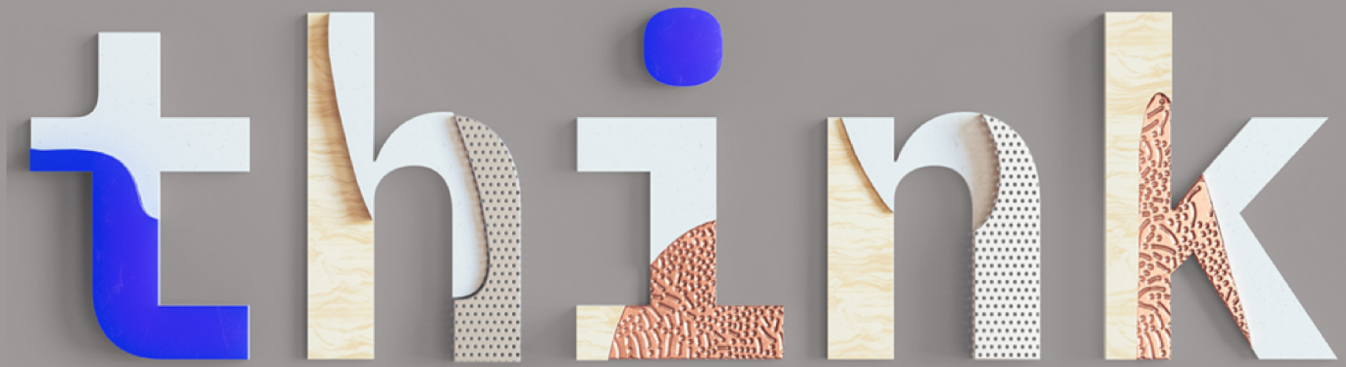
DevZone Lab 4990 Getting started with Watson IoT Platform Data Management

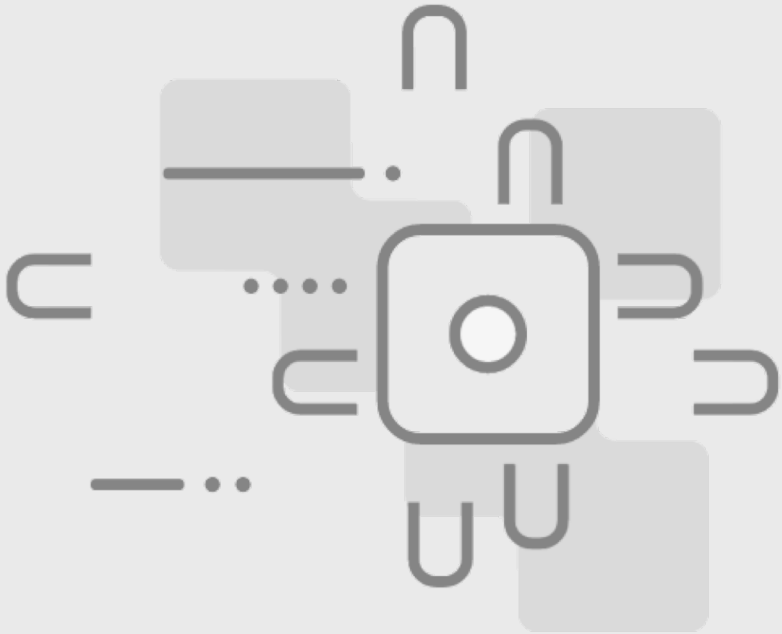
think 2018



Data Management in Watson IoT Platform

In this session we will show how to get started with the IoT Platform and use Data Management to transform and normalize data from connected devices

Using Data Management, you can define event type schemas for your device messages and create interfaces that define abstractions and compute the state of for your devices

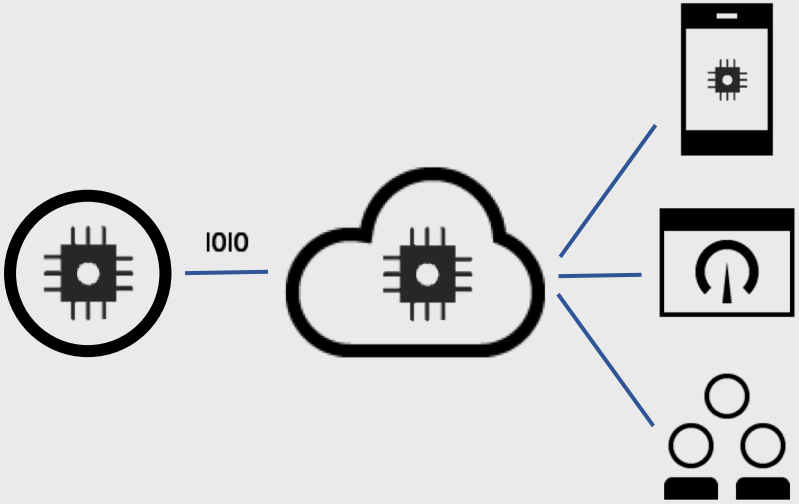


What is the Watson IoT Platform?

The IBM Watson Internet of Things Platform is a fully managed, cloud-hosted service available in IBM Cloud

Devices connect and start sending data securely to the IBM Watson IoT Platform service using the light-weight standardized MQTT messaging protocol

From there, applications can access real-time device data, and devices can be managed using secure APIs or the IoT Platform dashboard



What vocabulary are your devices taking?

Devices of different brands and models uses different message schemas for sending data, even for the same units of measured data

We find that specific code is required to manage the schema variability of device event and command

- Developers need to improve their productivity and speed of delivery
- Developers need a simple programming model and an abstract interface model to insulate the schema variability
- Developers need the platform to take the load off the application code and manage device state and behavior

Celsius temperature as “ambientTemp”

```
{
  "d": {
    "myName": "sensortag",
    "ambientTemp": "30.25",
    "objectTemp": "21.88",
    "humidity": "53.24178",
    "pressure": "1031.17",
    "altitude": "1.43789",
    "accelX": "-0.03",
    "accelY": "0.02",
    "accelZ": "-1.06",
    "gyroX": "-0.85",
    "gyroY": "0.28",
    "gyroZ": "1.30",
    "magX": "46.47",
    "magY": "84.69",
    "magZ": "-45.42",
    "light": "9.49"
  }
}
```

Kelvin temperature as “value”

```
{
  "ts":
  "2017-10-02T08:09:34.310+0000",
  "d": {
    "value": 296.77,
    "time": 1506931831214
  }
}
```

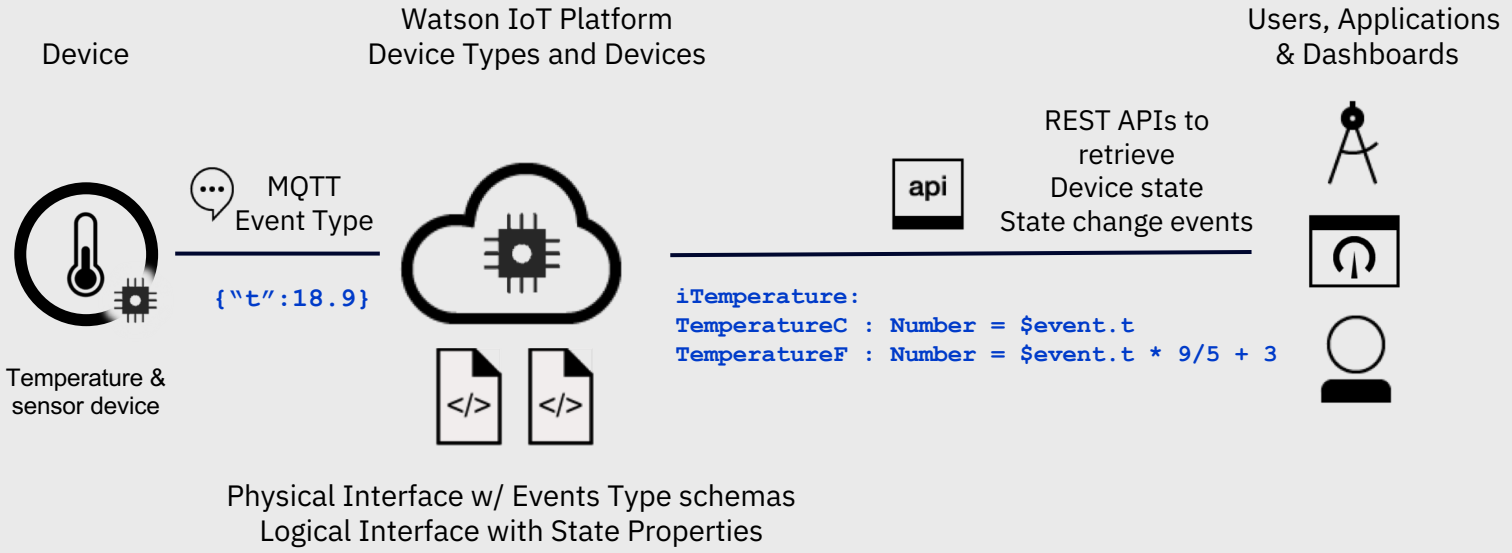
Celsius temperature as “t”

```
{"t": 18.9}
```

Device Abstractions

With Data Management we enable developers to

- Define Physical Interfaces that model events type schemas and filter events
- Define Logical Interfaces that model devices and expose state and change notifications to users and applications



Advanced use of Device Abstractions

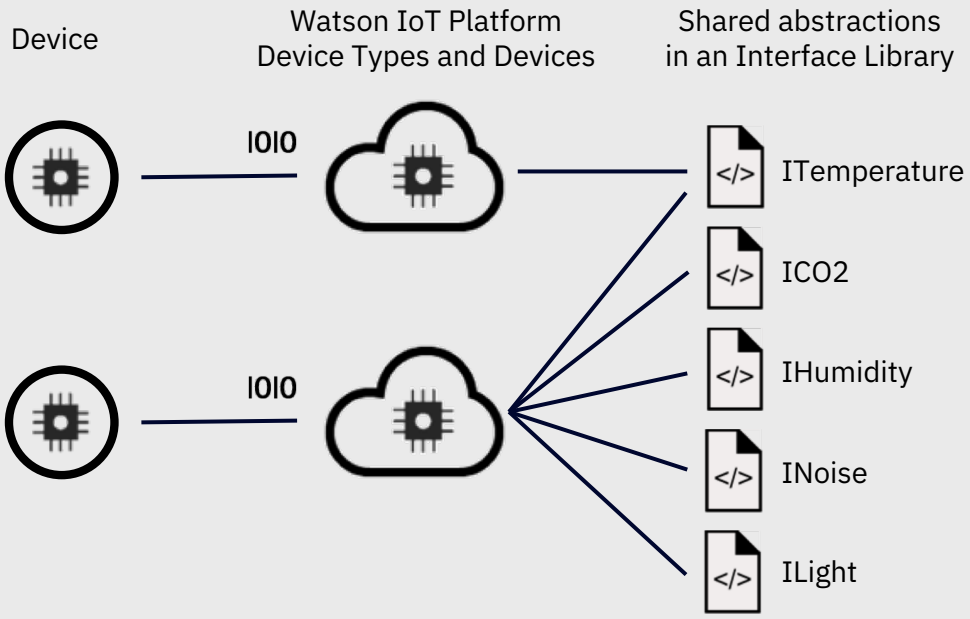
Devices of different brands and models may use shared Logical Interface abstractions

Devices with common device behavior uses shared interfaces

Devices may expose different views of device data to different types of application consumers

Developers may reuse schemas and interfaces in a library across IoT solutions

Developers may manage the lifecycle of schemas and interfaces across their versions



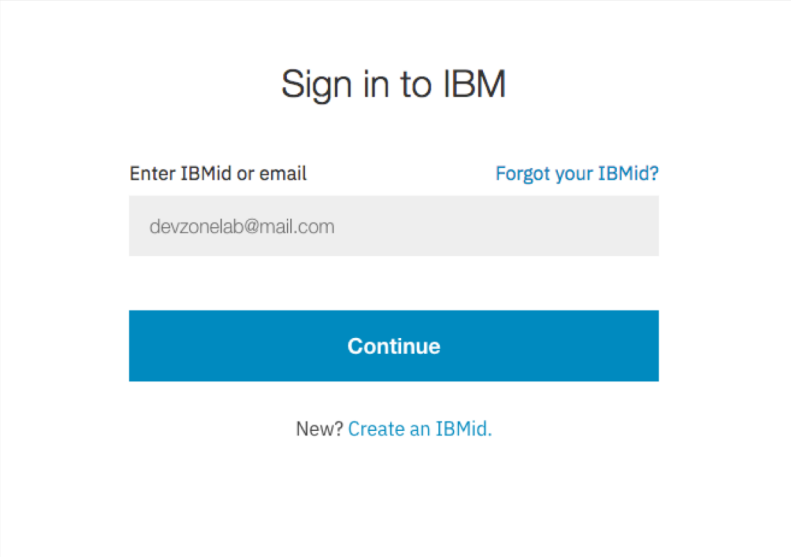
Log into IBM Cloud and the Watson IoT Platform

Lets first log into IBM Cloud

1. Open a browser window on your workstation
2. Go to http://bluemix.net
3. Log in as

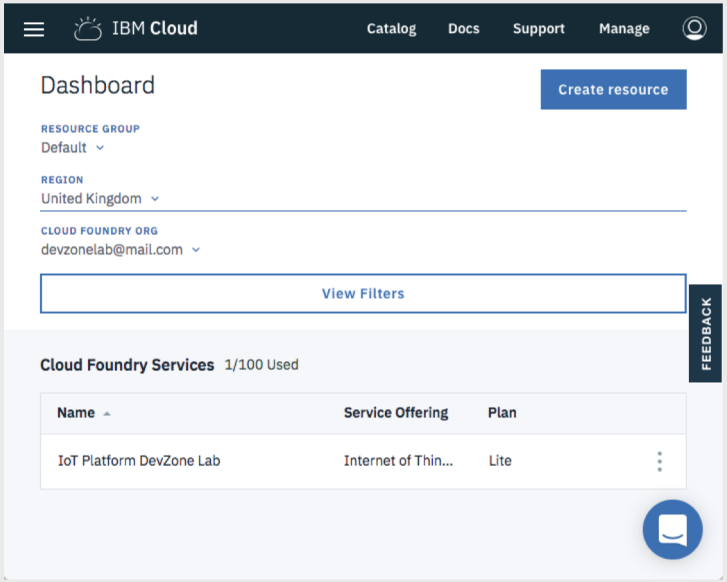
   User: -

   Password: -



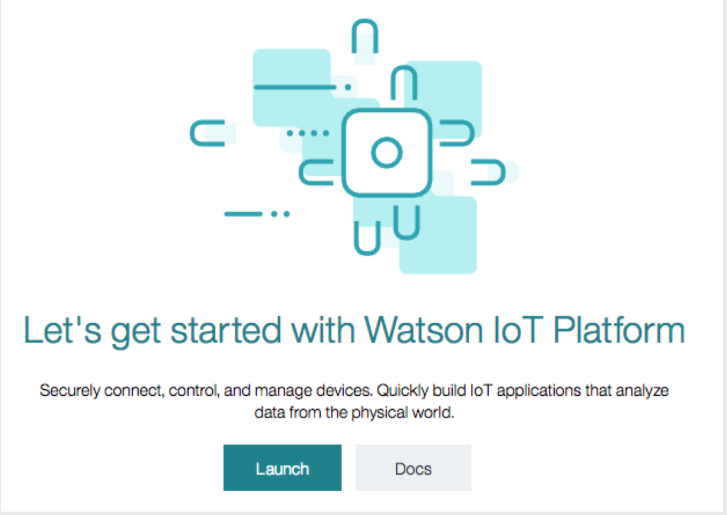
Note: You have to log out from IBM cloud if you are logged in as another user

Locate the Internet of Things Platform



The IBM Cloud Dashboard is loaded

1. In the list of Services, click and open IoT Platform DevZone Lab Internet of Things Platform service
2. In the Watson IoT Platform service page, click Launch to open the IoT platform web interface



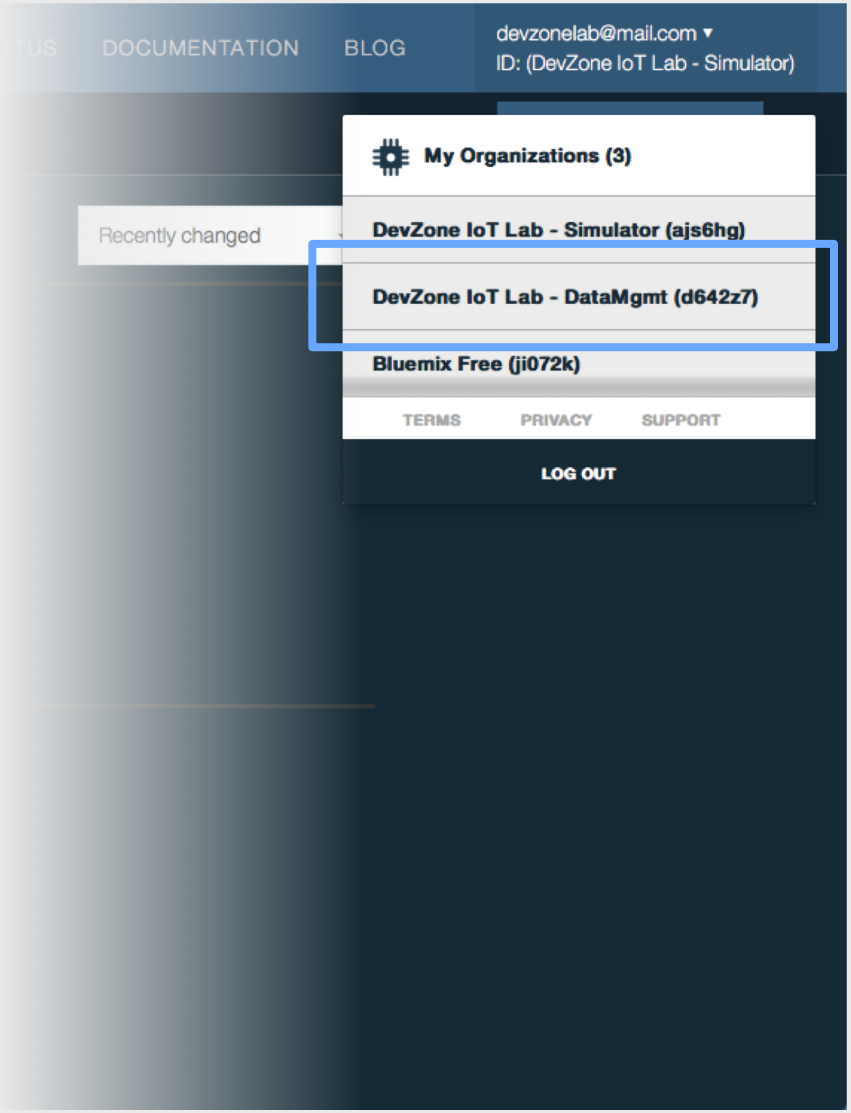
Locate the Internet of Things Platform Organization

The IoT Platform is loaded

The devzonelab user is configured for multiple Think DevZones labs

In this lab we will use the the IoT organization with id d642z7

From the menu, in the top right part of the screen, choose DevZone IoT Lab – DataMgmt (d642z7)



4 simple steps

We are now ready to start exploring IoT Platform Data Management

1

Explore Device Events

Monitor devices and the recently received events

2

Explore Physical Interfaces

View the Physical Interface and Event Types configured

3

Explore Logical Interfaces

View the state model and mapping expressions

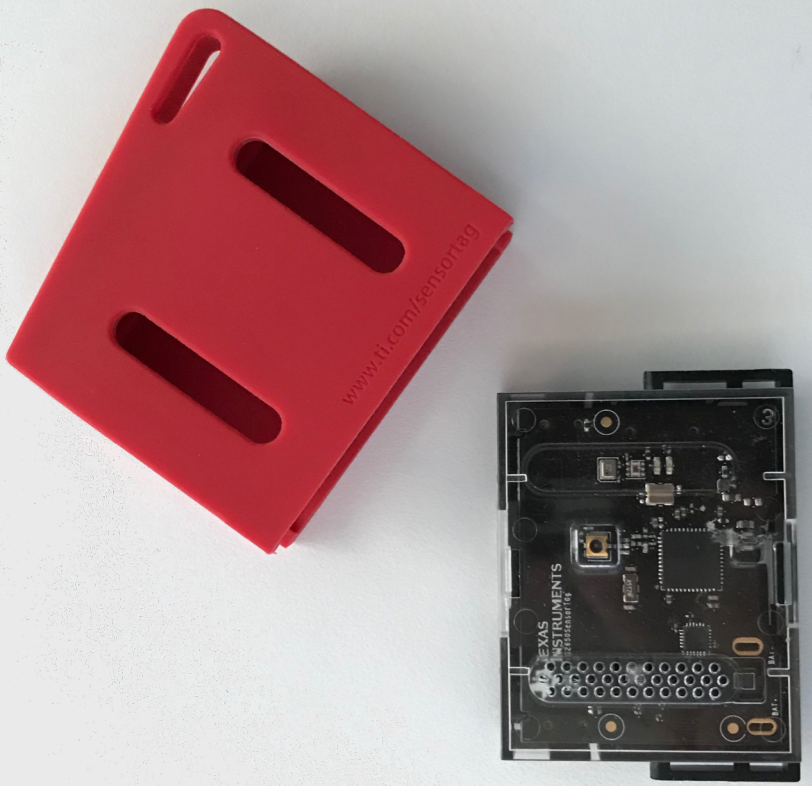
4

Explore Device State

View the device state of the logical interfaces

Device Events

In this session we will use a TI SensorTag sending device data



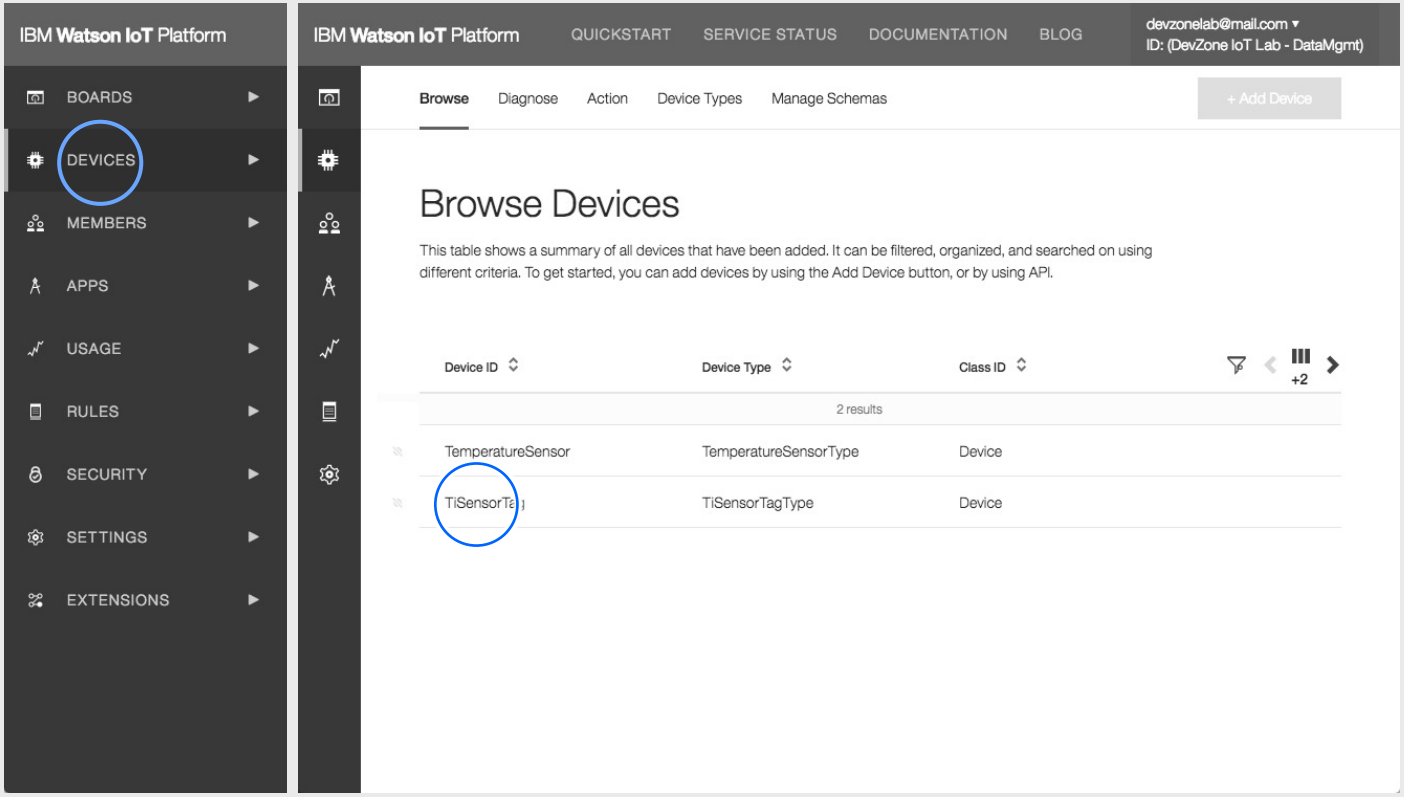

```
{
    "d": {
        "myName": "TiSensorTag",
        "ambientTemp": 24.72,
        "objectTemp": 20.38,
        "humidity": 49.92142,
        "pressure": 1009.75,
        "altitude": -0.08634908,
        "accelX": -0.08,
        "accelY": 0.07,
        "accelZ": 0.96,
        "gyroX": 1.79,
        "gyroY": 3.37,
        "gyroZ": 1.14,
        "magX": 139.56,
        "magY": 188.43,
        "magZ": -343.73,
        "light": 1519.36
    }
}
```

4 simple steps > Explore Device Events

Explore Device Events

Monitor devices and the recently received events

1. Go to Devices
2. In the list of devices, click on TiSensorTag

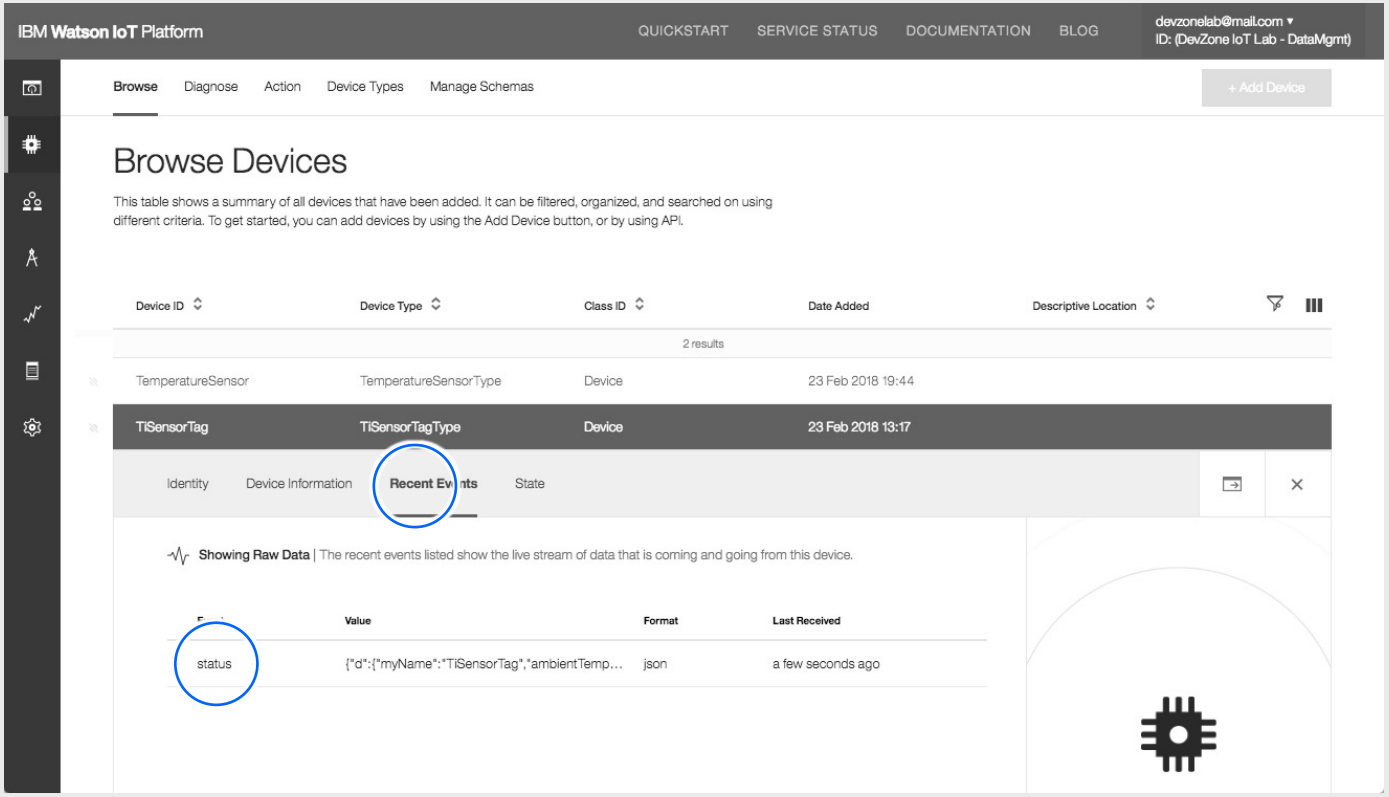


4 simple steps > Explore Device Events

Explore Device Events

The device details are shown

1. Go to the Recent Events tab
2. Wait for the next event to be sent from the Sensor Tag to the IoT Platform
3. Click on the new event

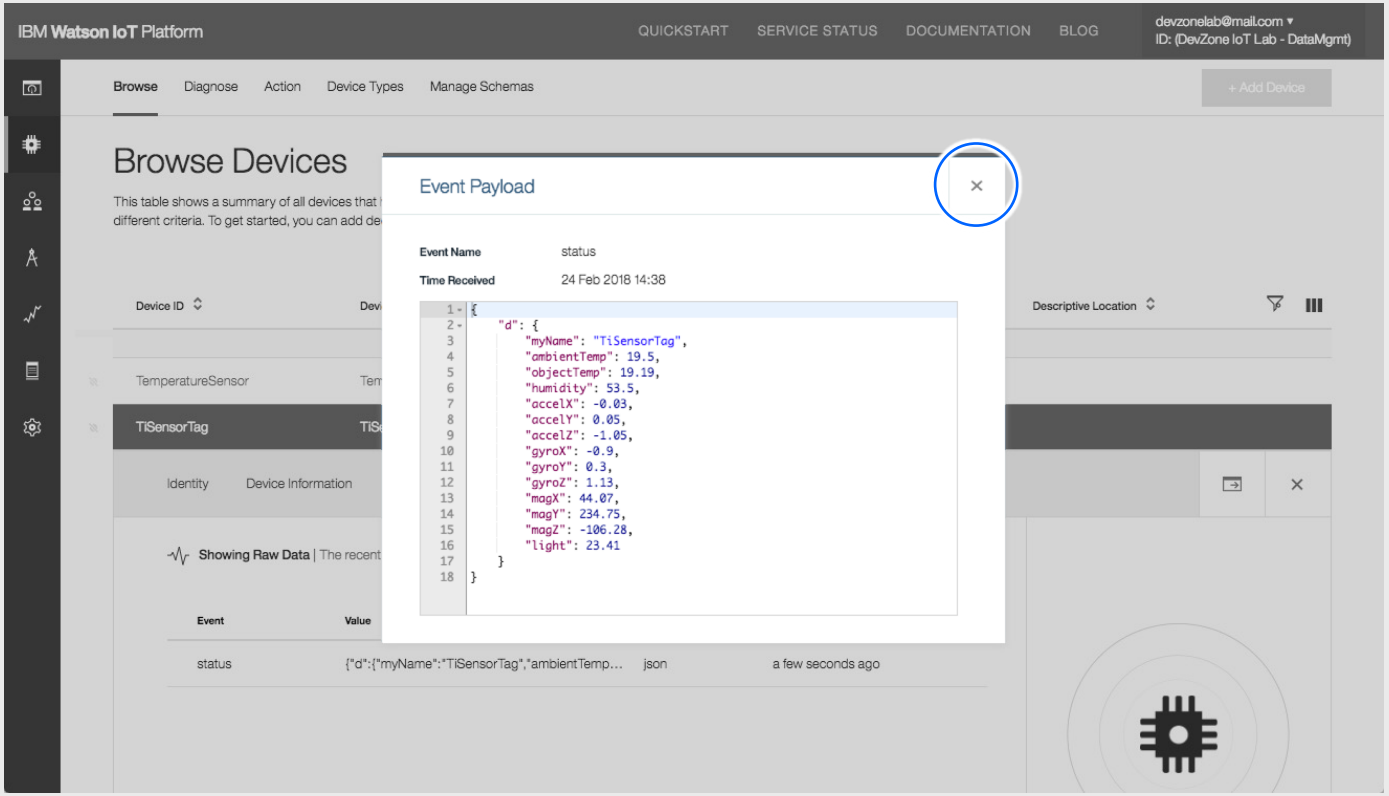


4 simple steps > Explore Device Events

Explore Device Events

The event message and payload is shown

1. View the event data
2. Close the dialog



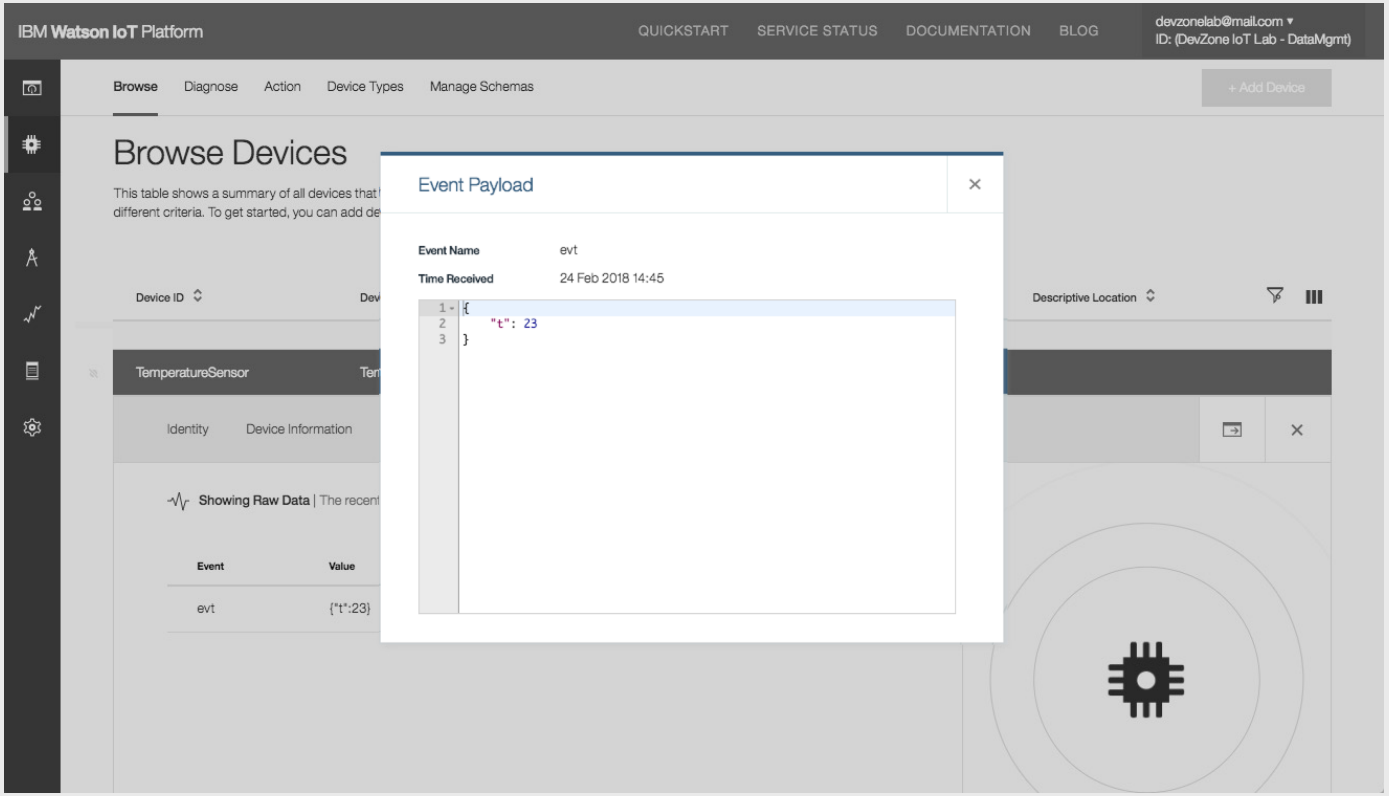
4 simple steps > Explore Device Events

Optionally, explore the TemperatureSensor device

Explore Device Events

Repeat the steps and view the events received from the TemperatureSensor device

1. Select the TemperatureSensor device in the list
2. Select the Recent Events tab
3. Wait for an event
4. View the event message payload



4 simple steps > Explore Device Events

1

Explore
Device Events

Monitor devices and the recently received events

Conclusions

- Devices are sending MQTT event messages to the IoT platform
- The messages contain a JSON payload with the device data
- Messages use different message schemas depending on the device brand, make or version
- Developers need to make changes in the IoT application to adopt to new device message formats
- The IoT Platform and Data Management removes this need

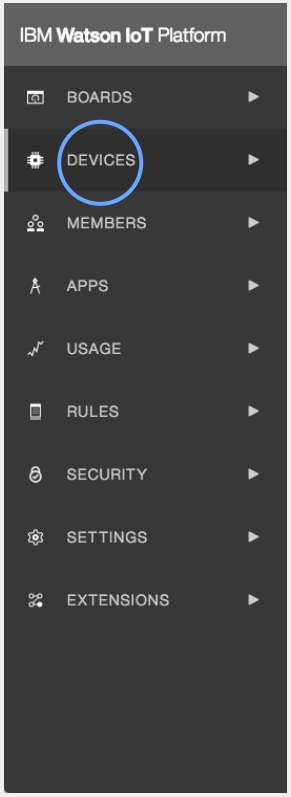
4 simple steps > Explore Physical Interfaces

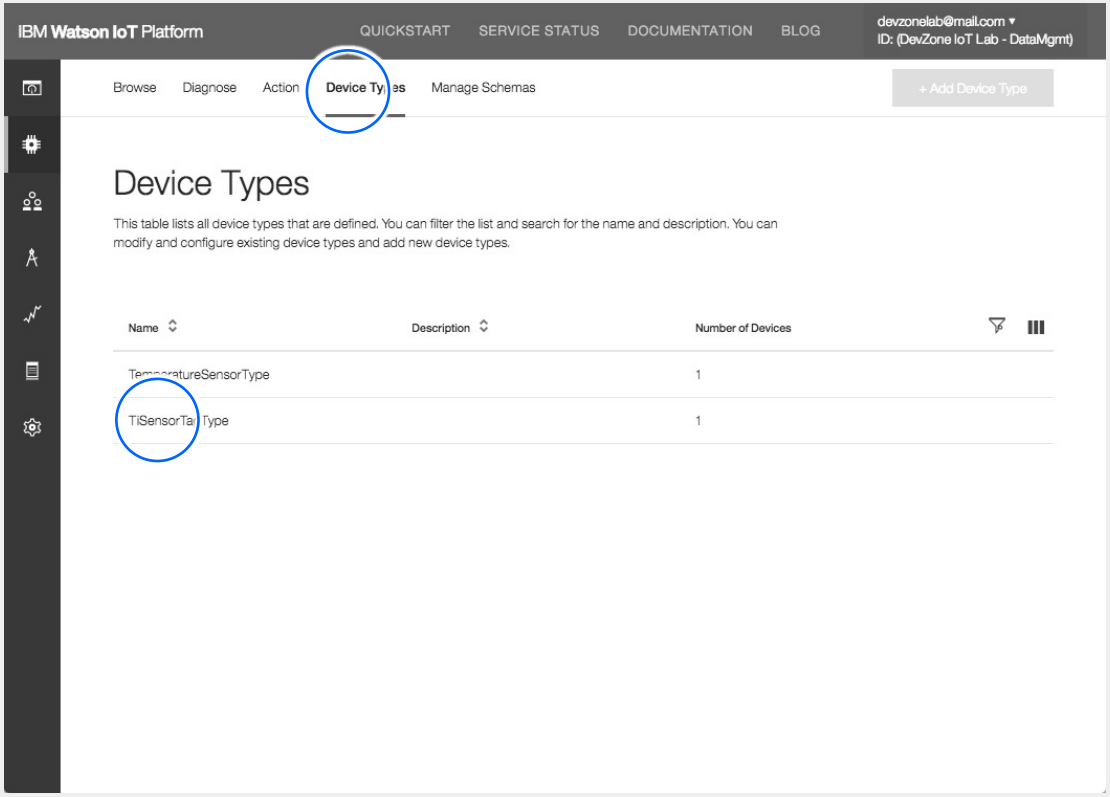
Lets start exploring Data Management Interfaces

Explore Physical Interfaces

View the Physical Interface and Event Types configured

1. In the Device section
2. Click on the Device Type tab
3. In the list, select the TiSensorTag device type



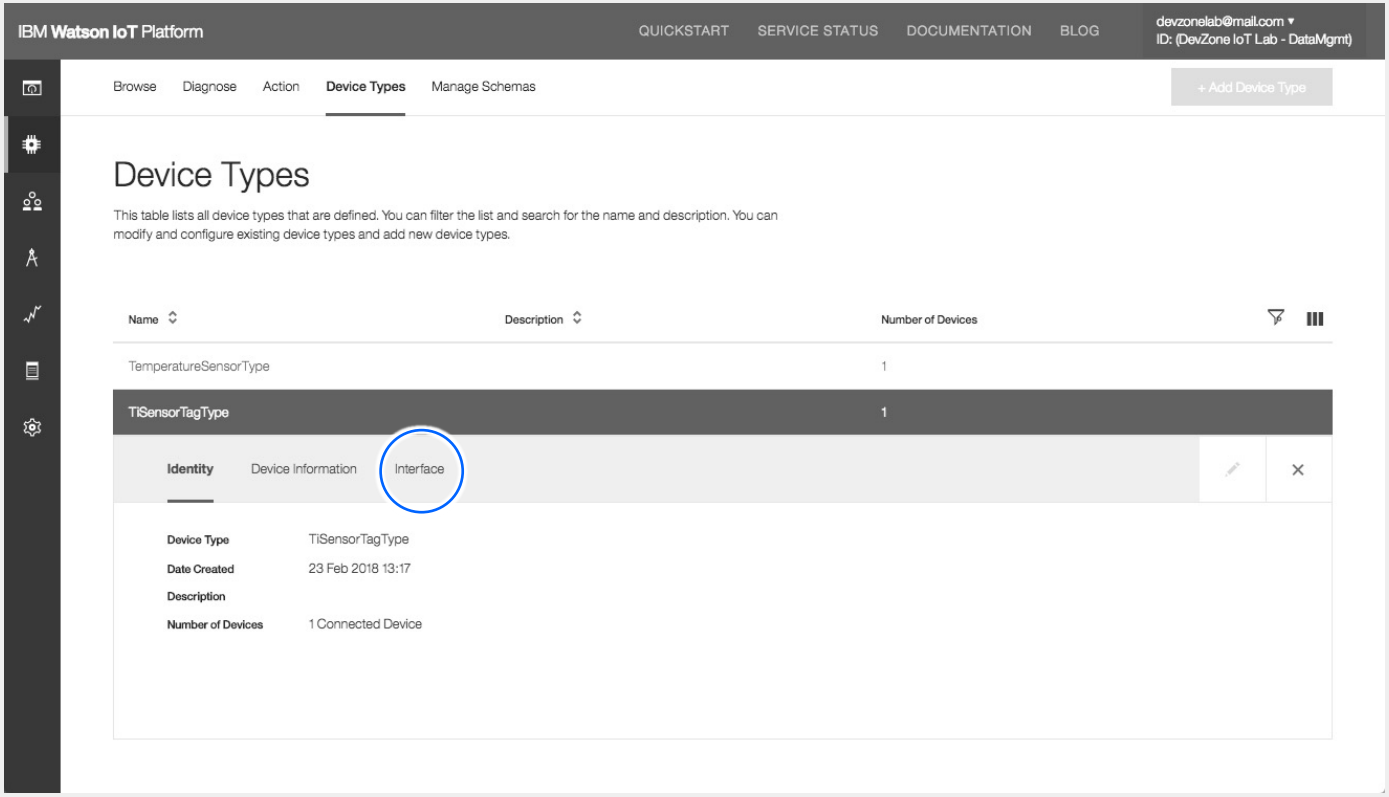


4 simple steps > Explore Physical Interfaces

Explore Physical Interfaces

The TiSensorTag row expands and show type details

Select the Interface tab



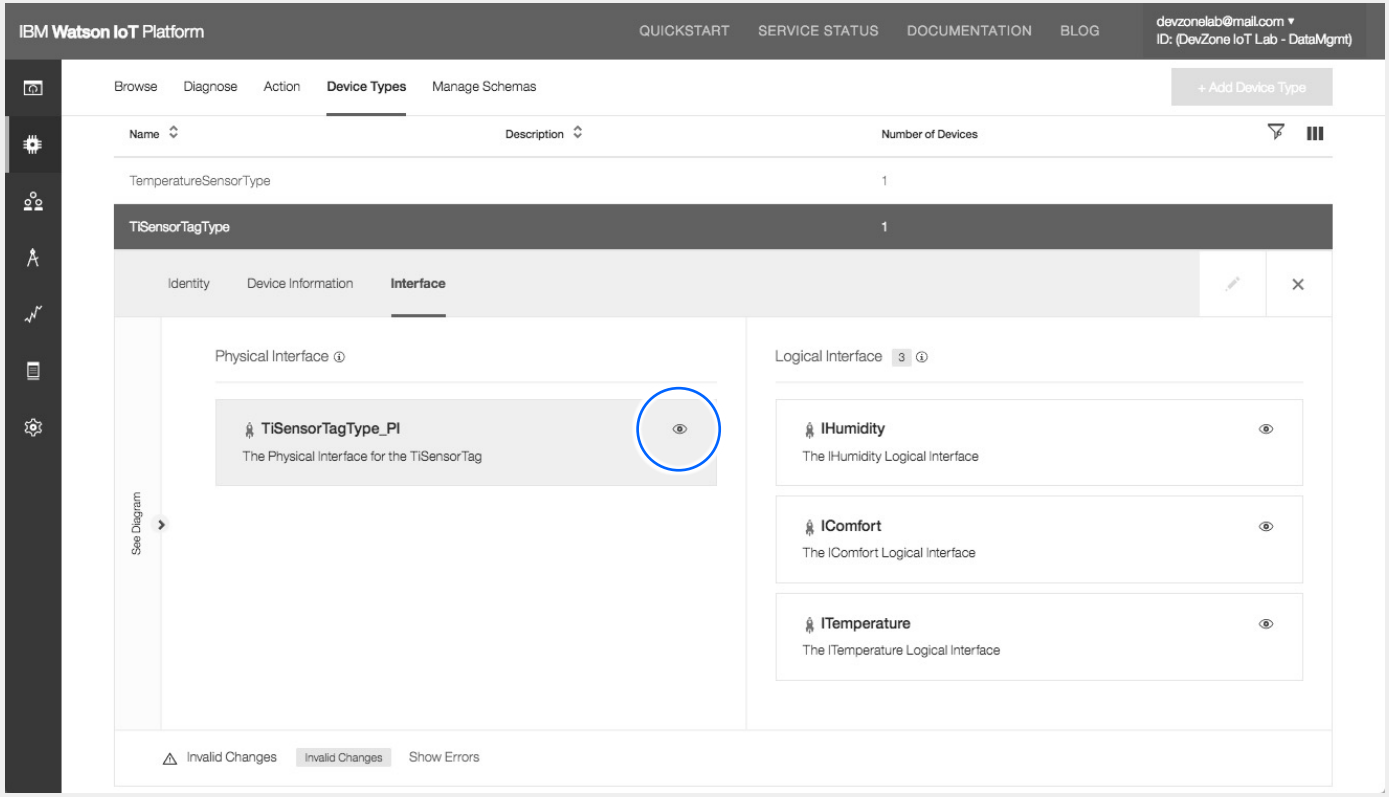
4 simple steps > Explore Physical Interfaces

Lets explore the Physical Interface

Explore Physical Interfaces

The Interfaces tab shows all Physical and Logical interfaces defined for the type

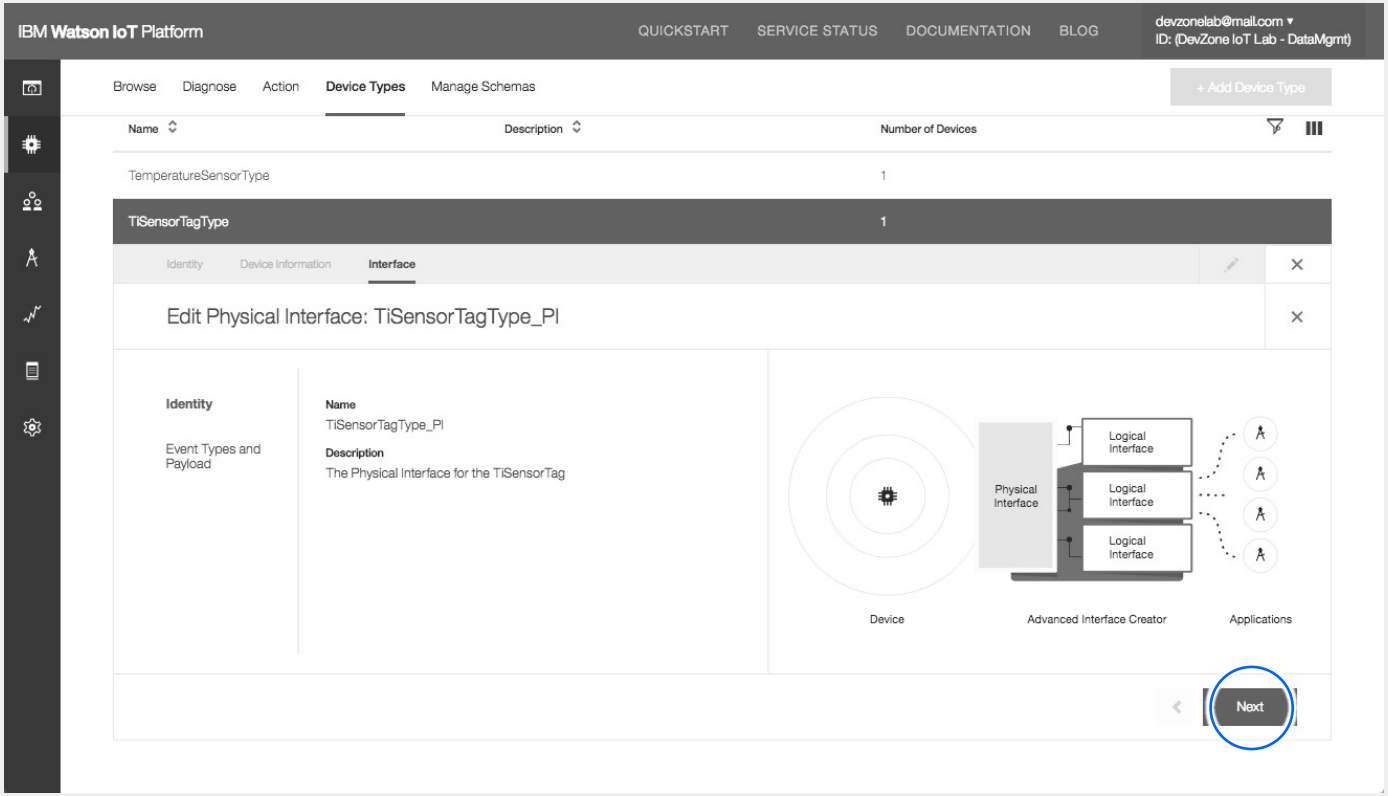
Click the eye icon to view the Physical Interface



4 simple steps > Explore Physical Interfaces

Explore Physical Interfaces

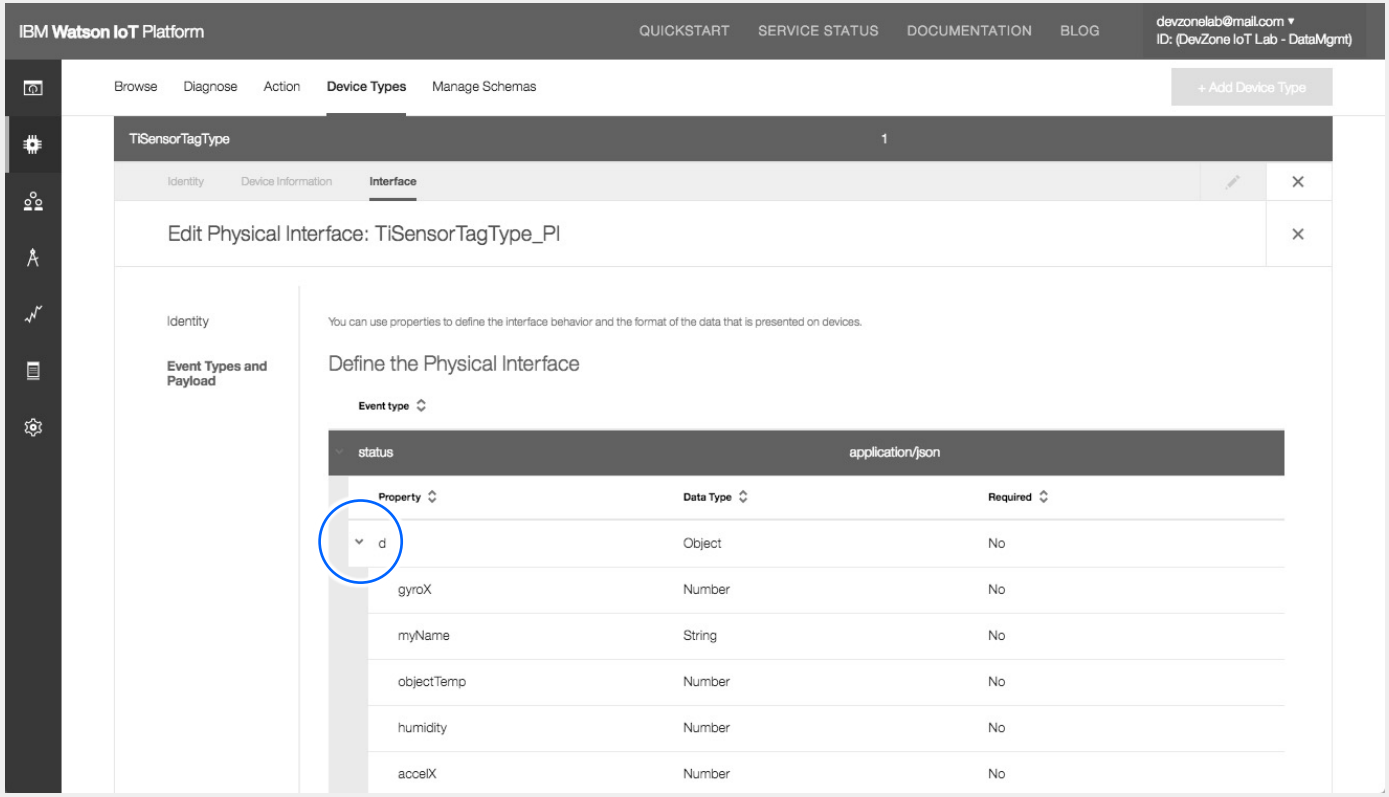
1. View the Identity information for the Physical Interface
2. Click Next



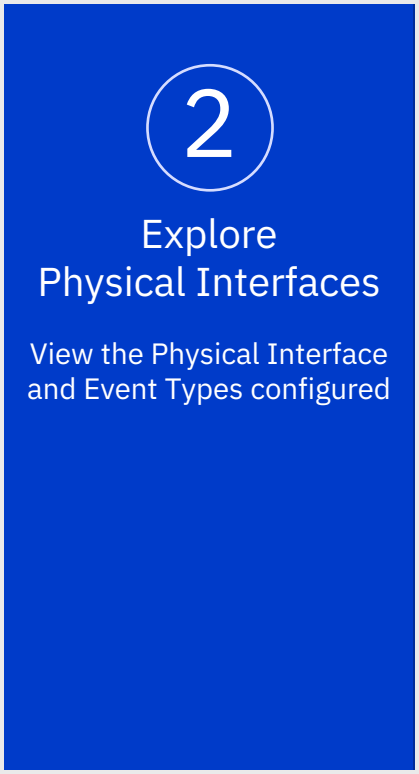
4 simple steps > Explore Physical Interfaces

Explore Physical Interfaces

1. View the Event Type schema for Status events
2. Expand the event type schema hierarchy
3. Click Done to return to the Interfaces page



4 simple steps > Explore Physical Interfaces



Conclusions

- A Physical Interface declares the Event Type schemas for a Device Type
- Events with a matching event type schema will be processed by Data Management
- Events with an unrecognized event type schema will be ignored and not processed. This provides a mechanist to filter device events
- Developers can reuse Event Type schemas across Device Types and their Physical Interfaces

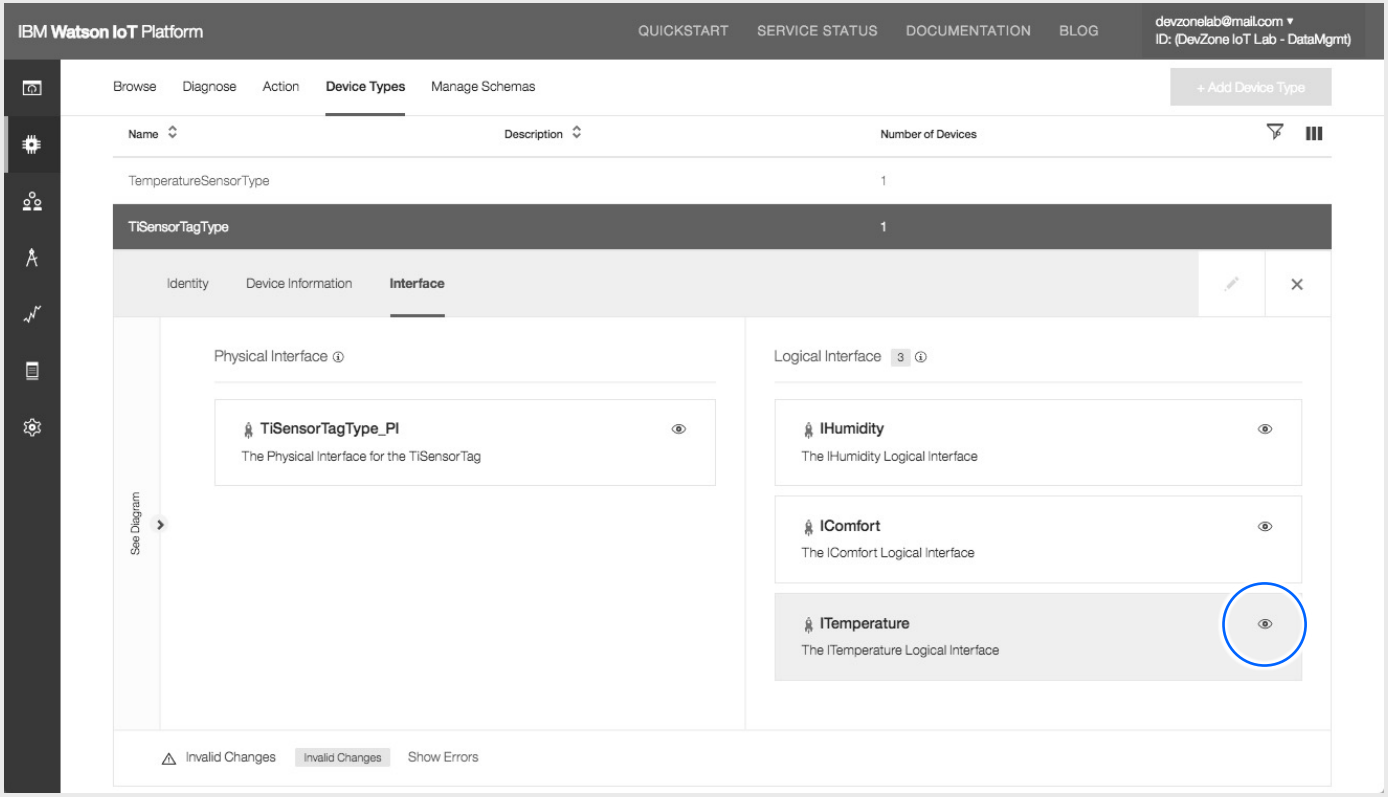
4 simple steps > Explore Logical Interfaces

Lets explore the Logical Interfaces

Explore Logical Interfaces

View the state model and mapping expressions

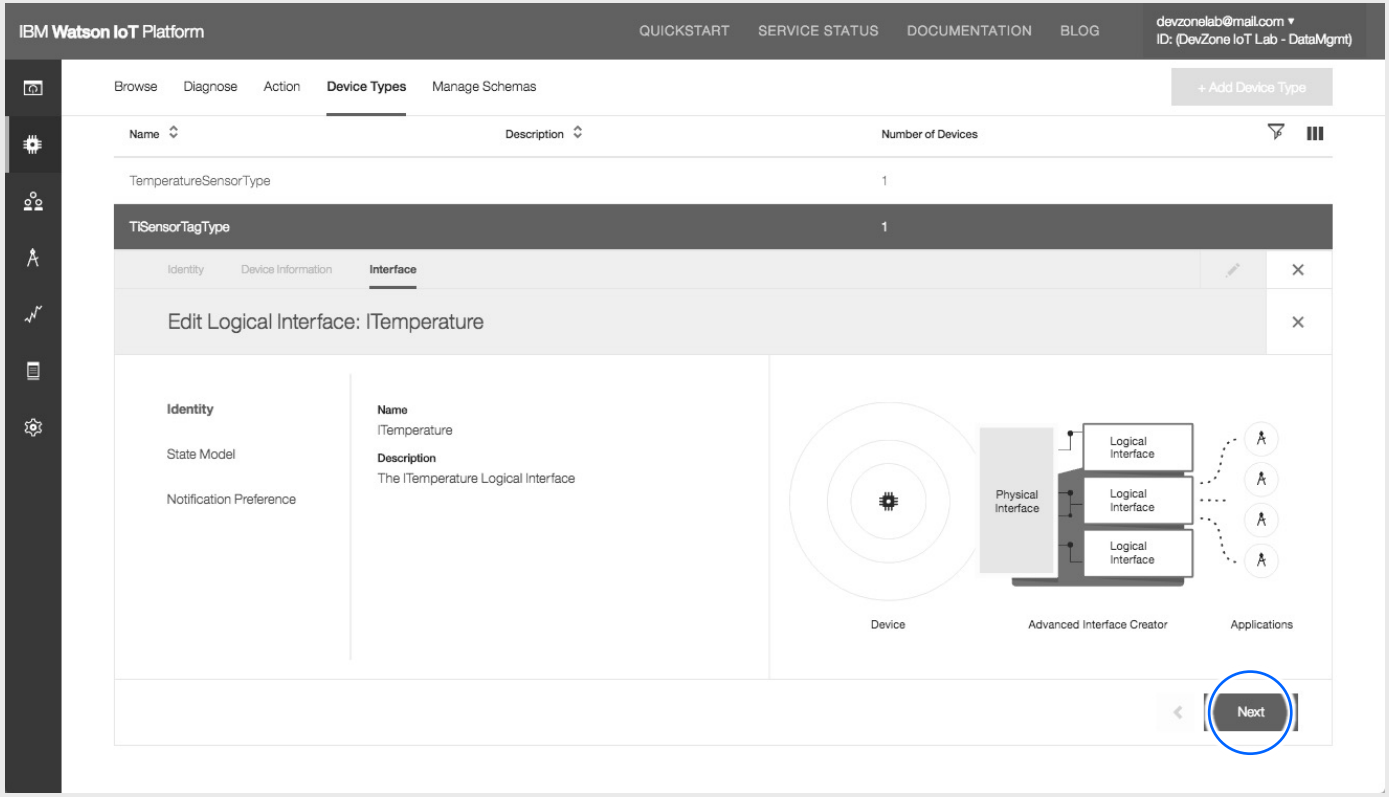
On the Interface tab, click the eye icon on the ITemperature Logical Interface



4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

1. View the Identity information for the Logical Interface
2. Click Next



4 simple steps > Explore Logical Interfaces

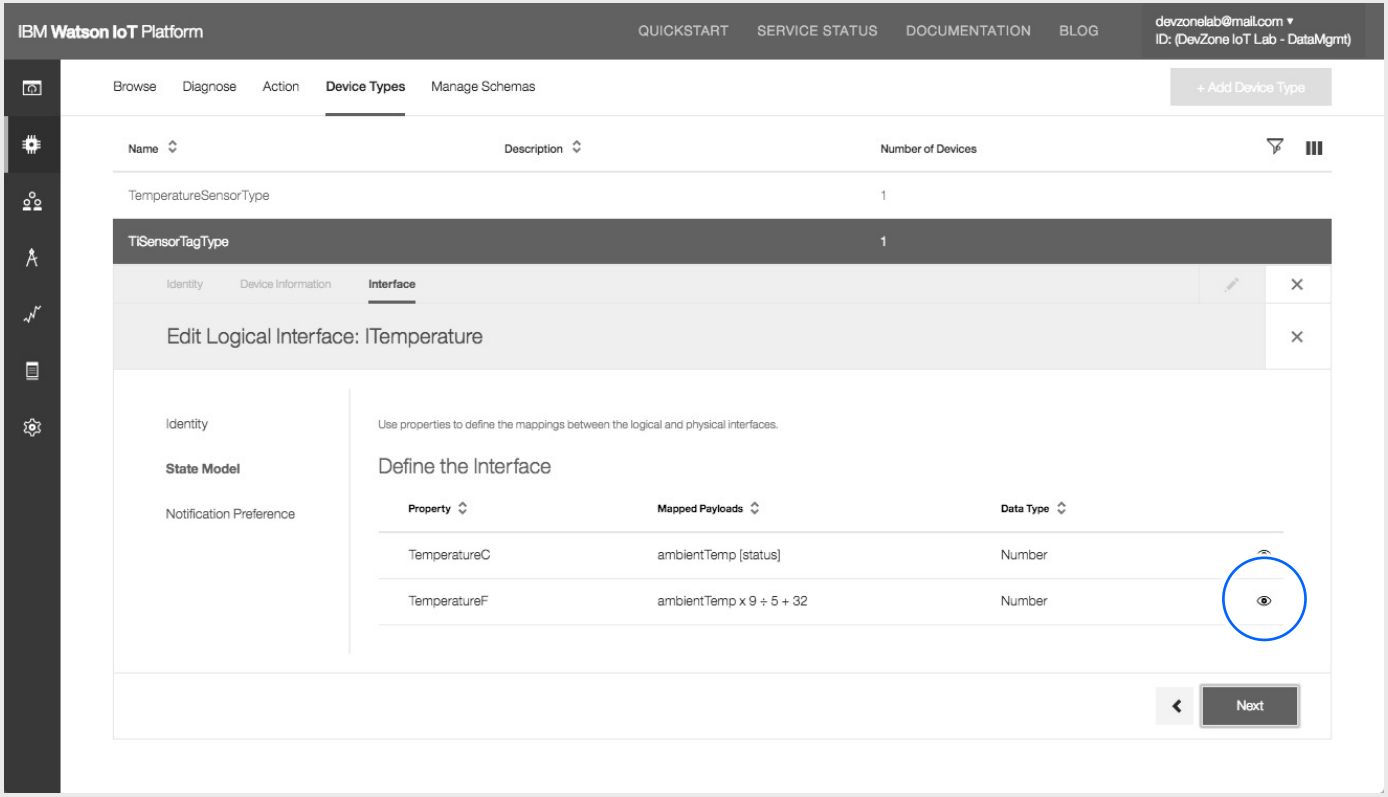
Explore Logical Interfaces

The Logical Interface editor shows the state properties defined in the ITemperature logical interface

The interface declares

- TemperatureC
  the Celsius temperature
- TemperatureF
  the Fahrenheit temperature

Click on the eye icon to view the mapping expression for TemperatureF



4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

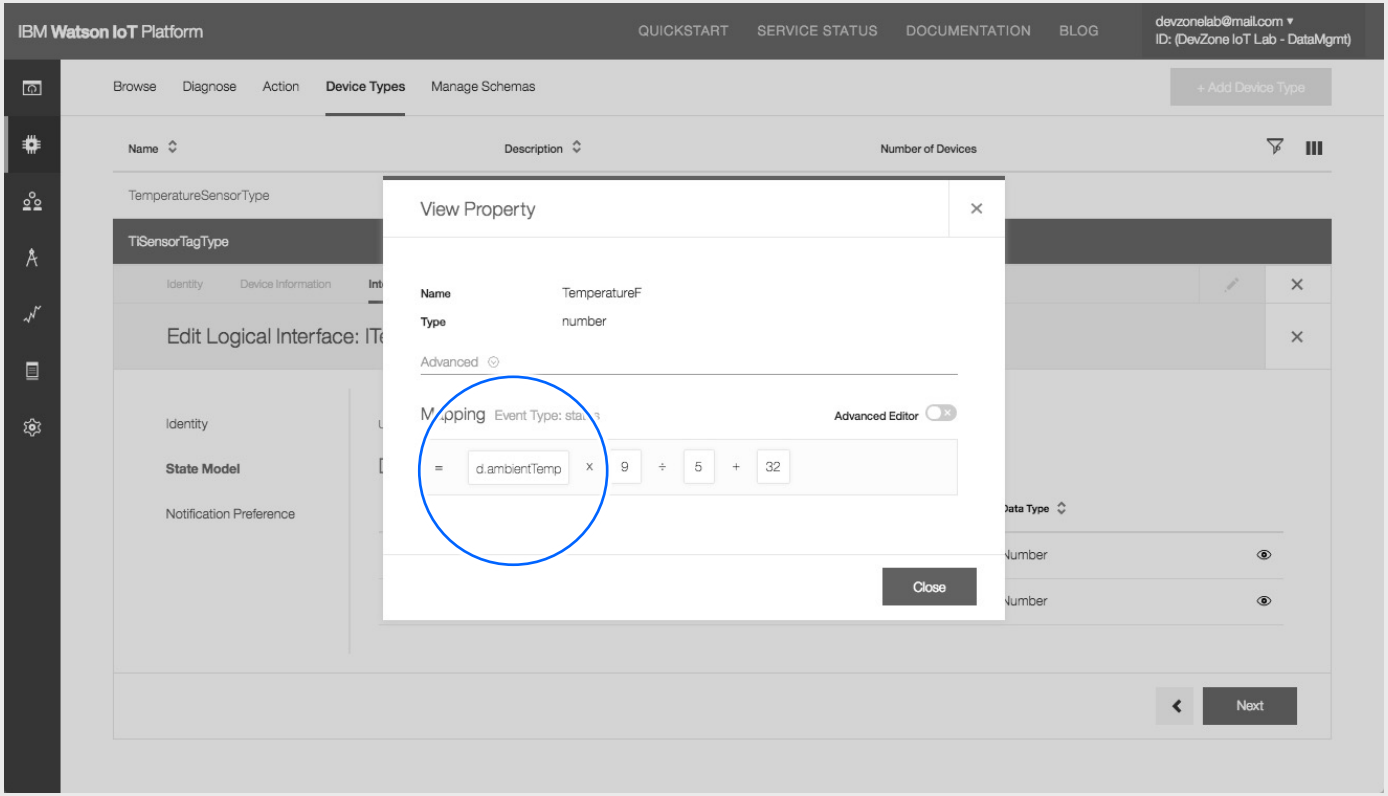
The mapping expression for TemperatureF is shown in the dialog

The mapping expression computes the temperature using the formula

$T_F = T_C * 9/5 + 32$

The T_C temperature value is mapped to the d.ambientTemp value in the Status event

1. Close the dialog
2. Click Next

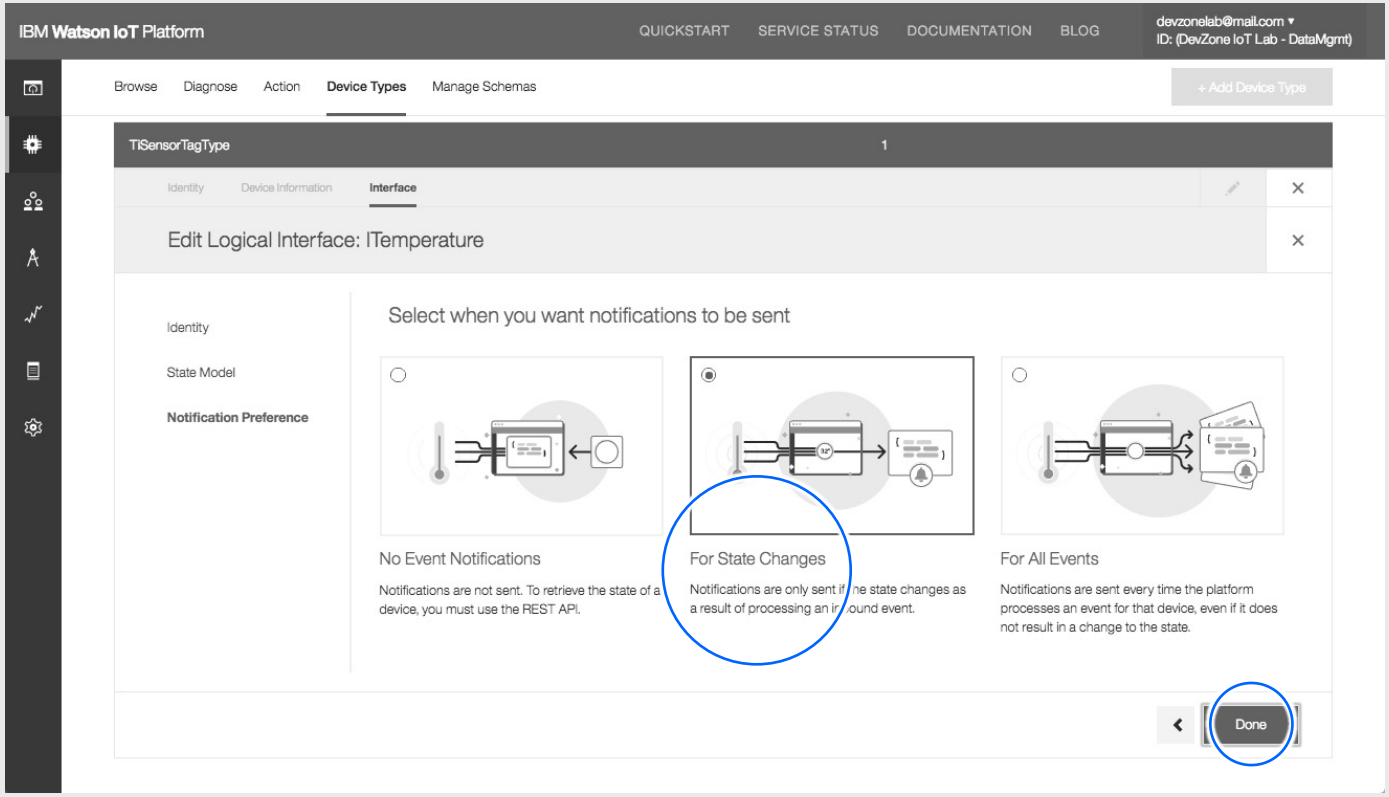


4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

The ITemperature interface has been configured to only send notifications to applications when the state value changes. This reduces network traffic and events sent to cloud applications

Click Done to return to the Interface page



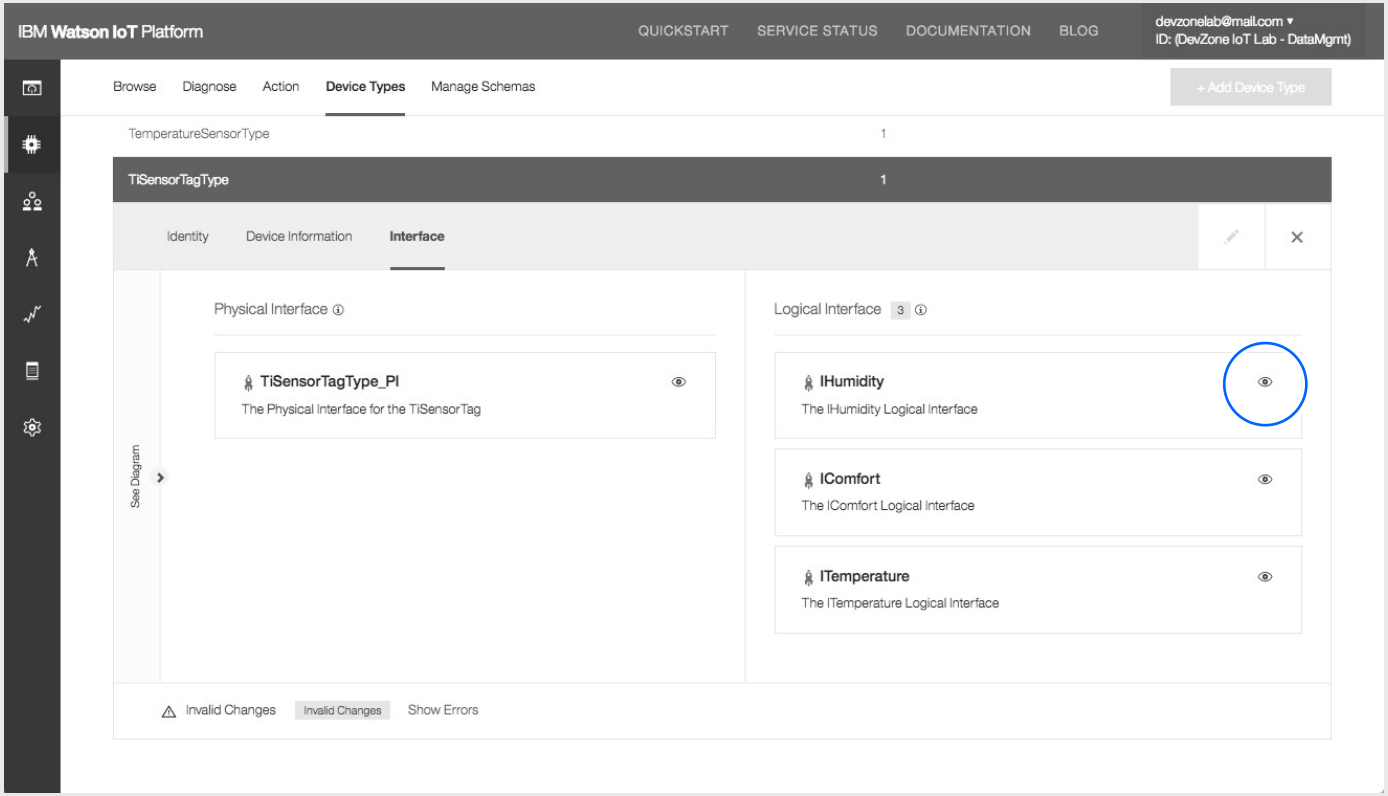
4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

Repeat the steps to view the other interfaces

- IHumidity
- IComfort

Start by clicking the eye icon on the IHumidity interface

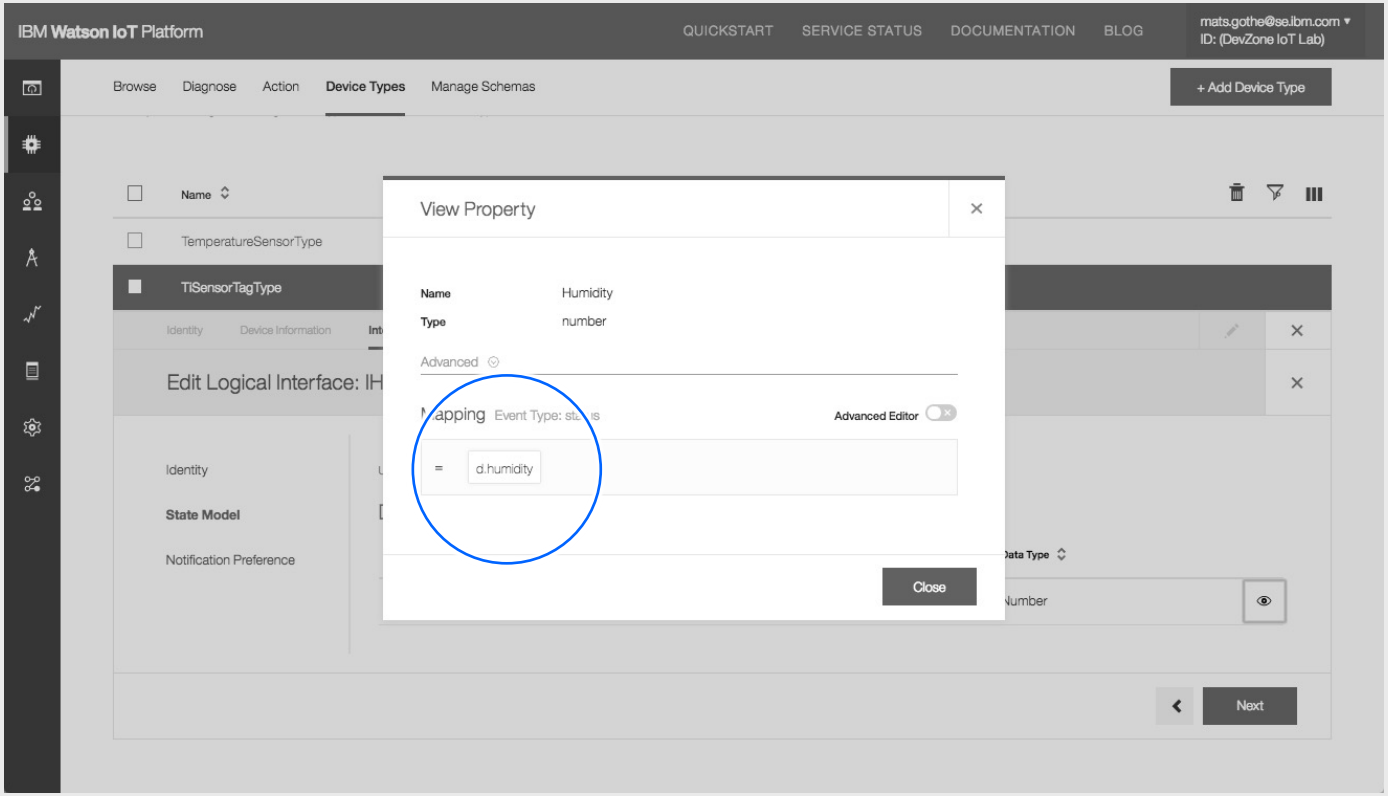


4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

1. In the IHumidity interface, click Next to view the state properties
2. Click on the eye icon on the Humidity property

   Note, the property is simply returning the d.humitity sensor reading
3. Click Next and then Done to return to the Interface tab

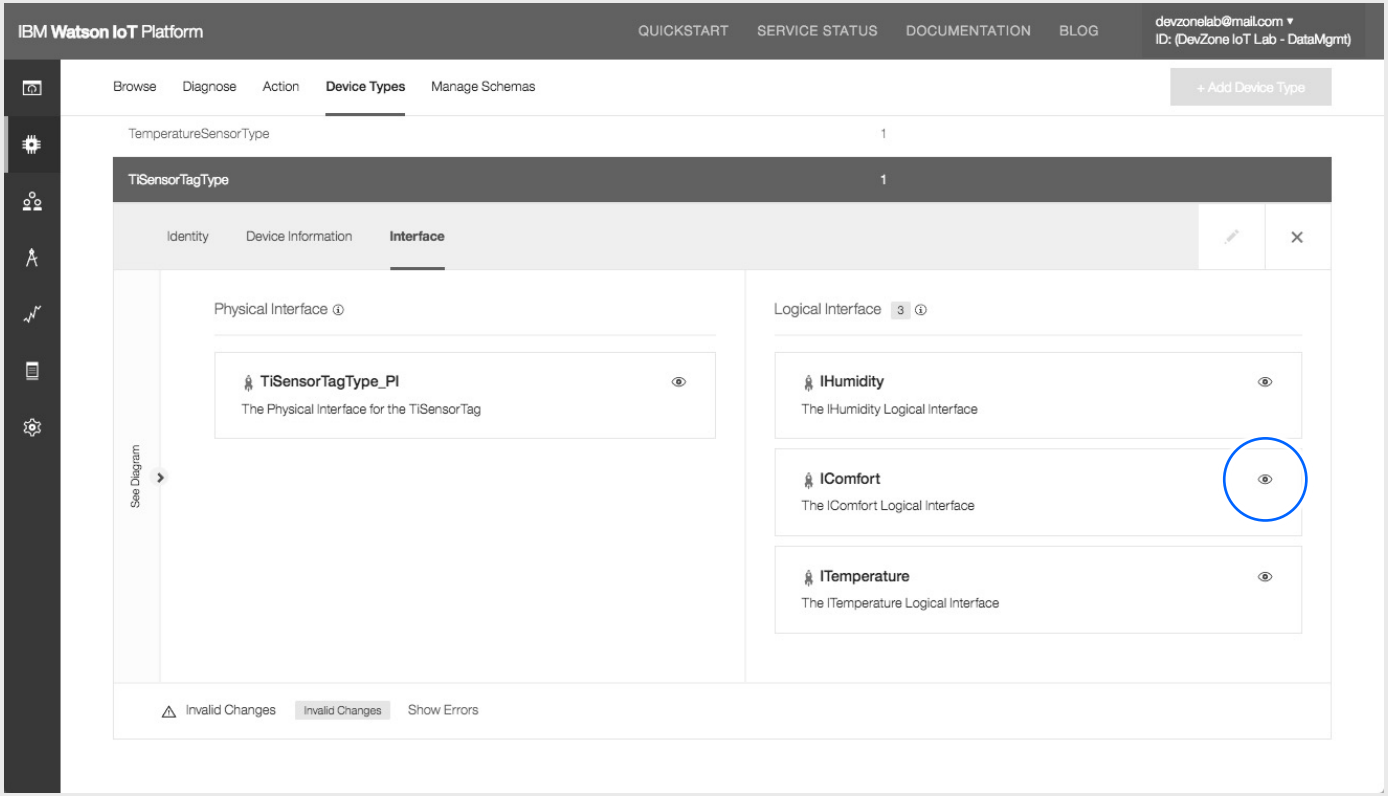


4 simple steps > Explore Logical Interfaces

Explore Logical Interfaces

Repeat the steps to view the IComfort interface

Start by clicking the eye icon on the IComfort interface



4 simple steps > Explore Logical Interfaces

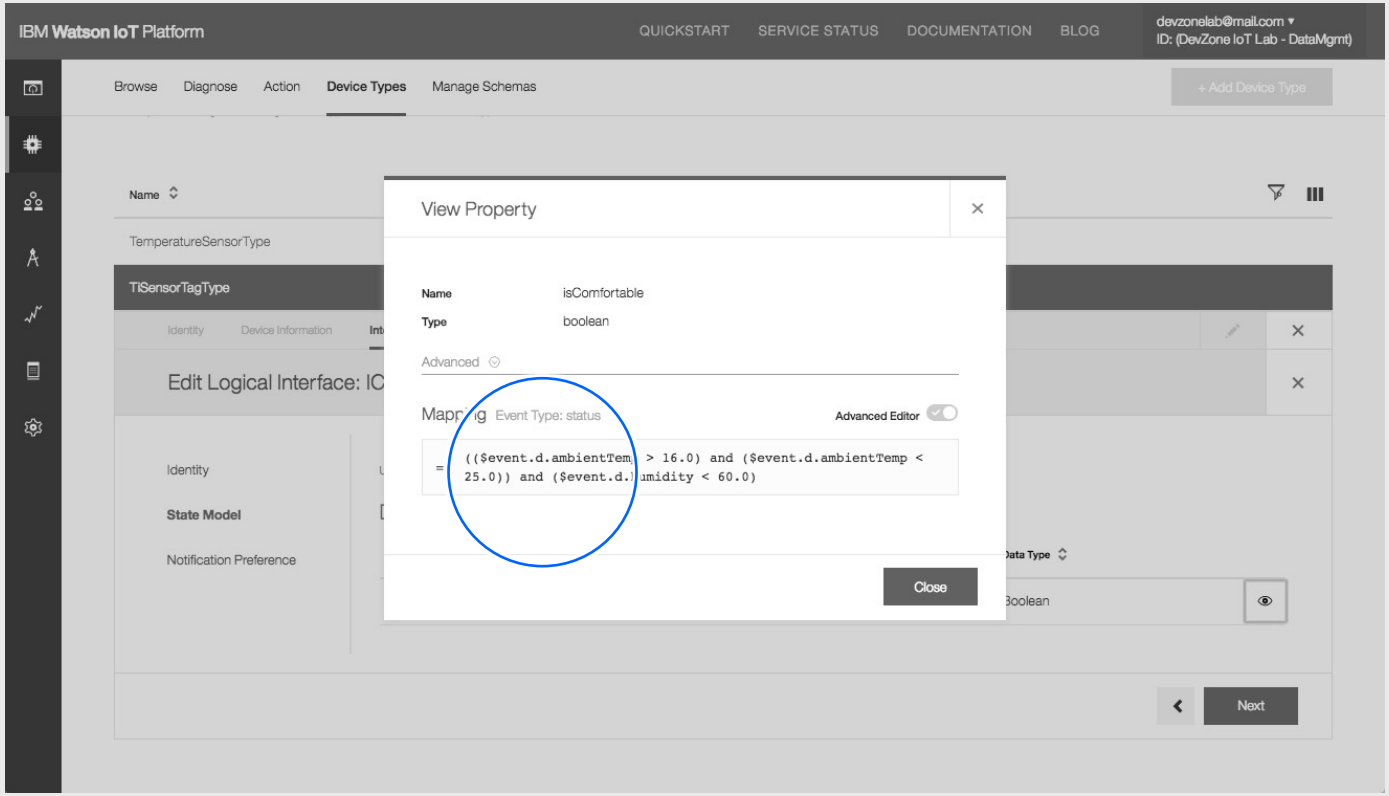
Explore Logical Interfaces

Repeat the steps to view the isComfortable property

The mapping expression for the 'isComfortable' state property is defined by a JSONata expression using the 'advanced' code editor

The expression sets the state to True if the temperature is between 16 and 25 degrees and the humidity is below 60%

Click Next and then Done to return to the Interface tab



4 simple steps > Explore Logical Interfaces

Explore
Logical Interfaces

View the state model and mapping expressions

Conclusions

- A logical interface is an abstraction of a device behavior
- A logical interface declares the state properties
- A device type exposing a logical interface needs to provide mapping statements for the state properties of the interface
- A mapping is a JSONata expression that uses the $event object to reference event data
- A mapping expression may also reference $state of the interface, or device metadata using $instance

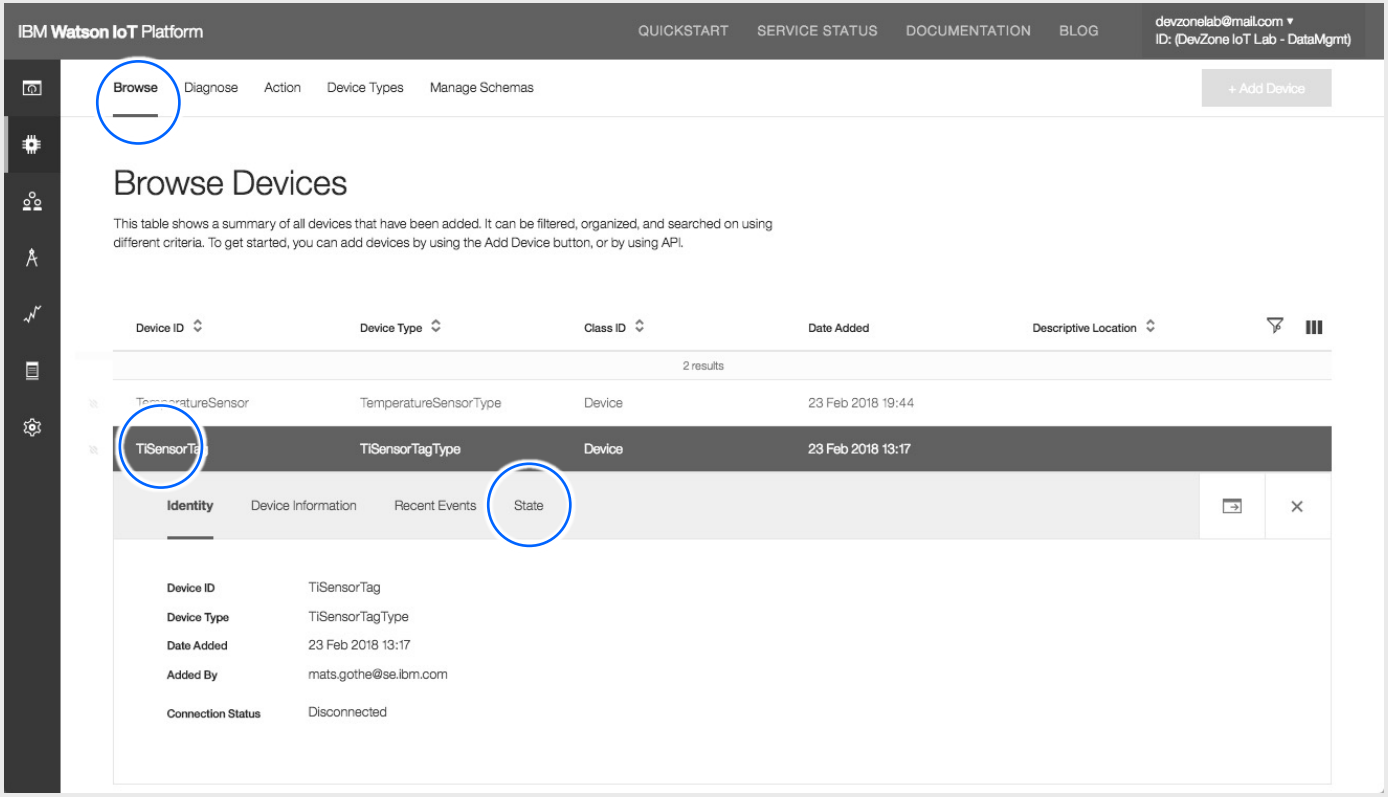
4 simple steps > Explore Device State

Lets explore the Device State

Explore Device State

View the device state of the logical interfaces

1. Return to the device list by clicking on Browse
2. Select the TiSensorTag device
3. Select the State tab

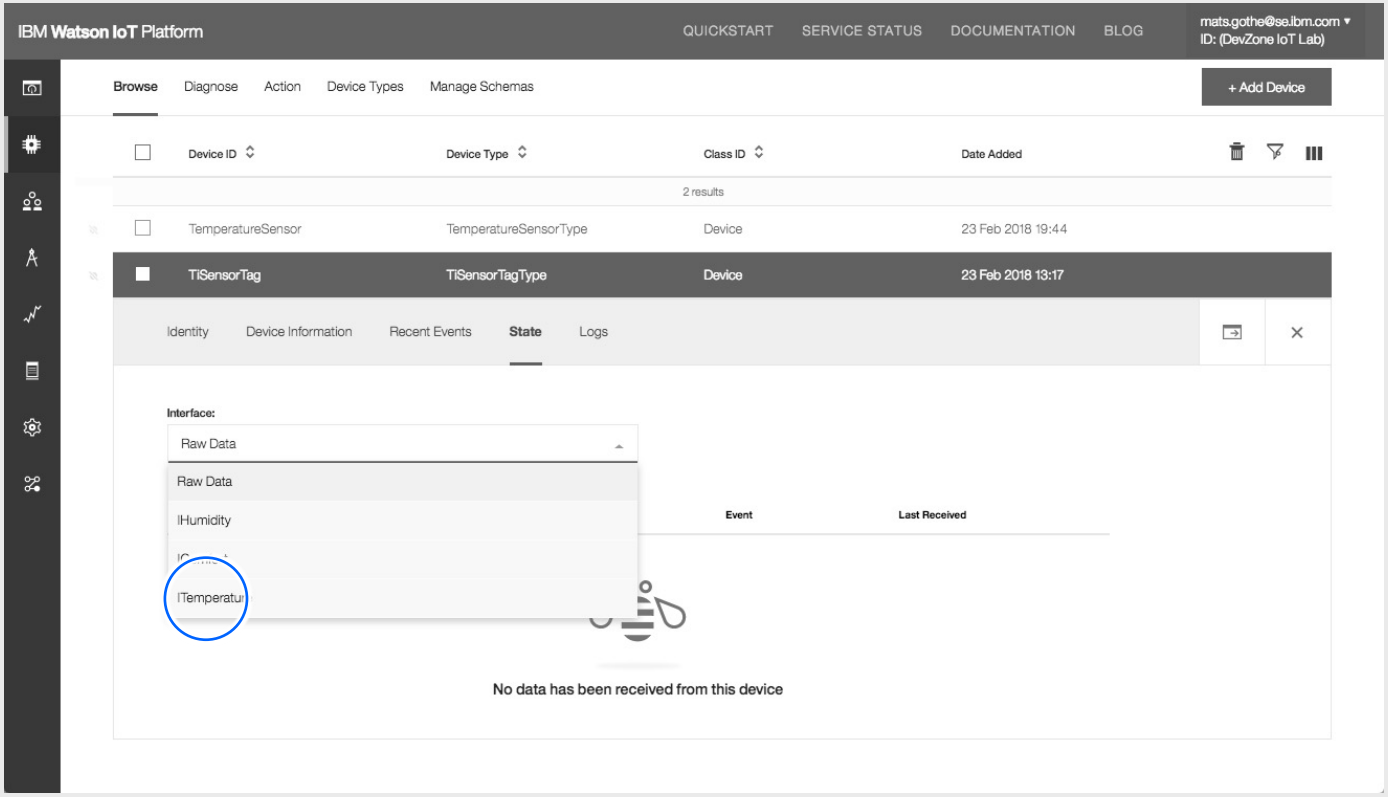


4 simple steps > Explore Device State

Explore Device State

The State pages loads and shows the state of a selected interface

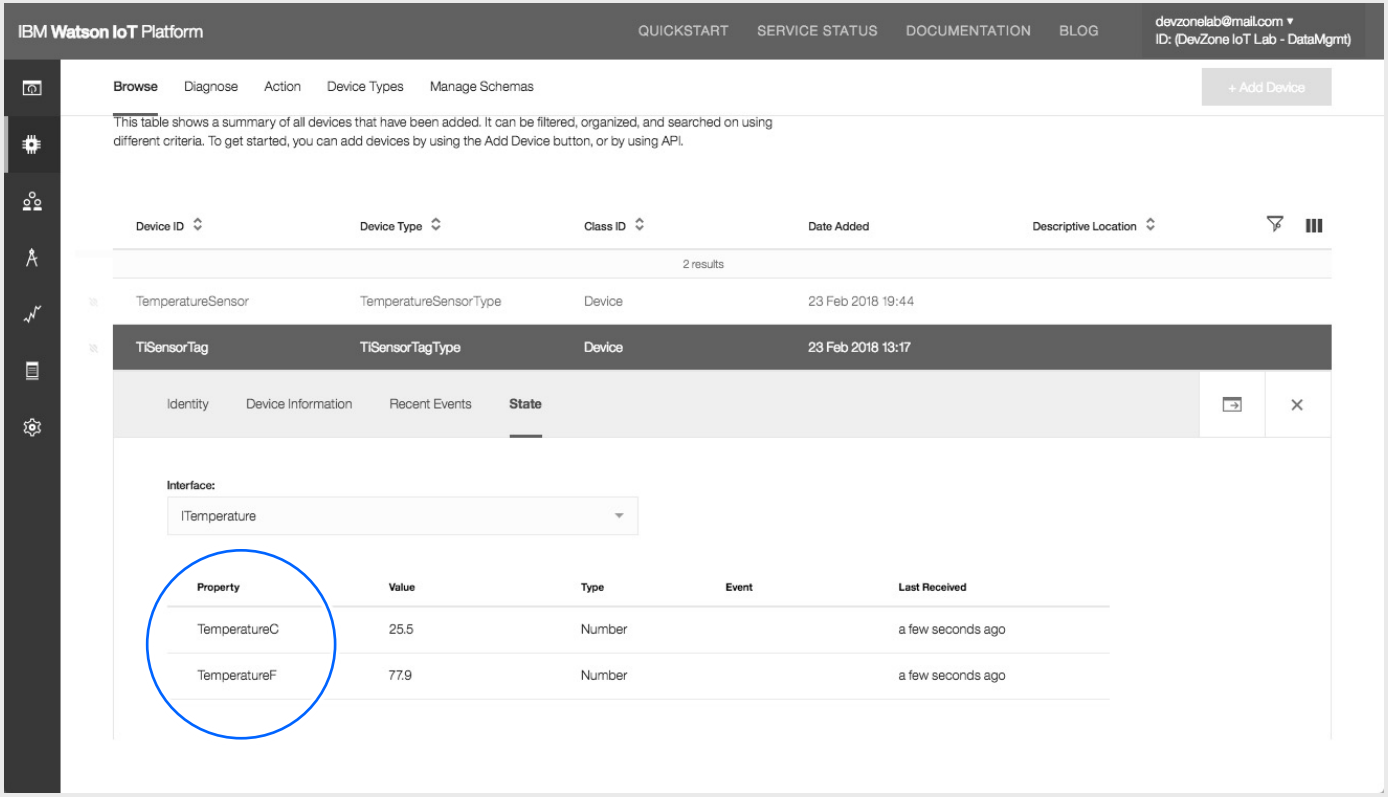
Select ITemperature in the list of interfaces



4 simple steps > Explore Device State

Explore Device State

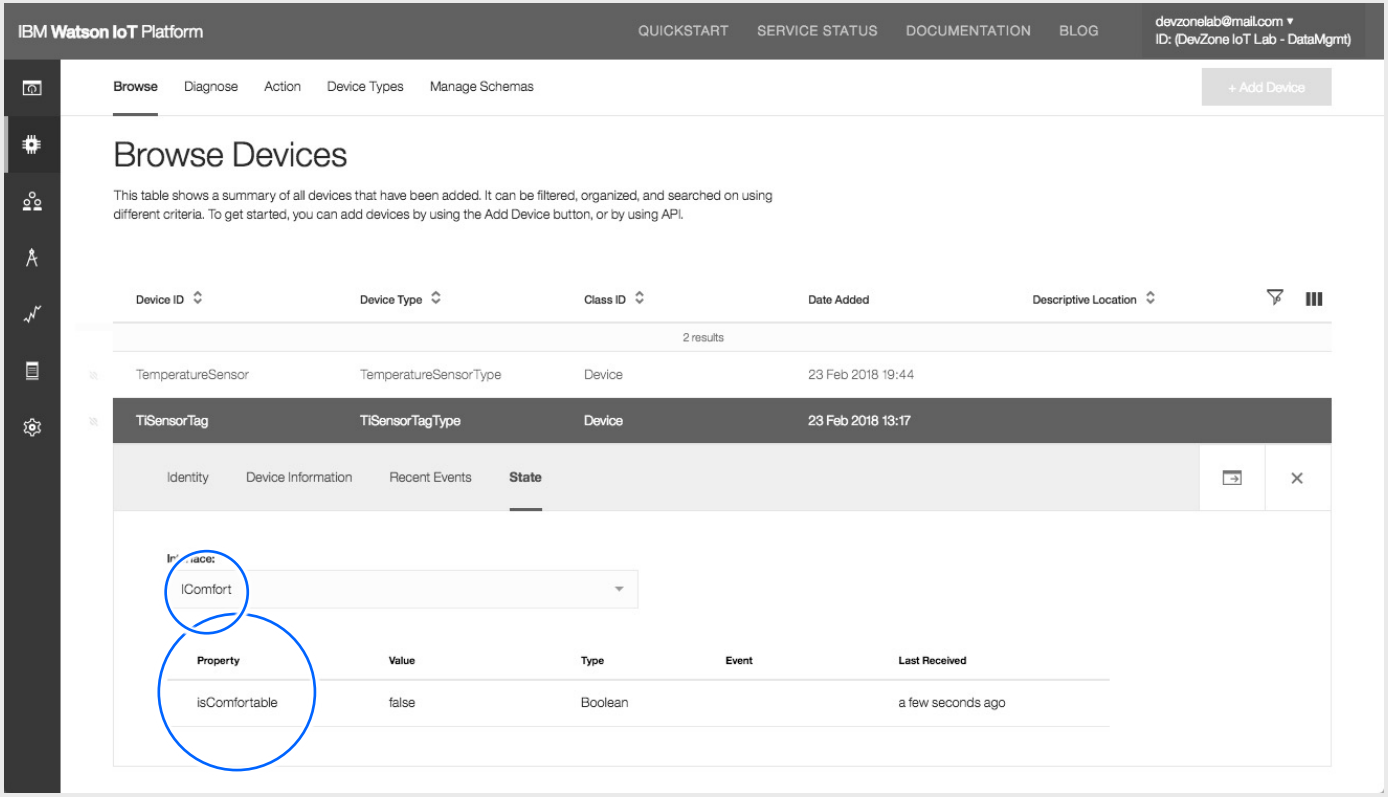
1. Wait for a new event to be received from the device
2. The state updates and shows the current device Celsius temperature and the computed Fahrenheit value
3. Wait for new events to be received and the state to be updated



4 simple steps > Explore Device State

Explore Device State

1. Select the IComfort interface in the list
2. Wait for the next event and view the computed comfort level

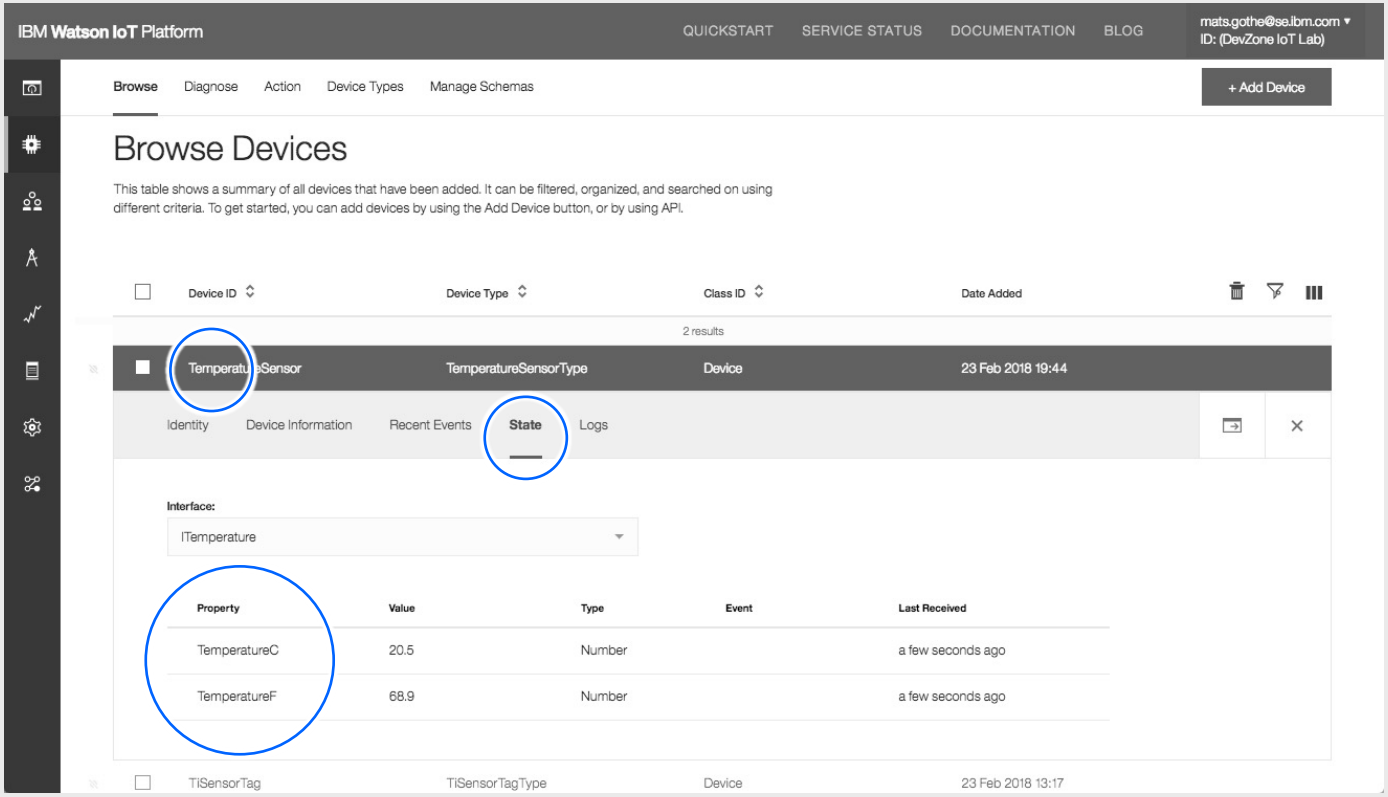


4 simple steps > Explore Device State

Explore Device State

Repeat the steps to view the state of the TemperatureSensor devcie

1. Select the TemperatureSensor devcie in the list
2. Choose the State tab
3. Select the ITemperature interface
4. View the devcie temperature



4 simple steps > Explore Device State

Explore
Device State

View the device state
of the logical interfaces

Conclusions

- Data Management is maintaining the device state of each logical interfaces exposed by the device type
- The device states may be viewed in the IoT platform dashboard or accessed using the IoT platform APIs or CLIs
- Applications may also subscribe to state change notification events from the IoT platform when a device state is updated

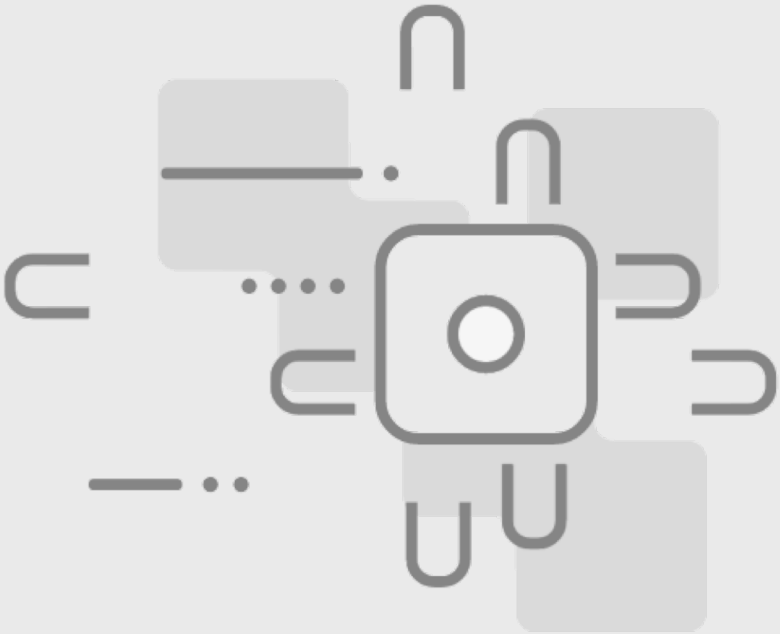
Getting started with Watson IoT Platform Data Management

You have now completed this lab

In this lab you have explored how to get started with the IoT Platform and Data Management

You have

- Learned about Data Management
- Explored device events
- Created a physical interface
- Created a logical interface
- Viewed device state be computed from device events by the IoT Platform



Getting started with Watson IoT Platform Data Management

Learn more about the IoT Platform

- Create your own free IBM Cloud account and explore IoT
- www.ibm.com/iot
- developer.ibm.com/iotplatform/