



Ask Me Anything: Connecting Node-RED Flows to the IBM Watson IoT Platform for an IoT Device **IDA-6963**

Henry Will IV – February 24, 2016





These Slides are from my Breakout Session: **IND-2119:**
“A Primer to Programming an Internet of Things Device
on IBM Bluemix”

Please review the replay of that presentation for more
information.

The last few slides here are reference slides and have
links which may be particularly helpful

InterConnect 2016

The Premier Cloud & Mobile Conference



#ibminterconnect

Agenda

- The Story Behind the Project
- Requirements
- Bluemix Recipes
- Hardware
- Architecture
- MQTT
- IoT Foundation Quickstart
- Creating the Application in Bluemix
- Binding to IoT Foundation
- Node-RED flow

Cabin Fever



#ibminterconnect

Requirements

1. If the temperature goes above a set point (“low set point”) or above a “high set point,” then a notification is sent to the owner (preferably by SMS). The Unit will also continue to report the temperature every hour for 12 hours.
2. The owner may change the set points via an email (via SET command)
3. The owner may also request (via an email) temperature reports to be sent (via CMD command)
4. In case the unit is malfunctioning, the owner may stop all reporting by issuing a command via email (via RPT command)
5. The unit will send the temperature when it first powers up (this will help the owner to know that there was possibly a power loss and power has returned)
6. Every week (in intervals of a week since powering up or the last command received), a report will be sent of the temperature. This will be a reminder that the unit is still working

Recipes

IBM developerWorks Developer Centers

Recipes

Sign in

QUICKLY PRODUCE RECIPES TO INSPIRE MILLIONS OF DEVELOPERS.
[Learn more](#)

Discover

Find recipes created by the community and documents from across the IBM developer network to learn something new.

Combine


Quickly copy and combine existing recipe steps, then sort and edit them to create your own completely new recipe.

Inspire


Creating and sharing recipes inspires others. Get kudos if your recipe is reused and see how your ideas are helping others.

Start your search...


98 recipes in total



IBM Loves

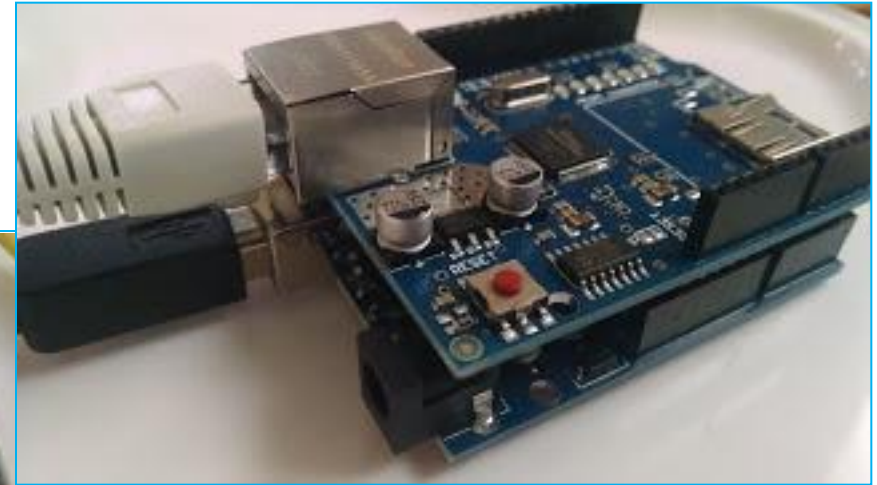


IBM Loves

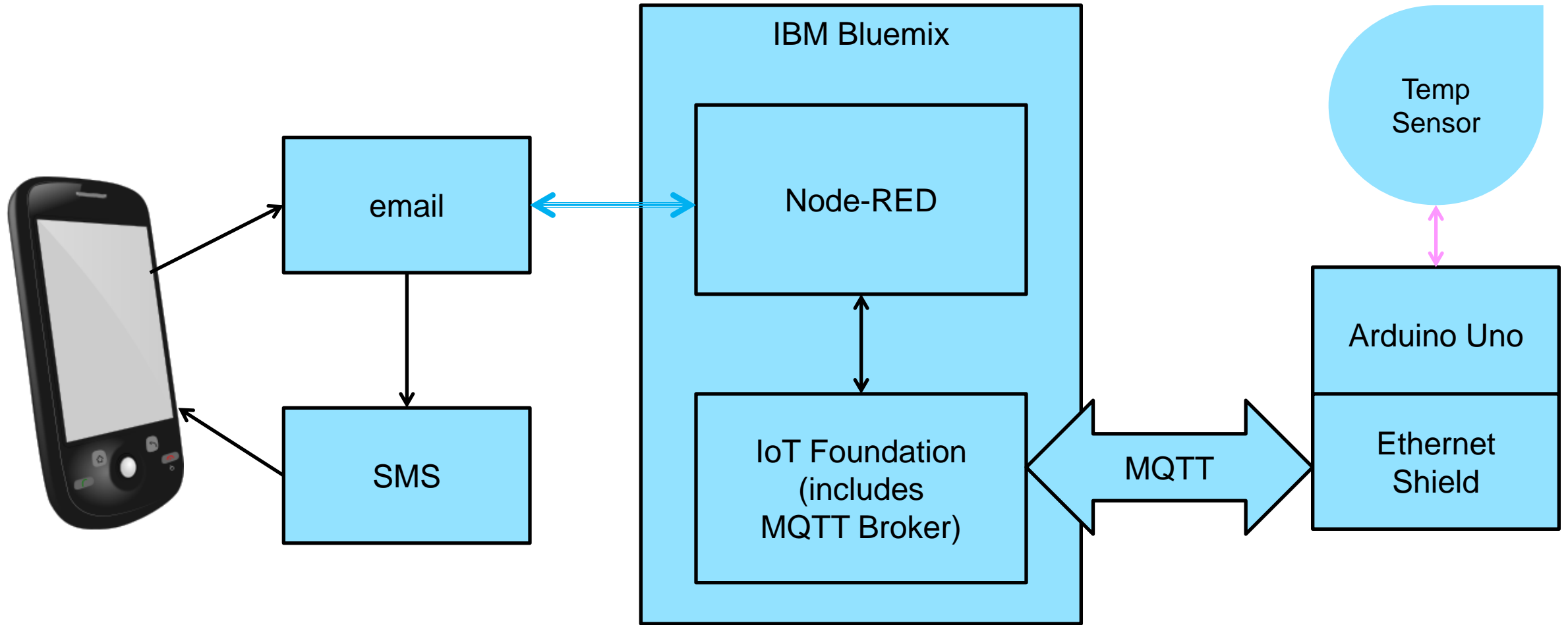


IBM Loves

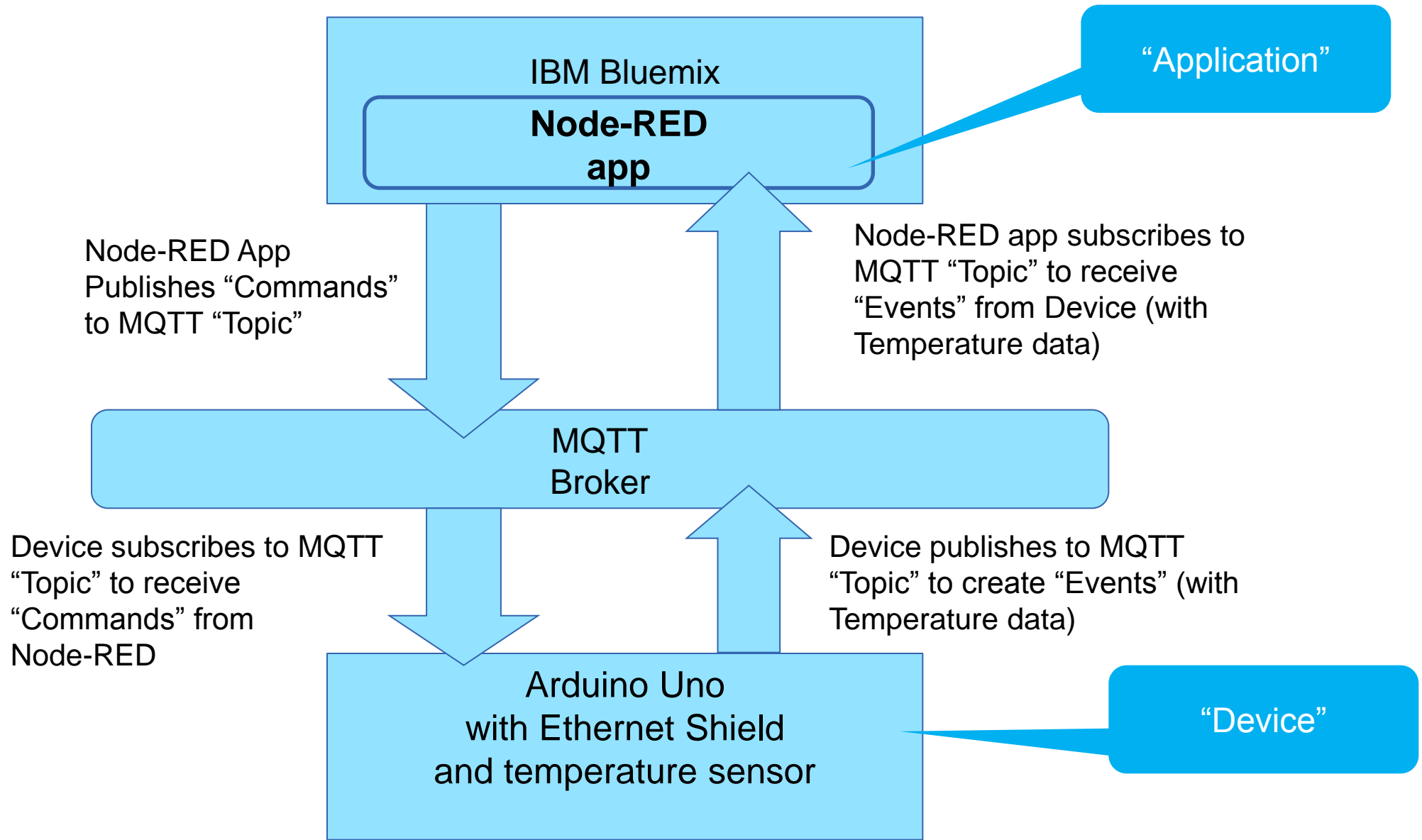
Hardware: Arduino Uno, Ethernet Shield, DS18B20 Temperature Sensor



Architecture




MQTT



IBM IoT Quickstart


[IBM Internet of Things Foundation](#) [Quickstart](#) [Service Status](#) [Documentation](#) [Blog](#) [SIGN IN](#)





Quickstart

No sign-up required to see how easy it is to connect your device to IoT Foundation and view live sensor data

[Go](#)

 Get your device (or simulate one)

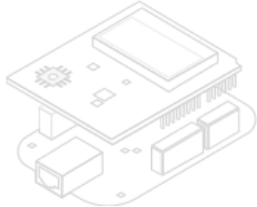
 Follow a recipe to get it connected

 View live data from your device

Got a physical device?

We have a partner program for IoT along with a set of verified instructions, or 'recipes', for connecting devices, sensors, and gateways.


[VIEW RECIPES](#)



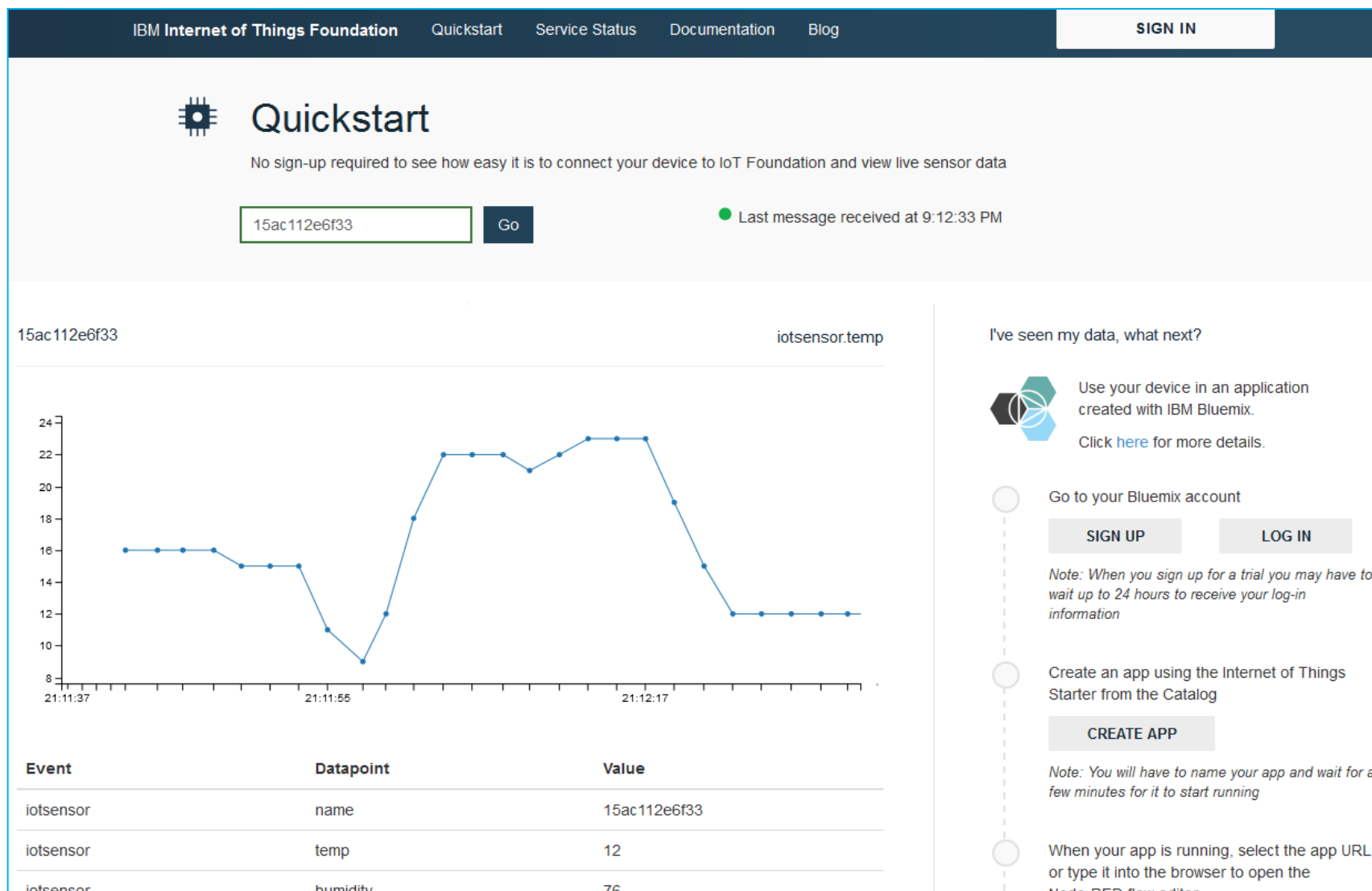
Don't have a device?

You don't need to have a physical device to see Quickstart in action. Try it out by using our simulator.

[FIND OUT MORE](#)




Quickstart Data Graph



Bluemix Catalog: IoT Boilerplate

The screenshot displays the IBM Bluemix Catalog web interface. The browser address bar shows the URL `https://console.ng.bluemix.net/catalog/`. The top navigation bar includes links for DASHBOARD, SOLUTIONS, CATALOG (highlighted with a green circle), PRICING, DOCS, and COMMUNITY. Below the navigation bar, there is a search bar with the placeholder text "Type here to search" and a dropdown menu labeled "ORG:" with the text "your org name here" (highlighted with a green circle). The left sidebar contains a list of categories: Starters, Compute, and Services, each with a sub-list of options. The main content area is titled "Starters // Choose a package of sample code and services, or start from scratch". It features a grid of starter packages. The "Boilerplates" section is highlighted with a green circle and contains the text "Get started with a new app, now". A mouse cursor is hovering over the "Internet of Things Foundation Starter" package, which is also highlighted with a green circle. A tooltip box appears over this package, containing the text: "Internet of Things Foundation Starter: Get started with an Internet of Things Foundation application using Node-RED in Bluemix. Try the sample flow with a simulator and customize it for your own devices." Other visible starter packages include "Web App Starter", "Java DB Web Starter", "Java Workload Scheduler Web Starter", "LoopBack Starter", "Mobile Cloud", "MobileFirst Services Starter", "Node.js Cloudant DB Web Starter", "Personality Insights Java Web Starter", and "Personality Insights Node.js Web Starter".

Creating an App



Internet of Things Foundation Starter


IBM

Get started with an Internet of Things Foundation application using Node-RED in Bluemix. Try the sample flow with a simulator and customize it for your own devices.


VERSION
0.4.19

TYPE
Boilerplate

[VIEW DOCS](#)



SDK for Node.js™



Cloudant NoSQL DB


Develop, deploy, and scale server-side JavaScript® apps with ease. The IBM SDK for Node.js™ provides enhanced performance, security, and serviceability.

[VIEW DOCS](#)

Pick a plan

Monthly prices shown are for country or region: [United States](#)

Plan	Features	
✓ Default	Run one or more apps free for 30 days (375 GB-hours free).	\$0.07 USD/GB-Hour

 This is a service plan for the IBM Bluemix Platform runtime.

[TERMS](#)

Create an app:

Space:
dev

Name:
my-own-nodeRED

Host:
my-own-nodeRED

Domain:
mybluemix.net

Selected Plan:

SDK for Node.js™
Default

Cloudant NoSQL DB
Shared

[CREATE](#)

Staging

The screenshot shows the IBM Bluemix dashboard. The top navigation bar includes links for DASHBOARD, SOLUTIONS, CATALOG, PRICING, DOCS, and COMMUNITY. The left sidebar shows the application 'my-own-nodeRED' with options for Overview, SDK for Node.js™, Files, Logs, Environment Variables, and Start Coding. The main content area displays a message: 'Your application is staging. <http://my-own-nodeRED.mybluemix.net>'. Below this, there is a 'Getting Started with:' section featuring the 'Internet of Things' service, which includes a description: 'Create Internet of Things apps with Node-Red visual editor and the Internet of Things service.' Further down, a section titled 'Start coding with Internet of Things' provides a two-step guide: 1. 'After your application has started, click on the **Routes URL** or enter the following URL in a browser: `http://<yourhost>.mybluemix.net`'. 2. 'Click **Go to your Node-RED flow editor**. You will see a ready-made flow that can process temperature readings from a simulated device.'

Running

The screenshot shows the IBM Bluemix dashboard. The left sidebar contains a navigation menu with the following items: 'Back to Dashboard...', 'my-own-nodeRED', 'Overview', 'SDK for Node.js™', 'Files', 'Logs', 'Environment Variables', 'Start Coding >', 'SERVICES', and 'Cloudant NoSQL DB'. The main content area has a top navigation bar with links to 'DASHBOARD', 'SOLUTIONS', 'CATALOG', 'PRICING', 'DOCS', and 'COMMUNITY'. Below this, a green checkmark icon is followed by the text 'Your app is running. <http://my-own-nodeRED.mybluemix.net>'. A green oval highlights this text, and a green arrow points from it to a text box below. The text box is titled 'Getting Started with:' and contains an 'Internet of Things' section with a gear icon and the text 'Create Internet of Things apps with Node-Red visual editor and the Internet of Things service.' Below this, the heading 'Start coding with Internet of Things' is followed by a numbered list. Step 1 says 'After your application has started, click on the **Routes URL** or enter the following URL in a browser:' and is followed by a dark box containing the URL 'http://<yourhost>.mybluemix.net'. A green oval highlights this URL, and a green arrow points from it to the 'Getting Started with:' text box. Step 2 says 'Click **Go to your Node-RED flow editor**. You will see a ready-made flow that can process temperature readings from a simulated device.'

IBM Bluemix

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY

Back to Dashboard...

my-own-nodeRED

Overview

SDK for Node.js™

Files

Logs

Environment Variables

Start Coding >

SERVICES

Cloudant NoSQL DB

✓ Your app is running. <http://my-own-nodeRED.mybluemix.net>

Getting Started with:

Internet of Things

Create Internet of Things apps with Node-Red visual editor and the Internet of Things service.

Start coding with Internet of Things

1 After your application has started, click on the **Routes URL** or enter the following URL in a browser:

`http://<yourhost>.mybluemix.net`

The Node-RED for Internet of Things landing page displays.

2 Click **Go to your Node-RED flow editor**. You will see a ready-made flow that can process temperature readings from a simulated device.

Internet of Things Foundation Service in Catalog

The screenshot displays the IBM Bluemix Catalog interface. On the left is a dark sidebar with a navigation menu. The main content area has a dark background with a header bar that reads "Services // The building blocks of any great app". Below this, the "Internet of Things" section is featured with the subtitle "A new generation of applications". Three service icons are shown in hexagonal frames: the "Internet of Things Foundation" by IBM (a gear icon, circled in green with a mouse cursor), "IoT Real-Time Insights" by IBM (a network icon), and "flowthings.io" by a Third Party (an "Ft" logo). At the bottom, a section titled "Looking for more?" with a beaker icon encourages users to check out the "Bluemix Labs Catalog" to try out experimental runtimes and services.

Starters

- Boilerplates

Compute

- Runtimes
- Containers

Services

- Watson
- Mobile
- DevOps
- Web and Application
- Network
- Integration
- Data and Analytics
- Security
- Storage
- Business Analytics
- ☒ Internet of Things

Provider

Services // The building blocks of any great app

Internet of Things
A new generation of applications

Internet of Things Foundation
IBM

IoT Real-Time Insights
IBM

flowthings.io
Third Party

Looking for more?
Check out the Bluemix Labs Catalog to try out experimental runtimes and services.
[Bluemix Labs Catalog](#)

Link to IoT Service



Internet of Things Foundation

IBM

PUBLISH DATE
10/26/2015

AUTHOR
IBM

TYPE
Service

LOCATION
US South

[VIEW DOCS](#)

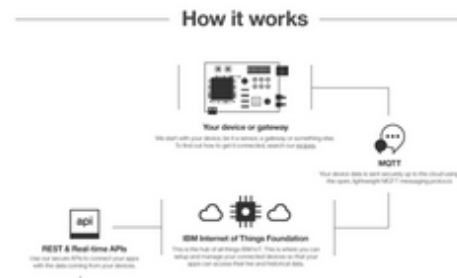
The IBM Internet of Things service lets your apps communicate with and consume data collected by your connected devices, sensors, and gateways. Our recipes make it super easy to get devices connected to our Internet of Things cloud. Your apps can then use our real-time and REST APIs to communicate with your devices and consume the data you've set them up to collect.

- **Connect your devices securely to the cloud**

Before your apps can get to work, you need to get your devices connected up! We have a set of verified instructions, or 'recipes', for connecting devices, sensors and gateways from a variety of partners and individuals.

- **Build an app that talks to your devices**

Communications between your devices and the cloud happen via the open, lightweight MQTT protocol. For example you might have a sensor that collects and sends humidity readings every minute. Our REST and real-time APIs allow you to quickly pull that device data into your apps for further analysis.



Add Service

Space:

dev

App:

my-own-nodeRED my-own-no...

Service name:

IoTf-mnR

Selected Plan:

Free

CREATE

IoT Service

IBM Bluemix

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY

124

ORG: Your Org Name

+ Create a Space

dev

CF APPS (1)

SERVICES (2)

CONTAINERS (0)

VIRTUAL MACHINES (0)

Applications

my-own-nodeRED

my-own-nodeRED.mybluemix.net

.js

Running

Services

IoTf-mnR

Internet of Things Foundation

my-ow...

Plan: iotf-service-free

my-own-nodeRED-cloudantNoS...

Cloudant NoSQL DB

my-ow...

Plan: Shared

Register the Device

IBM Bluemix

DASHBOARD SOLUTIONS CATALOG PRICING DOCS COMMUNITY 124

Back to Dashboard... ioTF-mnR DOCS

ioTF-mnR

Manage >

Service Access Authorization

Plan

APPS USING SERVICE

my-own-nodeRED

Hi! Welcome to the Internet of Things Foundation

Take a look at the steps below to get you going with your Internet of Things app

Connect your devices

Use our [recipes](#) to find out how to add your devices. We work with partners and have sample connection recipes for many devices.

Launch the Internet of Things Foundation dashboard and add your devices by clicking the 'Add Device' button under the 'Devices' tab.

Launch dashboard

Learn how to build your app

When you have added your devices, you can come back to Bluemix to start building your app using your real-time and historical device data.

Read the docs to find out how to make the most out of your app.

[Go to docs](#)

Learn how to extend your app

Use other Bluemix services to extend your app to start creating a great Internet of Things app.

Here are some of the services you could use:

Twilio
Third Party

Cloudant NoSQL DB
IBM

Dash DB
IBM

Geospatial Analytics
IBM

Time Series Database
IBM

IBM Analytics for Hadoop
IBM

Add Device Type

IBM Internet of Things Foundation Quickstart Service Status Documentation Blog

Organization ID: Your Org ID
Bluemix Free

OVERVIEW DEVICES ACCESS USAGE

Devices

DEVICES
0
Devices registered

There are no devices in your organization
[Click here to add new devices](#)

+ Add a device

Add Device

Choose Device Type

Choose Device Type

Or

Create device type

Create Type

Create device type

General Information

Name MyDevice

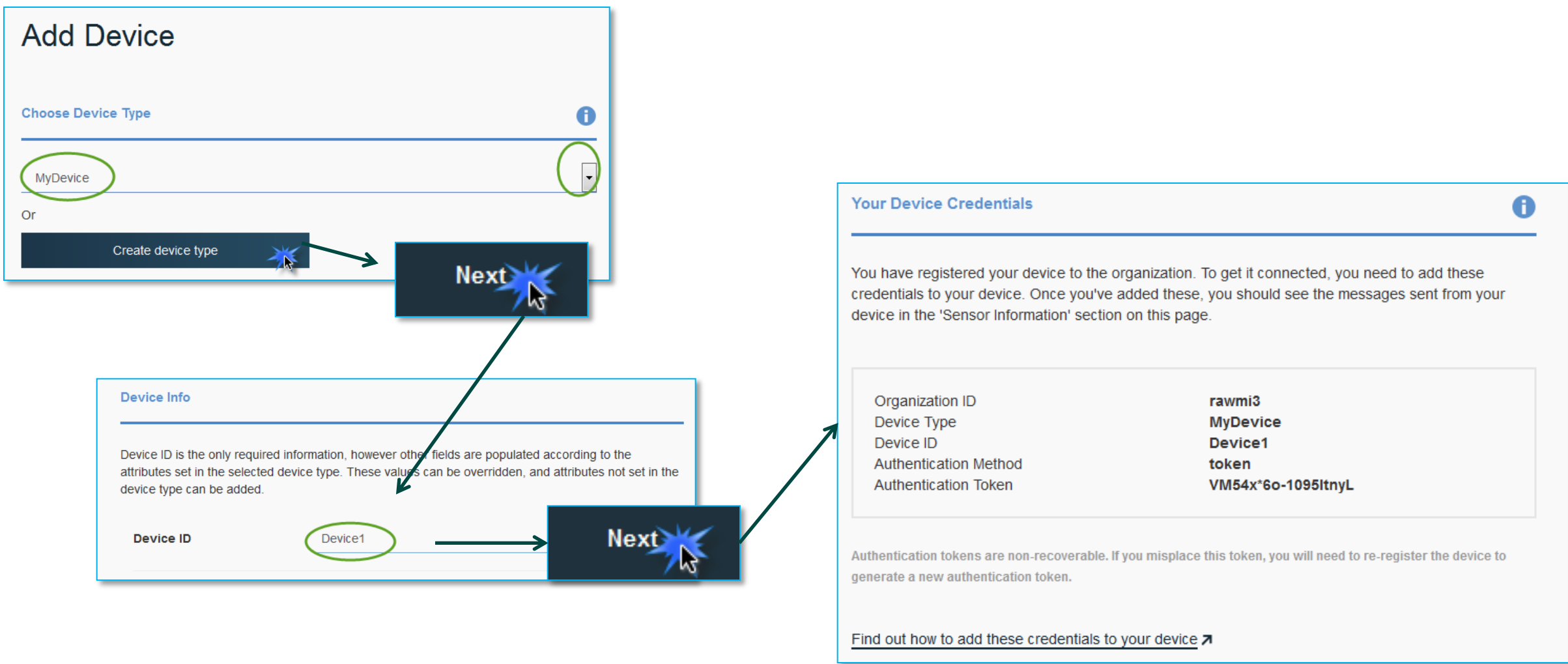
The device type name is used to identify the device type uniquely, using a restricted set of characters to make it suitable for API use.

Description my Hardware Device

The device type description can be used for a more descriptive way of identifying the device

Back Next

Add Device (and Registration)



Add Yourself to the Organization (Important)

The screenshot displays the IBM Internet of Things Foundation console interface. At the top, the navigation bar includes links for 'IBM Internet of Things Foundation', 'Quickstart', 'Service Status', 'Documentation', and 'Blog'. Below this, the 'Organization ID' is shown as 'Your Org ID' with a green border. The main navigation tabs are 'OVERVIEW', 'DEVICES', 'ACCESS', and 'USAGE'. The 'ACCESS' tab is currently selected, and its sub-menu is open, showing 'Members', 'Guests', 'API Keys', and 'Bluemix Apps'. The 'Members' sub-tab is highlighted. The main content area shows a message: 'THERE ARE NO MEMBERS IN YOUR ORGANIZATION. Your organization has only guest members at present. You may promote guests to full membership or add new members.' Below this message is a prominent 'Add Members' button, which is being clicked by a mouse cursor. On the left side of the console, a 'Devices' panel is visible, showing '1 Devices registered' and a list item 'MyDevice 1' with a red status indicator.

Bluemix Node-RED

The screenshot shows a web browser window with the address bar containing `cabinfever.mybluemix.net`. The page title is "Node-RED in Bluemix for IBM Internet of Things Foundation". The main heading is "Node-RED in Bluemix" with the subtitle "A visual tool for wiring the Internet of Things" and "IBM Internet of Things Foundation". Below this is a diagram of a Node-RED flow with four nodes: an input node, a join node, and two output nodes. A red button labeled "Go to your Node-RED flow editor" is highlighted with a green oval. Below the button is a link "Learn how to customise Node-RED".

Node-RED in Bluemix for IBM Internet of Things Foundation

Node-RED in Bluemix

A visual tool for wiring the Internet of Things

IBM Internet of Things Foundation

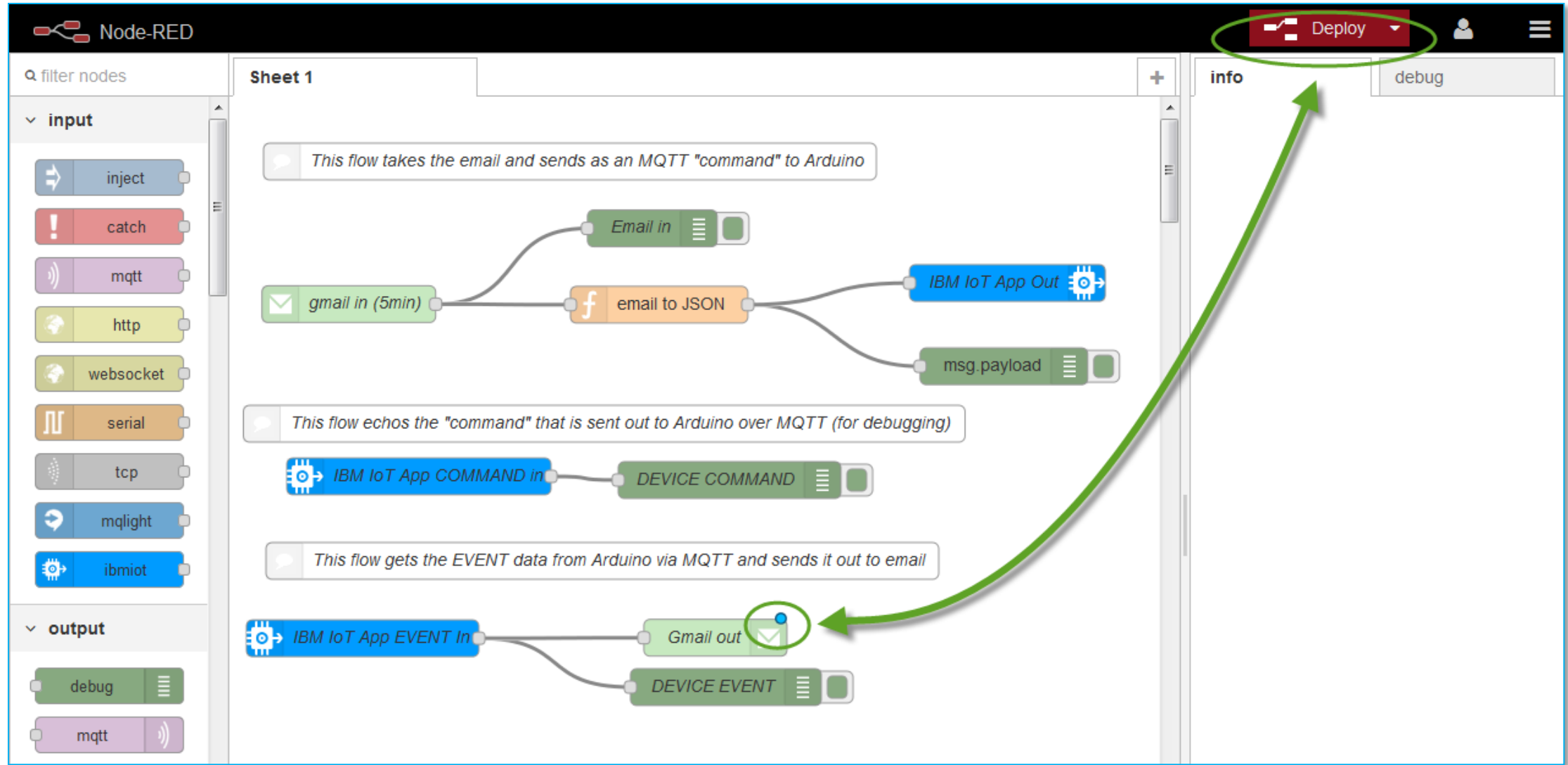
Node-RED provides a browser-based editor that makes it easy to wire together flows that can be deployed to the runtime in a single click.

The version running here has been customized for the IBM Internet of Things Foundation.

[Go to your Node-RED flow editor](#)

[Learn how to customise Node-RED](#)

Node-RED flow



“IBM IoT-out Node” Used to Send Command to Device

Edit ibmiot out node

Authentication:

Output Type:

Device Type:

Device Id:

Command Type:

Format:

Data:

Name:

Note: If there is a property in the message that corresponds to any of the values entered above, then the property in the message takes precedence. See the Info tab for more details.
Example JSON device event: `{"d":{"myName":"Arduino Uno", "temperature":989}}`

Node Info:

Node	
Name	IBM IoT App Out
Type	ibmiot out
ID	

Properties

Output node that can be used with the IBM Internet of Things Foundation to send a commands to a device or send an event on the behalf of a device

The following message properties take precedence and override the values configured in the node:

- msg.deviceId** overrides the value of "Device Id"
- msg.deviceType** overrides the value of "Device Type"
- msg.eventOrCommandType** overrides the value of "Event Type" or "Command Type"
- msg.format** overrides the value of "Format"
- msg.payload** overrides the value of "Data"

Note: Some of the functions of this node are not available when it is running in **Quickstart Mode**.

"IBM IoT-in Node" Used to Echo Command to Device

The screenshot shows the Node-RED web interface. In the center, a flow is being edited on 'Sheet 1'. The flow starts with an 'Email in' node, followed by a 'gmail in (5min)' node, and then the 'IBM IoT App COMMAND in' node. A green arrow points from the 'Email in' node to the 'IBM IoT App COMMAND in' node. Below the flow, there is a note: 'This flow gets the EVENT data...'. The 'IBM IoT App COMMAND in' node is highlighted with a blue border. To the right of the flow, the 'Edit ibmiot in node' dialog box is open, showing the configuration for the 'IBM IoT App COMMAND in' node. The dialog box has tabs for 'Authentication', 'Input Type', 'Device Type', 'Device Id', 'Command', 'Format', and 'Name'. The 'Authentication' tab is selected, showing 'Bluemix Service' as the authentication method. The 'Input Type' is set to 'Device Command'. The 'Device Type' is set to 'All or Arduino-tempsensor'. The 'Device Id' is set to '2C4FF659DCEA'. The 'Command' is set to 'All or status'. The 'Format' is set to 'All or json'. The 'Name' is set to 'IBM IoT App COMMAND in'. Below the configuration fields, there is a note: 'Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications. Check the info tab, to get more information about each of the fields'. The 'info' tab on the right side of the interface shows the node's properties: Name: IBM IoT App COMMAND in, Type: ibmiot in, ID: [empty]. The 'debug' tab is also visible.

Edit ibmiot in node

- Authentication: Bluemix Service
- Input Type: Device Command
- Device Type: All or Arduino-tempsensor
- Device Id: All or 2C4FF659DCEA
- Command: All or status
- Format: All or json
- Name: IBM IoT App COMMAND in

Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications. Check the info tab, to get more information about each of the fields

Ok Cancel

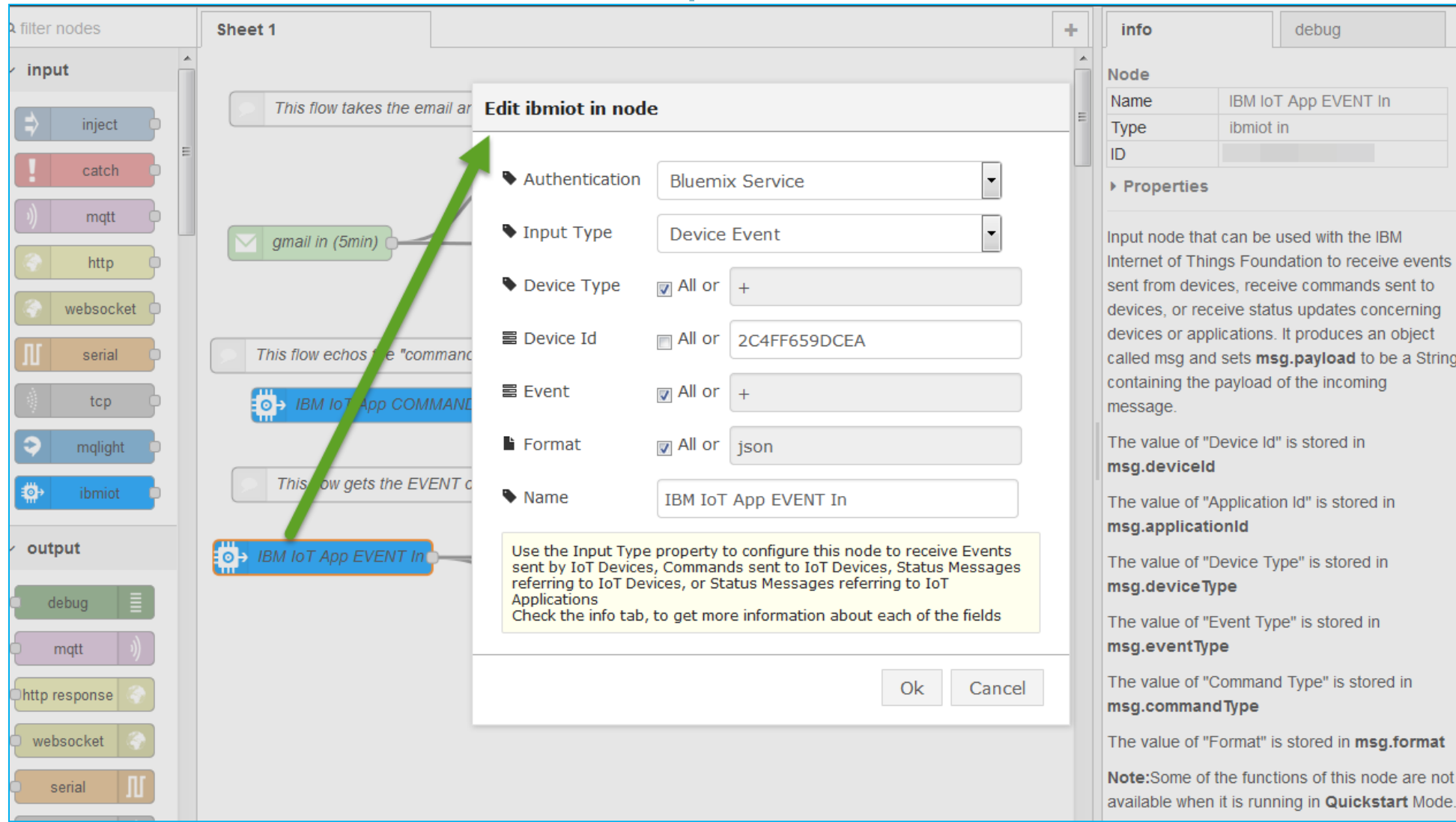
Authentication: Bluemix Service

Quickstart
API Key
Bluemix Service

Input Type: Device Command

Device Event
Device Command
Device Status
Application Status

“IBM IoT-in Node” Used to Input Device Event



The screenshot shows the Node-RED interface with the 'Edit ibmiot in node' dialog box open. A green arrow points from the 'ibmiot' node in the left sidebar to the dialog. The dialog has the following fields:

- Authentication: Bluemix Service
- Input Type: Device Event
- Device Type: All or +
- Device Id: All or 2C4FF659DCEA
- Event: All or +
- Format: All or json
- Name: IBM IoT App EVENT In

A yellow note box at the bottom of the dialog reads: "Use the Input Type property to configure this node to receive Events sent by IoT Devices, Commands sent to IoT Devices, Status Messages referring to IoT Devices, or Status Messages referring to IoT Applications. Check the info tab, to get more information about each of the fields".

The right sidebar shows the 'info' tab for the 'Node' with the following details:

Node	
Name	IBM IoT App EVENT In
Type	ibmiot in
ID	

Properties

Input node that can be used with the IBM Internet of Things Foundation to receive events sent from devices, receive commands sent to devices, or receive status updates concerning devices or applications. It produces an object called **msg** and sets **msg.payload** to be a String containing the payload of the incoming message.

The value of "Device Id" is stored in **msg.deviceId**

The value of "Application Id" is stored in **msg.applicationId**

The value of "Device Type" is stored in **msg.deviceType**

The value of "Event Type" is stored in **msg.eventType**

The value of "Command Type" is stored in **msg.commandType**

The value of "Format" is stored in **msg.format**

Note: Some of the functions of this node are not available when it is running in **Quickstart** Mode.

Converting eMail Command to JSON

Sheet 1

This flow takes the email and sends as an MQTT "command" to Arduino

Email in

gmail in (5min)

email to JSON

IBM IoT App Out

Edit function node

Name: email to JSON

Function:

```
1 var newone2 = "tst";
2 //newone2 = msg.payload.split(",")[0];
3
4
5 newone2 = msg.payload.substring(0, 3);
6
```

Outputs: 1

See the Info tab for help writing functions.

Ok Cancel

info debug

Node

Name	email to JSON
Type	function
ID	

Properties

A function block where you can write code to do more interesting things.

The message is passed in as a JavaScript object called `msg`.

By convention it will have a `msg.payload` property containing the body of the message.

Logging and Error Handling

To log any information, or report an error, the following functions are available:

- `node.log("Log")`
- `node.warn("Warning")`
- `node.error("Error")`

The Catch node can also be used to handle errors. To invoke a Catch node, pass `msg` as a second argument to `node.error`:

```
node.error("Error", msg)
```

Debugging

The screenshot shows the Node-RED web interface. The top bar includes the Node-RED logo and a 'Deploy' button. The left sidebar contains a 'filter nodes' search bar and two categories: 'input' and 'output'. The 'input' category includes nodes like inject, catch, mqtt, http, websocket, serial, tcp, mqlight, and ibmiot. The 'output' category includes a 'debug' node, which is circled in green. The main workspace, labeled 'Sheet 1', contains a flow with three comment boxes: 'This flow takes the email and sends as an MQTT "command" to Arduino', 'This flow echos the "command" that is sent out to Arduino over MQTT (for debugging)', and 'This flow gets the EVENT data from Arduino via MQTT and sends it out to email'. The flow starts with a 'gmail in (5min)' node connected to an 'email to JSON' node. The 'email to JSON' node has two outputs: one to an 'Email in' node (circled in green) and another to a 'msg.payload' node. The 'Email in' node is connected to an 'IBM IoT App Out' node. The 'msg.payload' node is also connected to the 'IBM IoT App Out' node. The 'IBM IoT App Out' node is connected to a 'DEVICE COMMAND' node. The 'DEVICE COMMAND' node is connected to a 'Gmail out' node. The 'Gmail out' node is connected to a 'DEVICE EVENT' node. The 'DEVICE EVENT' node is connected to a 'Gmail out' node. The right sidebar has tabs for 'info' and 'debug'. The 'debug' tab is active, showing a list of messages. Four green arrows point from the 'Email in' node, the 'IBM IoT App Out' node, the 'DEVICE COMMAND' node, and the 'DEVICE EVENT' node to the 'debug' tab. The messages in the 'debug' tab are as follows:

- 1/12/2016, 5:23:47 PM Email in
Test : msg.payload : string [29]
CMD2,5 sent from my moto x
- 1/12/2016, 5:23:47 PM b384b11d.46cc3
msg.payload : string [69]
{ "d": { "myName": "NodeRed CF", "mycmd": "CMD", "myparama": 2, "myparamb": 5 } }
- 1/12/2016, 5:23:47 PM DEVICE COMMAND
iot-2/type/Arduino-tempsensor/id/2C4FF659DCEA/cmd/status
/fmt/json : msg.payload : object
{ "d": { "myName": "NodeRed CF", "mycmd": "CMD", "myparama": 2, "myparamb": 5 } }
- 1/12/2016, 5:24:05 PM DEVICE EVENT
iot-2/type/Arduino-tempsensor/id/2C4FF659DCEA/evt/status
/fmt/json : msg.payload : object
{ "d": { "myName": "Arduino CF", "mymsg": "CMD", "tempr": 64, "lowpt": 40, "hipt": 100 } }
- 1/12/2016, 5:29:08 PM DEVICE EVENT
iot-2/type/Arduino-tempsensor/id/2C4FF659DCEA/evt/status
/fmt/json : msg.payload : object
{ "d": { "myName": "Arduino CF", "mymsg": "CMD", "tempr": 64, "lowpt": 40, "hipt": 100 } }

#ibminterconnect

Insights for Weather



Insights for Weather

IBM

PUBLISH DATE
10/22/2015

AUTHOR
IBM

TYPE
Service

LOCATION
US South

[VIEW DOCS](#)

This service lets you integrate historical and real-time weather data from The Weather Company into your IBM Bluemix application. You can retrieve weather data for an area specified by a geolocation. The data allows you to forecast, detect, and visualize disruptive weather events that might affect decision making in your application.

- **Standard 1-Day Hourly**

An hourly weather forecast for the next 24 hours starting from the current time, for a specified geolocation. This forecast includes the current conditions with appropriate units of measure for the location and in the language requested.

- **Standard 10-Day Daily**

A daily forecast for each of the next 10 days starting from the current day, including forecasts for the daytime and nighttime segments. This forecast includes the forecast narrative text string of up to 256 characters with appropriate units of measure for the location and in the language requested.

- **Current observations**

Observed weather data (temperature, wind direction and speed, humidity, pressure, dew point, visibility, and UV Index) that is collected from observation devices worldwide, plus a weather phrase and a matching weather icon.

- **Historical data**

Observed weather data from site-based observation stations for a specified geolocation that includes current observations and up to 24 hours of past observations.



Begin composing your service with

Insights for Weather

[LOG IN TO BLUEMIX](#)

Don't have an account?

[SIGN UP FOR A FREE TRIAL](#)

IoT Real-Time Insights



IoT Real-Time Insights

IBM

PUBLISH DATE
12/12/2015

AUTHOR
IBM

TYPE
Service

[VIEW DOCS](#)

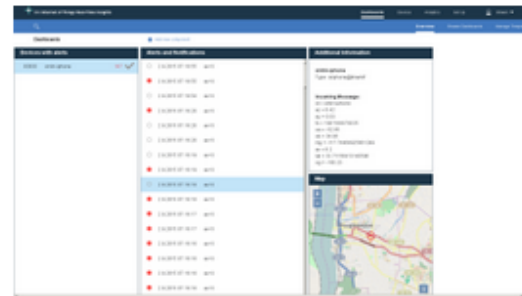
The IBM IoT Real-Time Insights services allows you to understand IoT data in context and monitor the conditions of your devices and operations. IoT Real-Time Insights works with Internet of Things Foundation to enrich and monitor data from you devices, visualize what's happening now, and respond to emerging conditions through automated actions.

- **Contextualize IoT data**

Apply a virtual data model which combines IoT data with asset master data to put data in context and gain greater insight. For example, incoming data can be merged with master asset data from Maximo Asset Management for a complete view of your assets.

- **Apply rules and take automated action**

Using the built in rules engine, monitor enriched data to detect when a device or process is not performing properly, and then automate the appropriate response to ensure that business operations are not disrupted. For example, you could automatically create a work order to track an issue in Maximo and respond with a service technician.



<http://www.ng.bluemix.net/docs/services/iotrtinsights/index.html>

Take aways

Let's take a moment to think of
one take-away you have from
this presentation

Also, please submit a survey!

How to Contact me

Please reach out and Connect to me!
(tell me why you want to connect)

- Blog: <http://www.henrywill4.com>
- Twitter: @henrywill and @pmlessons
- International Project Management Group
(Founded by Mr. Will):
<http://www.pmlessonslearned.com>
- Blog post about this project
<http://henrywill4.blogspot.com/2016/02/cabin-fever-part-6-smart-temperature.html>
- You'll find me updating that post with more info during this InterConnect 2016 conference



For More In-depth discussion...

Stop by and see me at my “Ask Me Anything” Session

- IDA-6963 - Connecting Node-RED Flows to the IBM Watson IoT Platform for an IoT Device
- Wednesday Feb. 24, 12:00 PM to 2:00 PM
- Venue : Mandalay Bay Solution EXPO
- Room : dev@interconnect - Ask Me Anything Station 1
- Please also review the replay of my Breakout Session:
“**IND-2119** : A Primer to Programming an Internet of Things Device on IBM Bluemix”





Reference Slides are
located after these Legal
Slides

InterConnect 2016
The Premier Cloud & Mobile Conference



#ibminterconnect

Notices and Disclaimers

Copyright © 2016 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law

Notices and Disclaimers Con't.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: www.ibm.com/legal/copytrade.shtml.



Thank You

Your Feedback is Important!

Access the InterConnect 2016 Conference Attendee Portal to complete your session surveys from your smartphone, laptop or conference kiosk.



Ask Me Anything:
Connecting Node-RED Flows to the IBM
Watson IoT Platform for an IoT Device
IDA-6963

The Benefits of IBM IoT

- Web Dashboard
- Cloud hosted
- Device Management: Reboot, update firmware, device diagnostics, bulk device addition and removal
- Responsive, Scalable Connectivity
- Secure communication (MQTT and TLS)
- Storage and access to data
- Ties to other Bluemix services (including Watson)

From <https://internetofthings.ibmcloud.com/#/>

IoT Game Using MQTT

Load this URL on your Mobile Device to play:

<http://iotgame.mybluemix.net/>

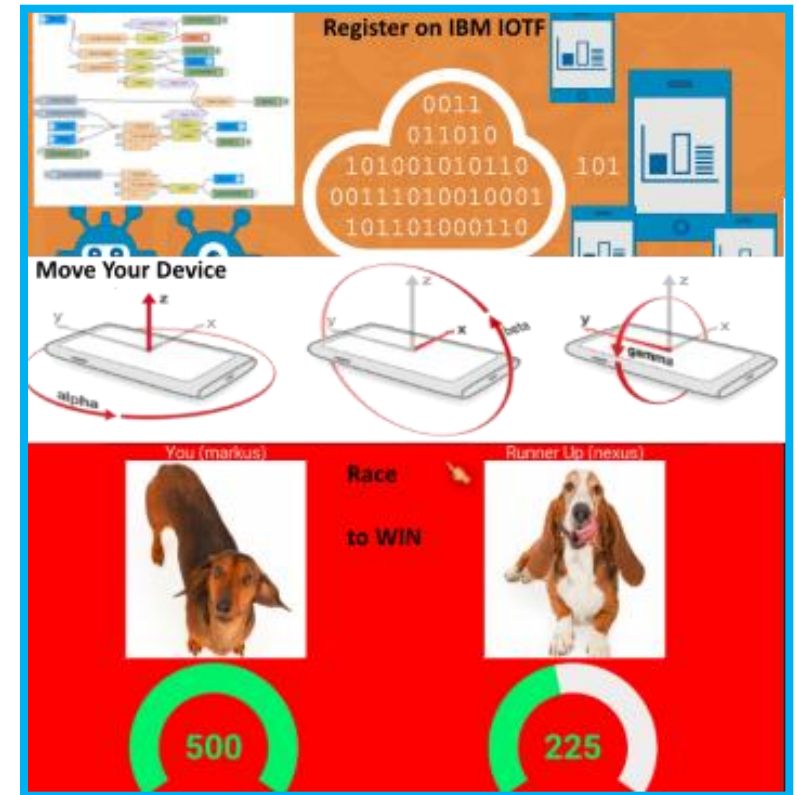
Dashboard:

<http://decodenr.mybluemix.net/freeboard/#start-34807>

Video about the technology:

<https://www.youtube.com/watch?v=sTfmU-UwXmU>

Thanks for Markus Van Kempen for this game





IND-2119 A Primer: Programming an IoT Device on IBM Bluemix

Code Example



Code Example: Page 1

```
ESP8266_NodeMCU-2016-02-16-Henry-Will | Arduino 1.6.7
File Edit Sketch Tools Help

ESP8266_NodeMCU-2016-02-16-Henry-Will $ mydefs.h myssid.h

1 /*
2 This code was taken from a Pubsub Example at:
3 https://github.com/knolleary/pubsubclient/tree/master/examples/mqtt_esp8266
4 Copyright (c) 2016 Nick O'Leary with this MIT License: Permission is hereby granted, free of charge, to any person obtaining a copy of this
5 software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights
6 to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is
7 furnished to do so, subject to the following conditions:
8 The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.
9 THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
10 FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
11 LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.
12 https://opensource.org/licenses/mit-license.php
13
14 This code was modified by Henry Will under the same license 2016-02-16
15
16 Basic ESP8266 MQTT example
17 This sketch demonstrates the capabilities of the pubsub library in combination with the ESP8266 board/library.
18
19 It connects to wifi and then to the IBM Bluemix IoT Foundation MQTT Server then:
20 - publishes a JSON message to the topic every two seconds
21 - subscribes to the command topic, printing out any messages it receives. NB - it assumes the received payloads are strings not binary
22 - If the first character of the command topic is an 1, switch ON the ESP Led,
23   else switch it off
24
25 It will reconnect to the server if the connection is lost using a blocking
26 reconnect function. See the 'mqtt_reconnect_nonblocking' example for how to
27 achieve the same result without blocking the main loop.
28
29 To install the ESP8266 board, (using Arduino 1.6.4+):
30 - Add the following 3rd party board manager under "File -> Preferences -> Additional Boards Manager URLs":
31   http://arduino.esp8266.com/stable/package_esp8266com_index.json
32 - Open the "Tools -> Board -> Board Manager" and click install for the ESP8266"
33 - Select your ESP8266 in "Tools -> Board"
34 - you will also need to add this USB driver for windows machines and reboot after installing it:
35 https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx
36 And, you need to add the pubsub library from here:
37 https://github.com/knolleary/pubsubclient/tree/master/src
38
39 More info at my blog here:
40 http://henrywill4.blogspot.com/2016/02/the-esp8266-nodemcu-using-arduino-ide.html
41 */
42
```


Code Example: Page 2

```
42
43 #include <ESP8266WiFi.h>
44 #include <PubSubClient.h>
45
46 // Explicit Function Prototypes (to prevent compiler errors
47 void setup_wifi();
48 void callback(char* topic, byte* payload, unsigned int length);
49 void reconnect();
50
51
52 #include "mydefs.h"
53 //mydefs.h includes the following:
54 // Update these with values suitable for IoTf org
55 //const char* mqtt_server = "myorgid.messaging.internetofthings.ibmcloud.com"; // change myorgid to the one you have in IoTf
56 // char* myclient = "d:<<orgid>>:<<Deviceid>>"; // where <<orgid>> is the orgid (do not include << and >>) and
57 //    and DeviceID is the Device ID that was registered (don not include << >>)
58 // char* myclientpwd = "zzzzzzzz"; // Authentication Token provided at time of registration fo this device
59 // for details, see https://docs.internetofthings.ibmcloud.com/devices/mqtt.html
60
61 #include "myssid.h"
62 //myssid.h includes the following:
63 // Update these with values suitable for your wifi network
64 //const char* ssid = "myssid";
65 //const char* password = "my pwd";
66
67
68 WiFiClient espClient;
69 PubSubClient client(espClient);
70 long lastMsg = 0;
71 char msg[50];
72 int value = 30;
73
```

Code Example: Page 3

```
73
74 void setup() {
75     pinMode(BUILTIN_LED, OUTPUT);    // Initialize the BUILTIN_LED pin as an output
76     Serial.begin(115200);
77     setup_wifi();
78
79     client.setServer(mqtt_server, 1883);
80     client.setCallback(callback);
81 }
82
83 void setup_wifi() {
84     delay(10);
85     // We start by connecting to a WiFi network
86     Serial.println();
87     Serial.print("Connecting to ");
88     Serial.println(ssid);
89
90     WiFi.begin(ssid, password);
91
92     while (WiFi.status() != WL_CONNECTED) {
93         delay(500);
94         Serial.print(".");
95     }
96
97     Serial.println("");
98     Serial.println("WiFi connected");
99     Serial.println("IP address: ");
100    Serial.println(WiFi.localIP());
101    delay(5000);
102
103 }
```

Code Example: Page 4

```
104
105 void callback(char* topic, byte* payload, unsigned int length) {
106     Serial.print("Message arrived [");
107     Serial.print(topic);
108     Serial.print("] ");
109     for (int i = 0; i < length; i++) {
110         Serial.print((char)payload[i]);
111     }
112     Serial.println();
113
114     // Switch on the LED if an 1 was received as first character
115     if ((char)payload[0] == '1') {
116         digitalWrite(BUILTIN_LED, LOW); // Turn the LED on (Note that LOW is the voltage level
117         // but actually the LED is on; this is because
118         // it is active low on the ESP-01)
119     } else {
120         digitalWrite(BUILTIN_LED, HIGH); // Turn the LED off by making the voltage HIGH
121     }
122 }
123
124
125
126
```

Code Example: Page 5

```
127 void reconnect() {
128     // Loop until we're reconnected
129     while (!client.connected()) {
130         Serial.print("Attempting MQTT connection...");
131         // Attempt to connect
132         // see https://docs.internetofthings.ibmcloud.com/devices/mqtt.html
133         if (client.connect(myclient, "use-token-auth", myclientpwd)) {
134             Serial.println("connected");
135             // Once connected, resubscribe
136             client.subscribe("iot-2/cmd/status/fmt/json");
137         } else {
138             Serial.print("failed, rc=");
139             // Note: list of errors here: https://github.com/knolleary/pubsubclient/blob/master/src/PubSubClient.h
140             Serial.print(client.state());
141             Serial.println(" try again in 5 seconds");
142             // Wait 5 seconds before retrying
143             delay(5000);
144         }
145     }
146 }
```

Code Example: Page 6

```
147 void loop() {
148     if (!client.connected()) {
149         reconnect();
150     }
151     client.loop();
152
153     long now = millis();
154     if (now - lastMsg > 2000) {
155         lastMsg = now;
156         ++value;
157         snprintf (msg, 75, "{\"d\":{\"myName\":\"ESPnodeMCU\",\"temp\":%ld}}", value);
158         Serial.print("Publish message: ");
159         Serial.println(msg);
160         client.publish("iot-2/evt/status/fmt/json", msg);
161     }
162 }
```



IND-2119 A Primer: Programming an IoT Device on IBM Bluemix

Reference Slides



Backup and Reference

- My blog post about the “Cabin Fever” project: <http://henrywill4.blogspot.com/2015/09/cabin-fever-part-5-mqtt-and-ibm.html>
 - And the most recent: <http://henrywill4.blogspot.com/2016/02/cabin-fever-part-6-smart-temperature.html>
- My Blog post about ESP 8266 NodeMCU <http://henrywill4.blogspot.com/2016/02/the-esp8266-nodemcu-using-arduino-ide.html>
- Bluemix: <http://bluemix.net>
- Getting started with IoT
 - Great Video about how to get started with Bluemix and IoT using the temperature sensor <https://www.youtube.com/watch?v=sCcFR92DA8w>
 - This is the place to start (and has lots of links to help: Getting Started in IoT Foundation: <https://www.ng.bluemix.net/docs/#services/IoT/index.html#gettingstartedtemplate>
 - Docs (Info) about IoT: http://iotf.readthedocs.org/en/latest/reference/concepts.html?cm_mc_uid=62867461194814392103270&cm_mc_sid_50200000=1441455248
- Stackoverflow (a great place to ask questions and get help): <http://stackoverflow.com/questions/tagged/bluemix>
- IoT Recipes:
 - How they work (video): <https://www.youtube.com/watch?v=wE6Yo8RWJIY>
 - Where they are located: <https://developer.ibm.com/iotfoundation/>
 - Search for IoT Recipes: https://developer.ibm.com/recipes/?post_type=tutorials&s=IoT
 - Quickstart: <https://quickstart.internetofthings.ibmcloud.com/#/>

More Backup and Reference

- Creating Applications with Node-RED for Bluemix: <https://www.ng.bluemix.net/docs/services/IoT/index.html#iot180>
- Arduino IDE: <https://www.arduino.cc/en/Main/Software>
- MQTT:
 - MQTT: <http://mqtt.org/>
 - Paho Project (open-source client libraries for MQTT) <http://www.eclipse.org/paho/>
 - Arduino MQTT Library: <http://pubsubclient.knolleary.net/>
- Amazing Watson Rover Demo of what can be done with Watson on Bluemix and IoT: <https://player.vimeo.com/video/130135196>
- Node_RED: <http://nodered.org/> and <http://noderedguide.com/> and <https://groups.google.com/forum/#!forum/node-red>
- Email to SMS Text Messages: <http://www.emailtextmessages.com/>

Recipes and Code that I Found Helpful – page 1 of 2

- This article “One Wire Digital Temperature. DS18B20 + Arduino ” was helpful to learn how to get the temperature sensor to work: <http://bildr.org/2011/07/ds18b20-arduino/>
- This one shows how to connect to quickstart. There is a second part, use the link near top for “view more content in this series”: <http://www.ibm.com/developerworks/cloud/library/cl-bluemix-arduino-iot1/index.html>
Note: the second part has a link to download the sample sketch (code). I like the way that this code’s “loop” will try to reconnect if it’s not connected (pull the internet connection out and plug it back in, you’ll see that it will reconnect)
- Note that the second part of that recipe includes a description of what the JSON format looks like <http://www.ibm.com/developerworks/cloud/library/cl-bluemix-arduino-iot2/index.html>
- This code README explains how to form the client ID (under the heading "Registered Flow from device and to device") <https://github.com/ibm-messaging/iot-arduino/blob/master/README.md>
- The Arduino example code located at this link gives an example of how to subscribe to MQTT and the commands from the “application”
https://github.com/knolleary/pubsubclient/tree/master/examples/mqtt_publish_in_callback
- This recipe was helpful to see the difference between a quickstart (unregistered) and registered flow: <https://developer.ibm.com/recipes/tutorials/connect-an-arduino-uno-device-to-the-ibm-internet-of-things-foundation/>

Recipes and Code that I Found Helpful – page 2 of 2

- This IBM Redpaper “MQTT and Arduino Devices” and the code there was also found to be helpful (but, note that it only starts the internet connection in the startup and therefore it does not automatically reconnect if disconnected as in the recipe above): <http://www.redbooks.ibm.com/redpapers/pdfs/redp4929.pdf>
- The following questions posted by me were also very helpful
 - <http://stackoverflow.com/questions/32961280/how-can-ibm-bluemix-node-red-iotout-node-be-set-up-correctly-to-send-command-to/32961307#32961307>
 - <http://stackoverflow.com/questions/33029450/how-can-i-convert-between-plain-text-in-json-in-node-red-function-node>
- When writing the JS code in the Node-Red “function” node to convert email to JSON, I found the following websites helpful:
 - http://www.w3schools.com/jsref/jsref_obj_string.asp
 - <http://stackoverflow.com/questions/4090518/string-to-int-use-parseint-or-number>
 - <http://nodejs.org>
 - <http://code.tutsplus.com/tutorials/nodejs-for-beginners--net-26314>
 - developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/
- This recipe was helpful to learn how to bind the IoT service to the application, register the device with IoT, and get an API key. It also explains the format of the topics for publishing and subscribing:
<http://www.ibm.com/developerworks/cloud/library/cl-mqtt-bluemix-iot-node-red-app/index.html>
- My blog post on Cabin Fever: <http://henrywill4.blogspot.com/2015/09/cabin-fever-part-5-mqtt-and-ibm.html>



Thank You



Ask Me Anything:
Connecting Node-RED Flows to the IBM
Watson IoT Platform for an IoT Device
IDA-6963