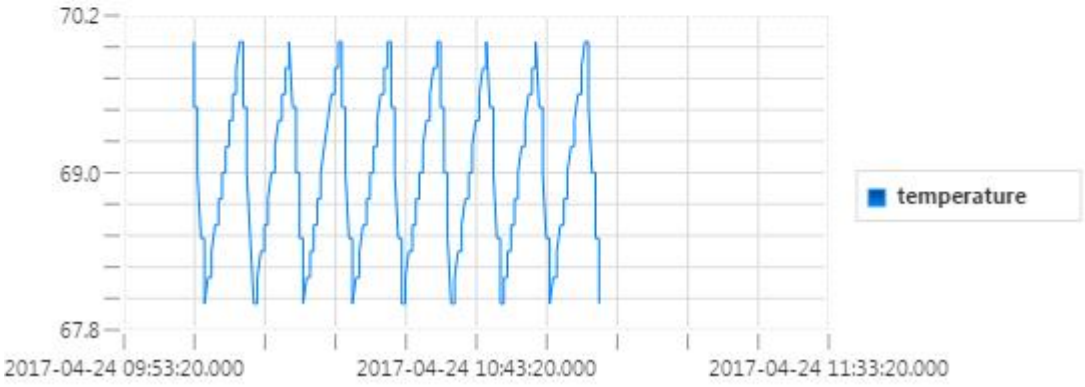


Click the the Monitoring icon on the left, then click ScriptLog to see your message written to the script log



Modify the application model, enhance your UI, and add features to the house monitoring application to simulate a request as it might come from an end user.

For this step, we do not provide explicit instructions, so you can use critical thinking to apply your skills. After completing the previous steps, your Mashup should look like



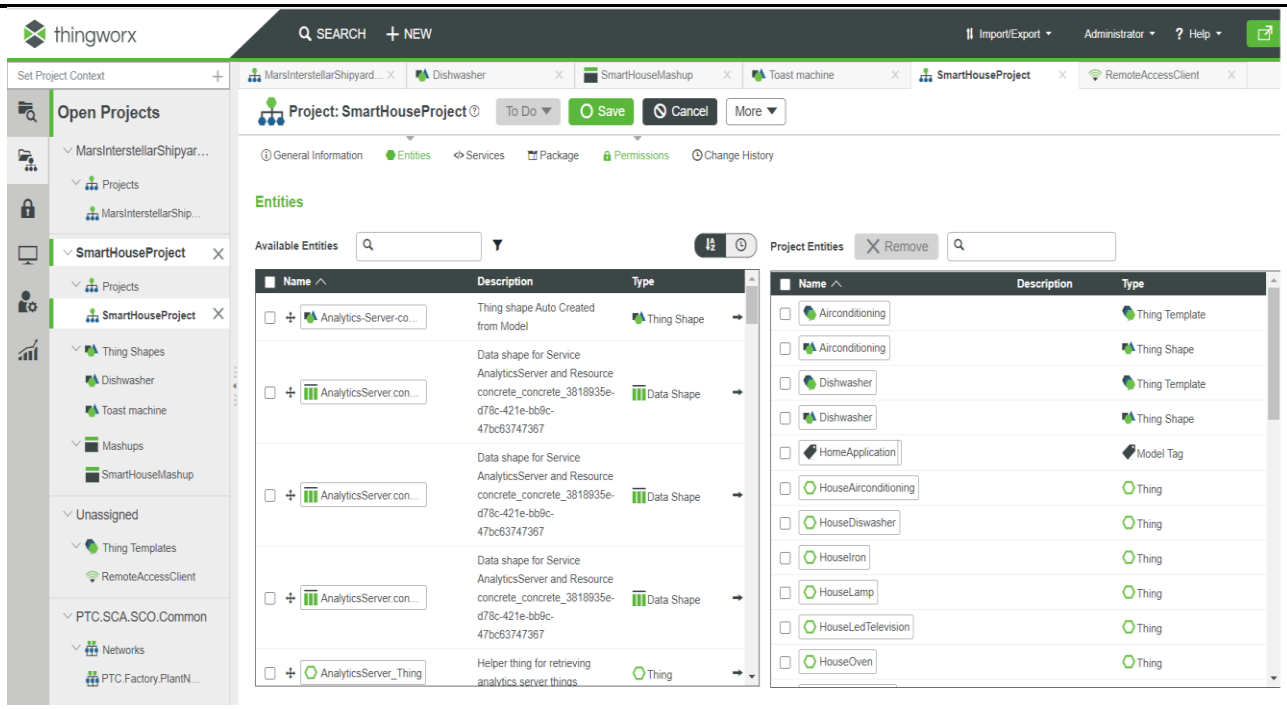
İŞ YAPRAĞI	Staj Tarihi:	Yapılan İş:	Sayfa No:	
	01.07.2020-28.07.2020	Table of Contents Aection	Yetkili İmza	

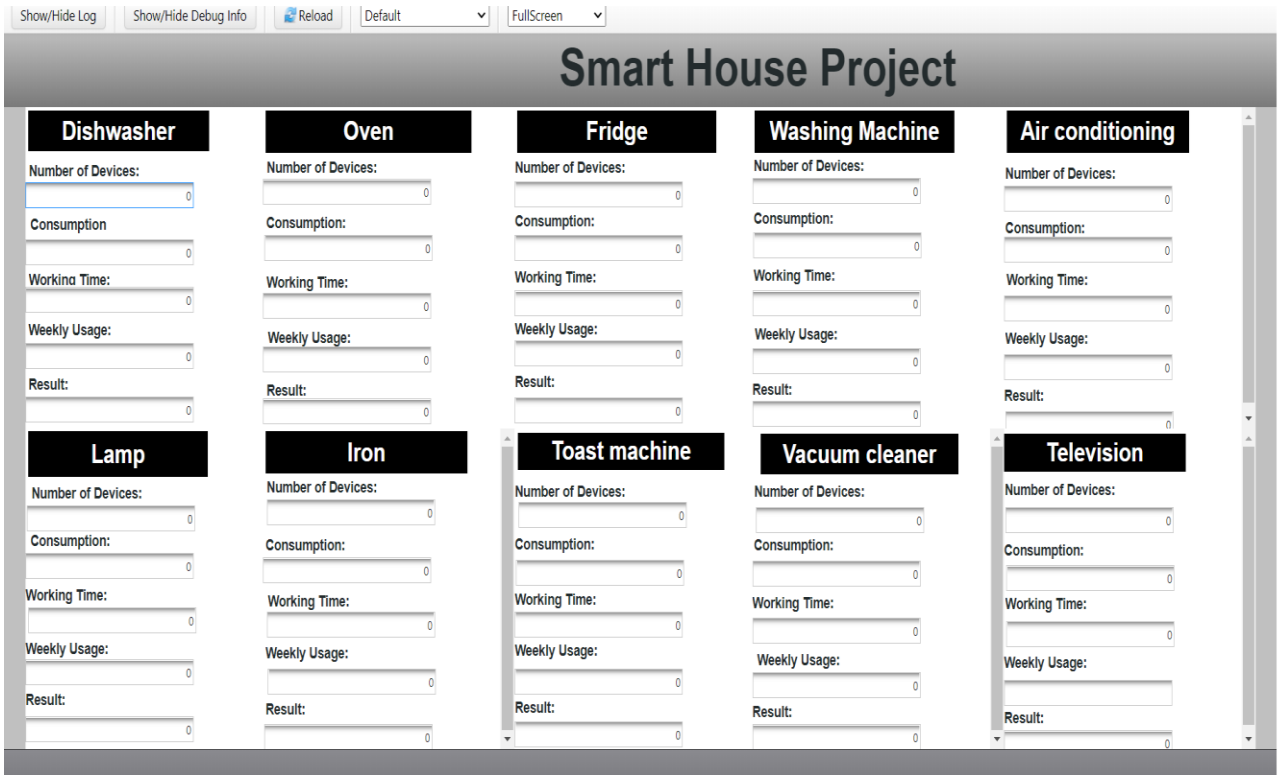
İŞ YAPRAĞI	Staj Tarihi:	Yapılan İş:	Sayfa No:	0
	01.07.2020-28.07.2020	Introduction	Yetkili İmza	

This internship will be on the internet of things. We can say briefly iot for the internet of things. iot is in a very important position today. Exactly iot means , The Internet of Things technology, is the communication and communication of smart devices. Today, the internet of things extends from small appliances to smart cities. The data consists Here the big data that large data emerges as. In addition, this internship will be a software internship. At the end of this internship, by linking the products with the internet, I can provide more convenience with these analyzes. I will also make a simple project within the internship and at the end of the internship, I have a more complex project. It is used in many factories today, absolutely iot should be used in every factory because they can reduce the number of employees and achieve better results in less time.I hope to have a good internship.

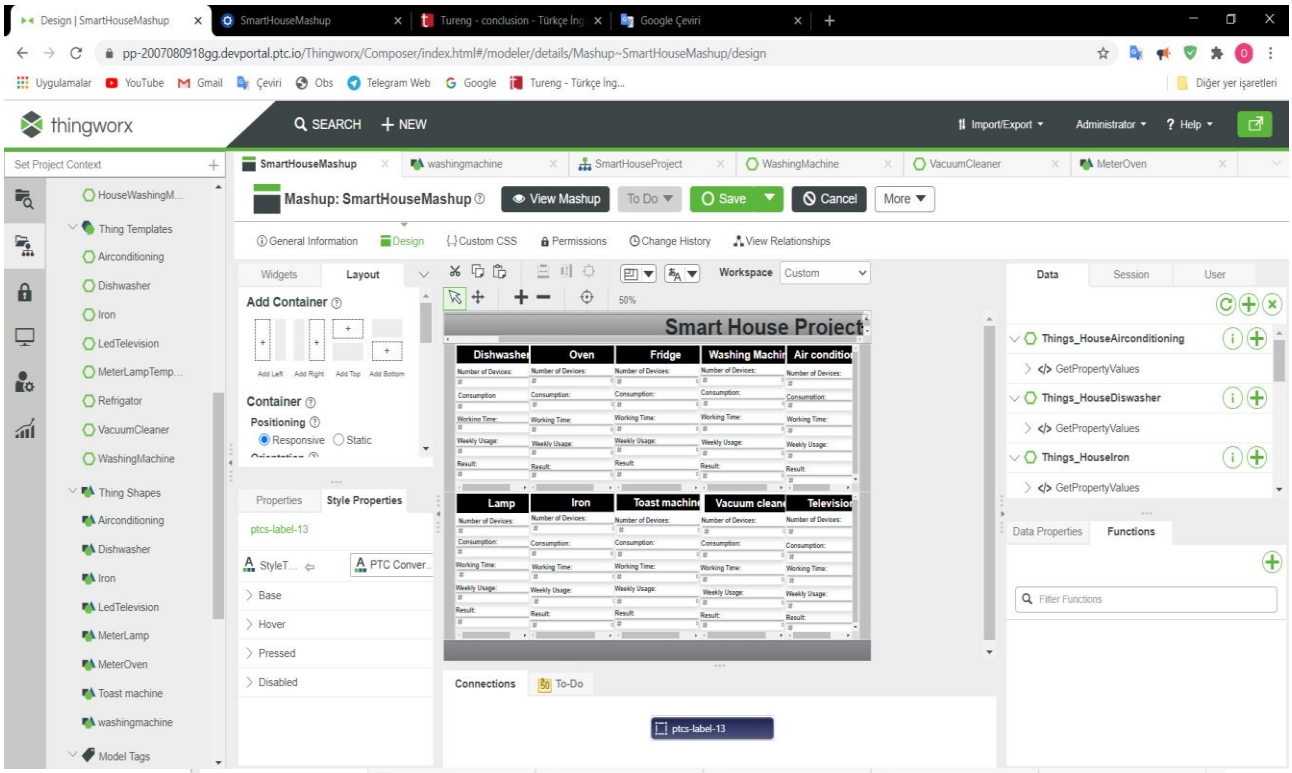
1. https://trainingcentral.ptcu.com/learn/catalog/view/14
2. https://trainingcentral.ptcu.com/learn/course/82/thingworx-basics-of-the-thingworx-development-process
3. https://trainingcentral.ptcu.com/learn/course/83/thingworx-basics-of-the-thingworx-composer
4. https://trainingcentral.ptcu.com/learn/course/86/thingworx-basics-of-modeling
5. https://trainingcentral.ptcu.com/learn/course/92/thingworx-basics-of-mashup-design-principles
6. https://trainingcentral.ptcu.com/learn/course/87/thingworx-basics-of-mashup-stylizing
7. https://trainingcentral.ptcu.com/learn/course/73/thingworx-basics-of-the-edge-connectivity-development-process
8. https://trainingcentral.ptcu.com/learn/course/89/thingworx-basics-of-connectivity
9. https://pp-2007080918gg.devportal.ptc.io/Thingworx/Composer/index.html

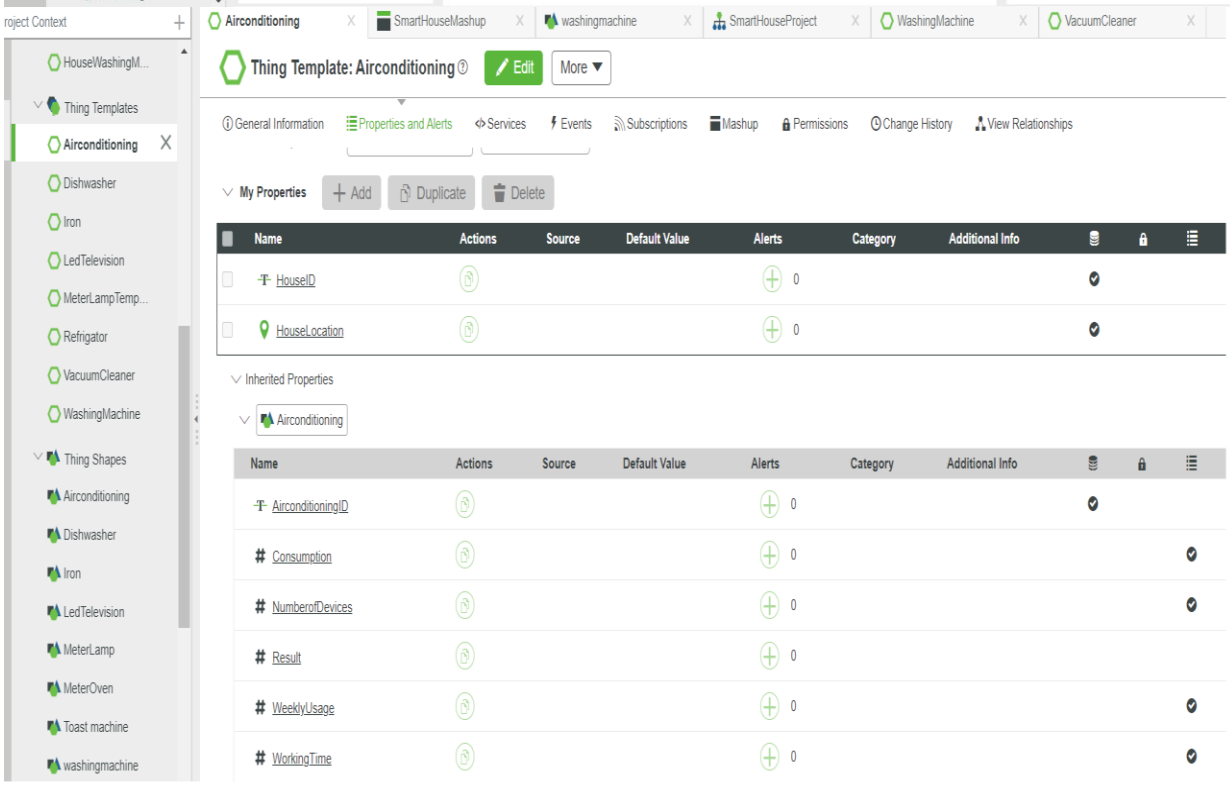
In a busy internship, I had a basic knowledge of the internet of things. I have observed that many analyzes are done correctly thanks to the Internet of Things. During the internship period, I used the thingworx application, which I will apply to what I learned about the internet of things. To use Thingworx, I first finished a training kit. I continued to use the application with a simple project. It was a bit difficult for me to finish the project on my own because we could not go very often because of the covid-19, but thanks to the tutorial's tutorial, I finished this simple project. Then I started a more complex application that I am working on for my own development and to prove what I learned. My ongoing project is an application that balances the electricity of the whole house and shows how much it has been used in the end. Thanks to this application, many people will be able to easily analyze which product uses how much electricity. My suggestions about the Thingworx application are as follows: Certainly, many factories should be implemented immediately. Thanks to this, they can make an easy reporting. Employees can simply explain. Many engineers can use this application because the training set is very useful and can do basic applications in this regard. High level projects can be done when they spend more time. In the developing world of technology, the internet of things and iot platforms will be at a very high level.

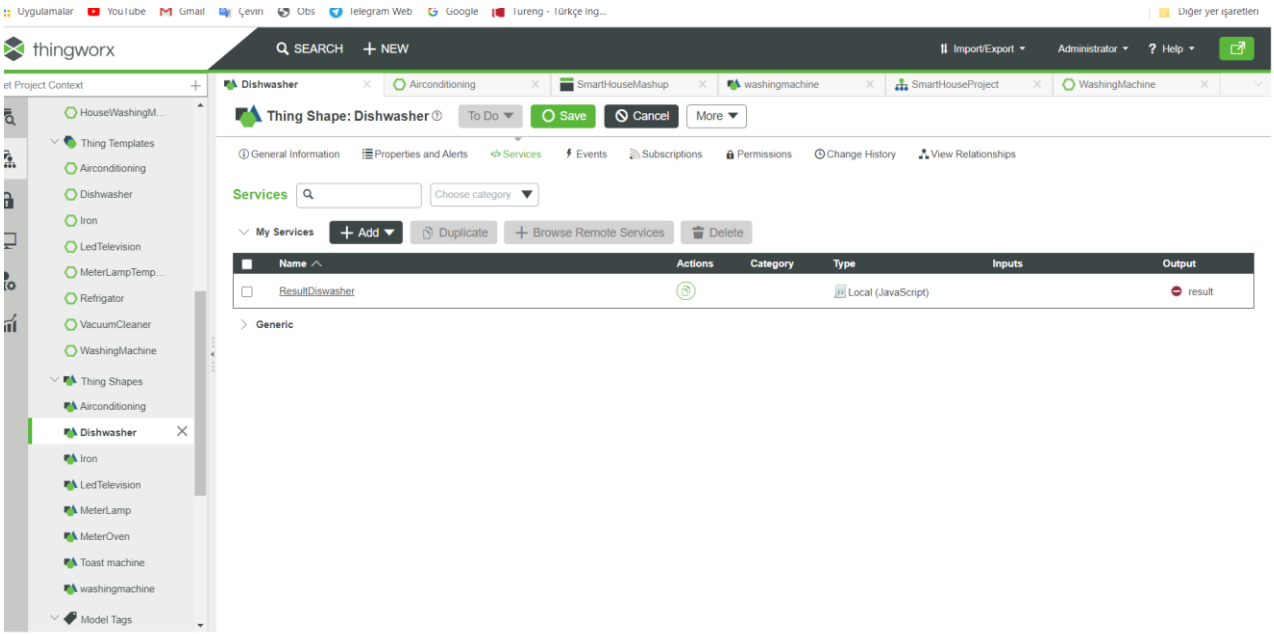

thingworx
SEARCH
NEW
Import/Export
Administrator
Help
Open Projects
MarsInterstellarShipyar...
Projects
MarsInterstellarShip...
SmartHouseProject
Projects
SmartHouseProject
Thing Shapes
Dishwasher
Toast machine
Mashups
SmartHouseMashup
Unassigned
Thing Templates
RemoteAccessClient
PTC.SCA.SCO.Common
Networks
PTC.Factory.PlantN...
Project: SmartHouseProject
To Do
Save
Cancel
More
General Information
Entities
Services
Package
Permissions
Change History
Entities
Available Entities
Project Entities
Remove
Name
Description
Type
Analytics-Server-co...
Thing shape Auto Created from Model
Thing Shape
AnalyticsServer.con...
Data shape for Service AnalyticsServer and Resource concrete_concrete_3818935e-d78c-421e-bb9c-47bc63747367
Data Shape
AnalyticsServer_Thing
Helper thing for retrieving analytics server things
Thing
Airconditioning
Thing Template
Dishwasher
HomeApplication
Model Tag
HouseAirconditioning
HouseDiswasher
HouseIron
HouseLamp
HouseLedTelevision
HouseOven


Show/Hide Log
Show/Hide Debug Info
Reload
Default
FullScreen
Smart House Project
Dishwasher
Oven
Fridge
Washing Machine
Air conditioning
Lamp
Iron
Toast machine
Vacuum cleaner
Television
Number of Devices:
Consumption:
Working Time:
Weekly Usage:
Result:

ZZ


thingworx
SEARCH
NEW
Mashup: SmartHouseMashup
View Mashup
To Do
Save
Cancel
More
Smart House Project
Dishwasher
Oven
Fridge
Washing Machin
Air condition
Lamp
Iron
Toast machin
Vacuum clean
Televisio
Things_HouseAirconditioning
Things_HouseDiswasher
Things_HouseIron
GetPropertyValues
Connections
To-Do
ptcs-label-13


Thing Template: Airconditioning
Edit
More
General Information
Properties and Alerts
Services
Events
Subscriptions
Mashup
Permissions
Change History
View Relationships
My Properties
Add
Duplicate
Delete
Name
Actions
Source
Default Value
Alerts
Category
Additional Info
HouseID
HouseLocation
Inherited Properties
Airconditioning
AirconditioningID
Consumption
NumberofDevices
Result
WeeklyUsage
WorkingTime


thingworx
SEARCH
NEW
Import/Export
Administrator
Help
Dishwasher
Airconditioning
SmartHouseMashup
washingmachine
SmartHouseProject
WashingMachine
Thing Shape: Dishwasher
To Do
Save
Cancel
More
General Information
Properties and Alerts
Services
Events
Subscriptions
Permissions
Change History
View Relationships
Services
Choose category
My Services
Add
Duplicate
Browse Remote Services
Delete
Name
Actions
Category
Type
Inputs
Output
ResultDiswasher
Local (JavaScript)
result
Generic
Thing Templates
Thing Shapes
Model Tags