

developer.ibm.com

Turn your smartphone into an IoT device

Tutorial

11-14 minutes

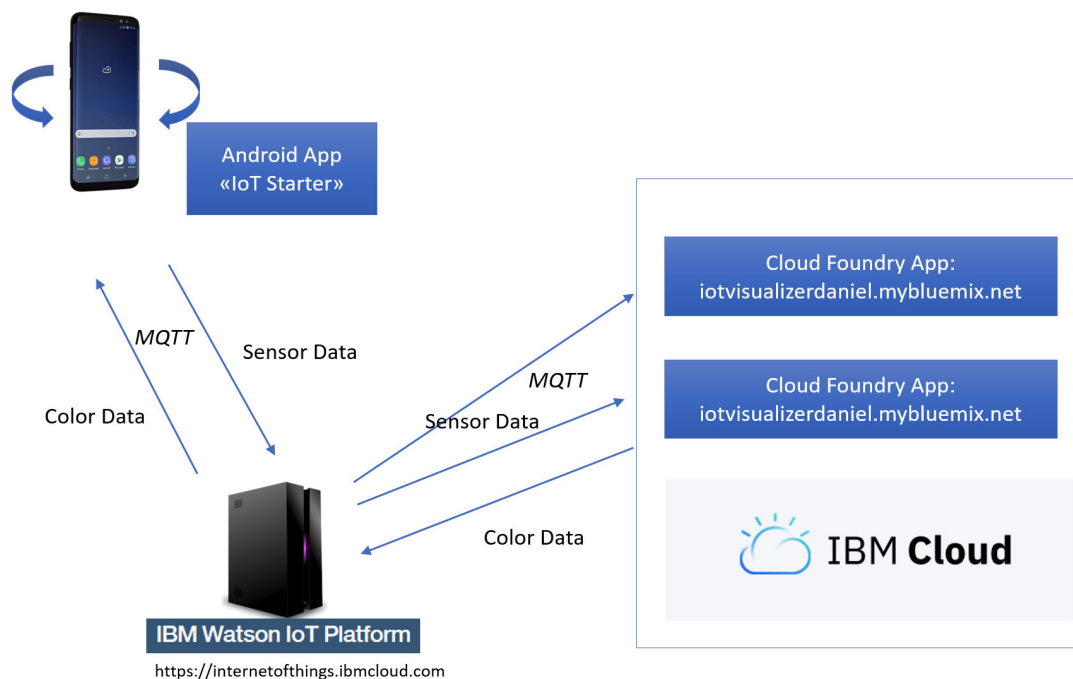
By Daniel Beguelin

Updated June 24, 2020 | Published November 10, 2015

This code pattern is part of the [2020 Call for Code Global Challenge](#).

It shows how you can send sensor data that is generated by your smartphone to the IBM Watson IoT Platform cloud-hosted service, and then create Cloud Foundry apps on the IBM Cloud that process, visualize, and store the data. Lastly, it shows you how to create an Android application for a smartphone.

Here is an overview of the architecture:



What you'll need to build your apps

- An IBM Cloud account; if you do not have an existing IBM Cloud account, start your [free trial](#). Learn more about IBM Cloud by reviewing the [Getting Started documentation](#).
- Download and install the [IBM Cloud CLI and Developer Tools](#).
- A smartphone (an Android or iOS device)

Step 1 Create an IoT app in the IBM Cloud

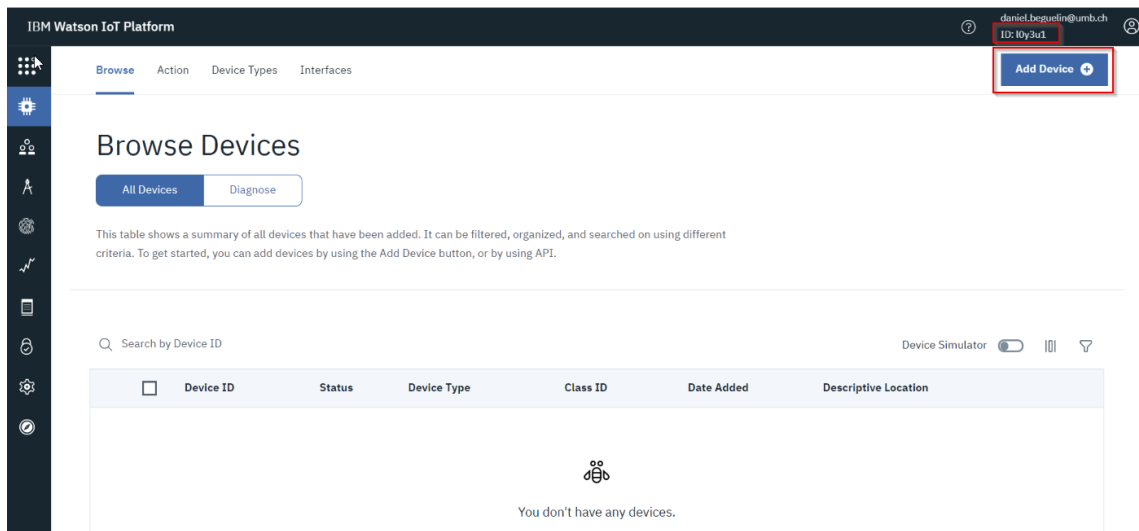
Work through the steps in this tutorial to [create a Node-RED and Watson IoT Platform starter app in IBM Cloud](#).

As a result, you will have a Node-RED Starter Kit application connected to the IBM IoT platform.

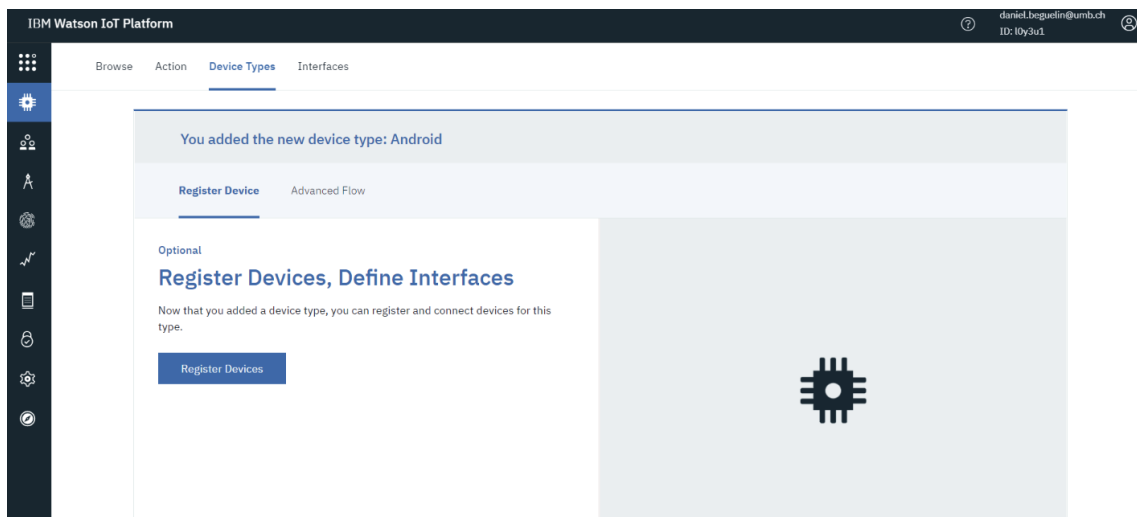
Step 2 Add a device that will send MQTT messages to the IBM Watson IoT Platform

After the last step in the tutorial in Step 1, the IBM Watson IoT Platform console is opened.

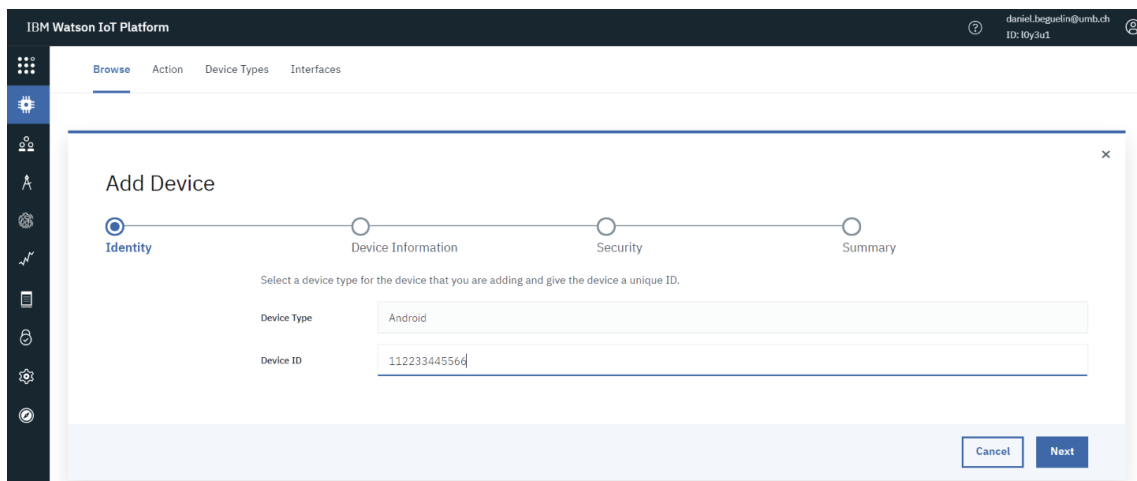
1. An organization ID is assigned to your app, and you will need this ID later when developing the mobile app. In the following image, the organization ID is `10y3u1`, which is displayed under your login information in the upper right corner of the dashboard.



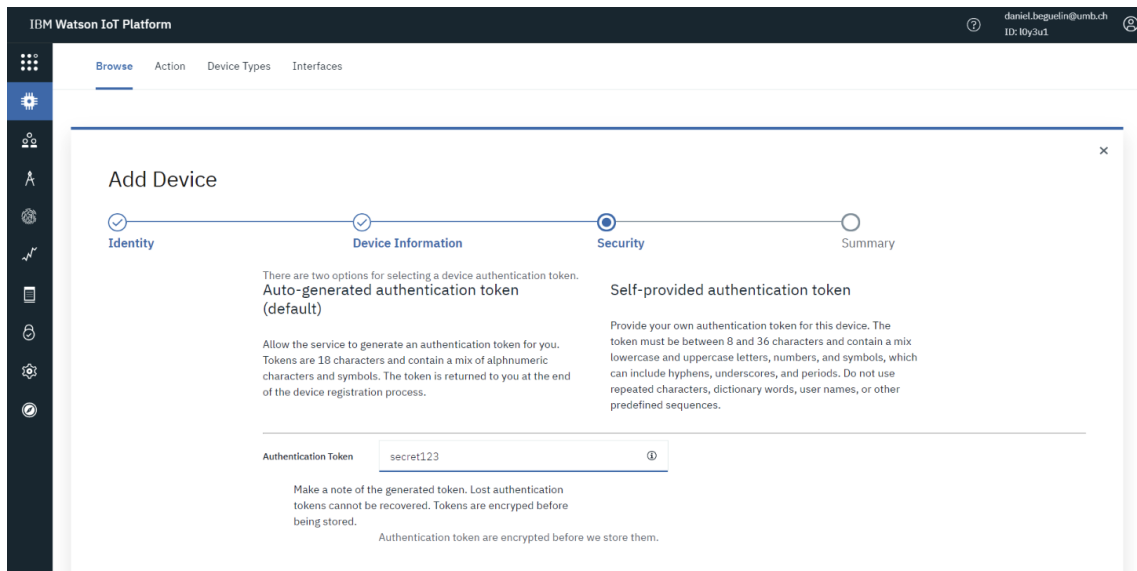
2. Click tab **Device Types**, then click **Add Device Type**. In your organization, you can have multiple device types each with multiple devices. A *device type* is a group of devices that share characteristics; for example, they might provide the same sensor data. In our case, the device type name *must* be “Android” (this device type name is required by the app that you will use later).
3. Click **Next**. A page is displayed where you can enter metadata about the device type, such as a serial number or model. You don't need to specify this information for this tutorial. Just click **Finish**.



4. Click **Register Devices**. Enter the device ID. The *device ID* can be, for example, the MAC address of your smartphone. However, it must be unique within your organization only. Therefore, you might enter, as I did here, something like “112233445566”.



5. Click **Next**. A page is displayed where you could enter metadata about the device. Leave it blank, and click **Next**.
6. On the security page, enter a value for the authentication token. Remember this value for later. Then, click **Next**.



7. Click **Finish**.

Now you are ready to send MQTT messages from a device to the IBM Watson IoT Platform.

Step 3 Install and configure the Android app

If you have an iPhone or iPad, you can use the [IoT Starter for iOS](#). Installation on the iPhone is not covered in this article, but if you do it on your own, you can use the iOS app for the rest of this tutorial.

You will use the IoT Starter for Android app to read and send sensor data on your smartphone. The source code and documentation of the app are in the [iot-starter-for-android GitHub project](#).

If you are experienced in Android development, you can [download the code](#) from GitHub, import it into your Android development environment, and then build the apk file.

Otherwise, to get the app installed and running quickly, follow these steps.

[Get the code](#)

1. On your phone, go to **Settings > Apps & notifications**. Tap

Advanced. Scroll down up to end of the page and tap **Special app access**

2. Select **Install unknown apps**

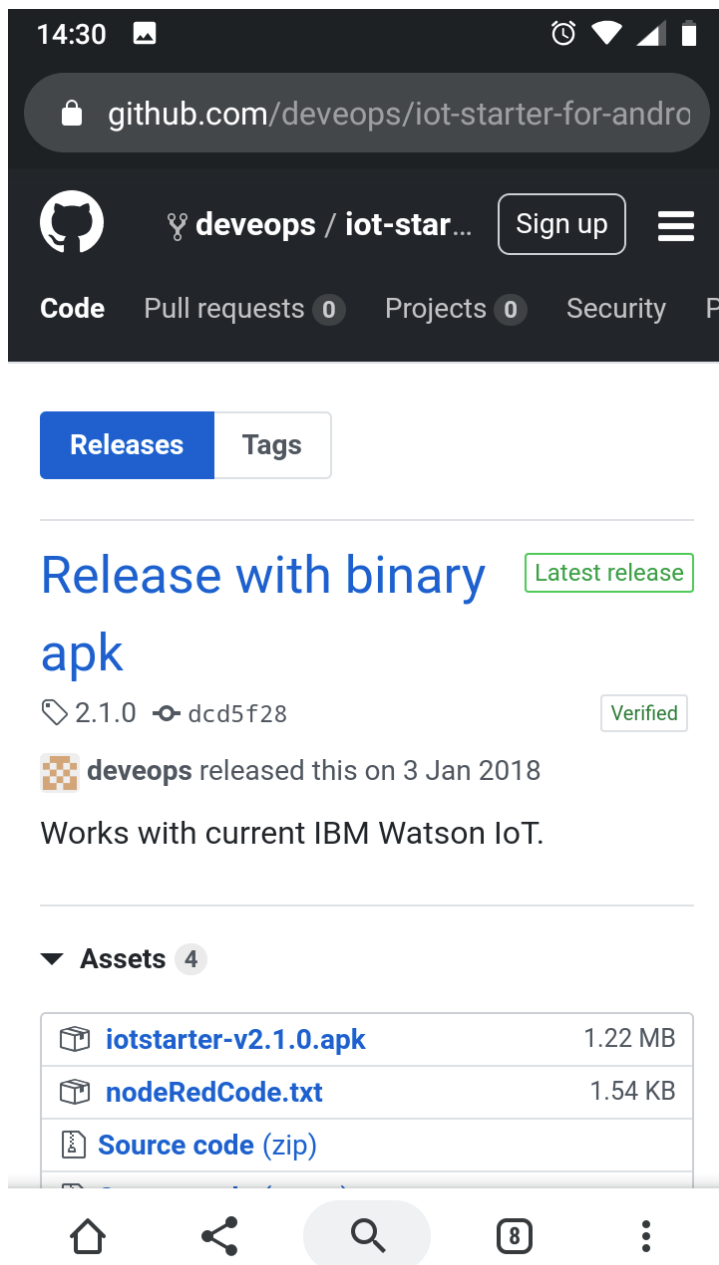
3. Click **Chrome** and enable **trust this source**

Note: on versions other than Android 9, the settings are slightly different.

Now you can install `.apk` files from outside of Google Play using Chrome.

1. Open the Chrome browser on your phone, and enter this URL:

```
https://github.com/deveops/iot-starter-for-android/releases
```



2. Open **Assets** and click the `iotstarter-v2.1.0.apk` link to download the `.apk` file. Confirm the warning saying that `.apk` files might be dangerous.

3. Once downloaded, click **Install** to install the app.

The IoT Starter app is now installed on your Android device.

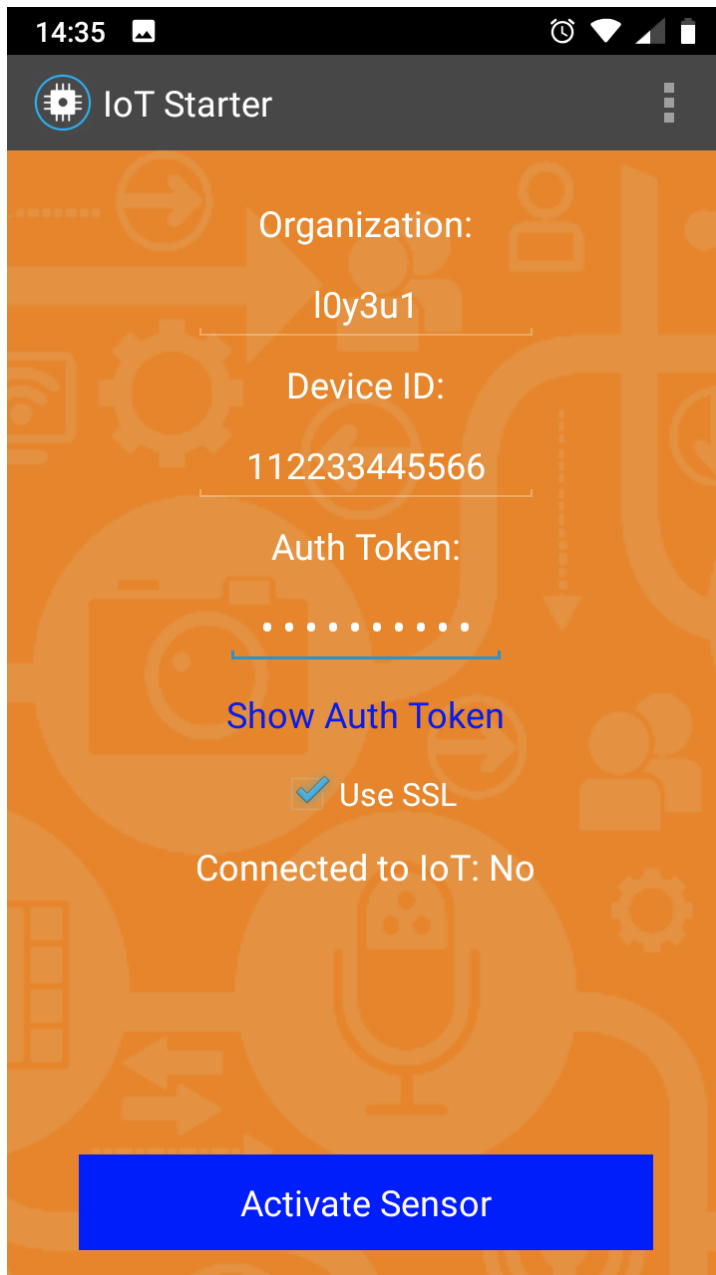
Next, you need to configure your Android app.

1. Start the IoT Starter app.

2. Click **Skip tutorial**.

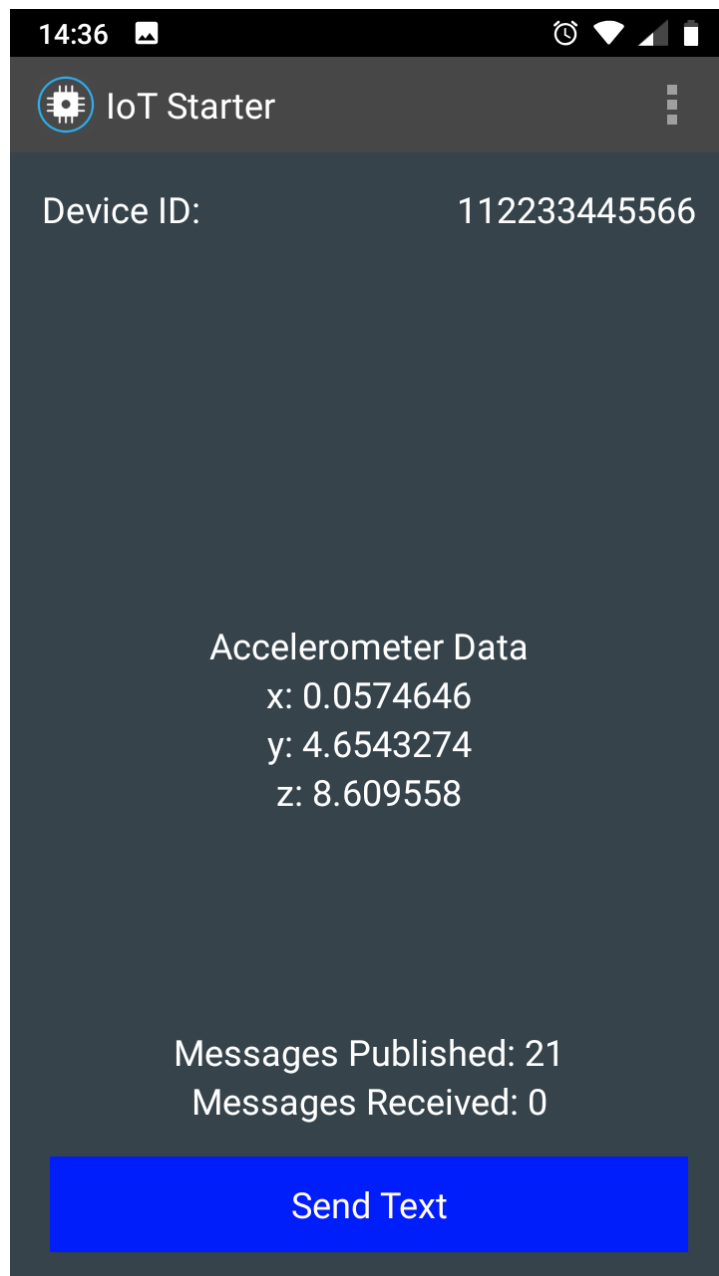
3. Enter the following parameters:

- **Organization:** The organization ID that was displayed on the IBM IoT server (at the start of " "). For example, `10y3u1` in this tutorial.
- **Device ID:** The device ID that you configured [above](#). For example, "112233445566" in this tutorial.
- **Auth Token:** The authorization token that you specified [earlier](#).
- Make sure that **Use SSL** is checked.



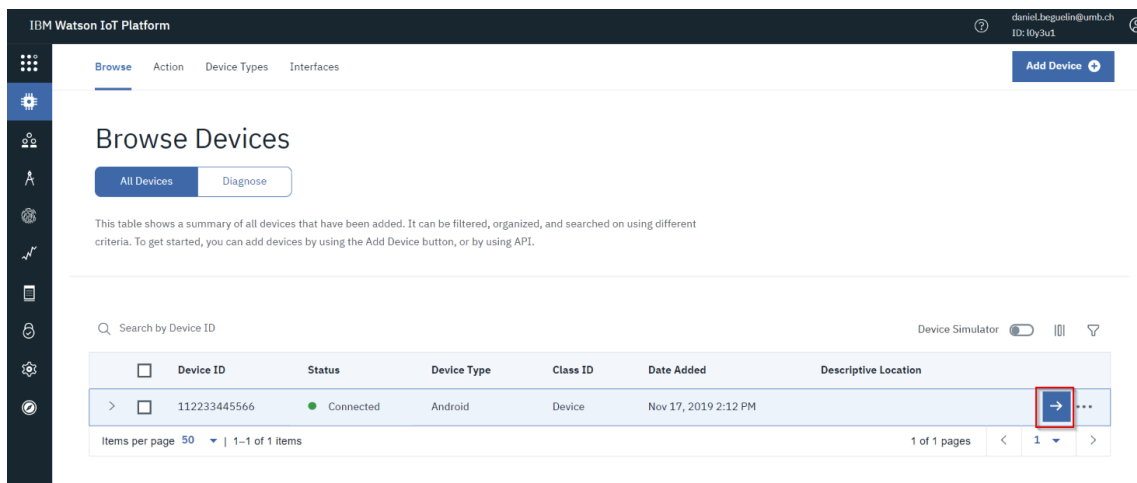
4. Click **Activate Sensor**. Now the app collects data from the acceleration sensor in your smartphone and sends the data to the IBM IoT server. The app displays the accelerometer data

and the number of messages that were published or received.

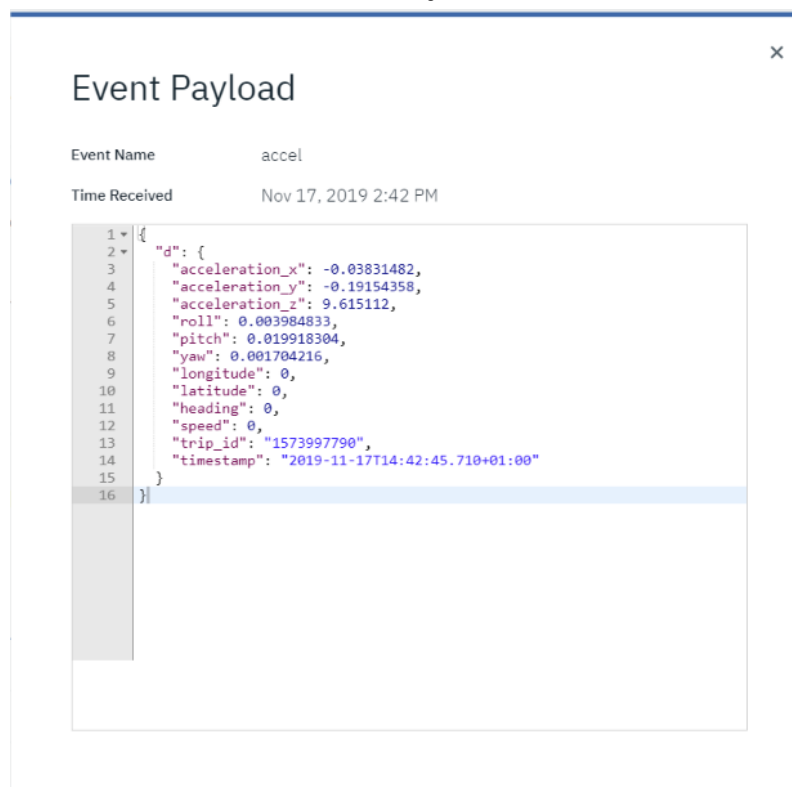


Step 4 Verify that messages are being sent from your smartphone to the Watson IoT Platform

1. Back on your computer, open the IBM Watson IoT Platform page for your organization again
2. In the left menu, click **Devices**. Your Android device is displayed.



- Click the arrow icon on the right of your Device ID. This opens an new page where you can see “Recent Events”. You should see events coming from your smartphone.
- Click one of the events. The messages that are sent from your smartphone are in JSON format. They contain acceleration and



position data.

Now you are ready to work with the message data on IBM Cloud.

Step 5 Process messages in a Node-RED flow

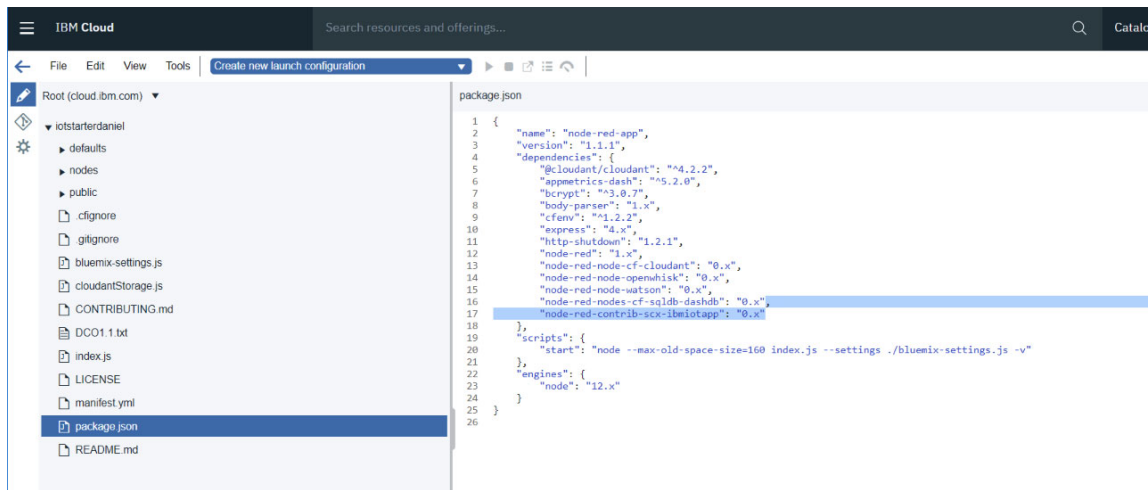
In this section, you will enhance your IBM Cloud IoT app by

using a Node-RED flow to process messages from your smartphone, and then send messages back to your smartphone. The phone will react on these messages by changing the background color in the app.

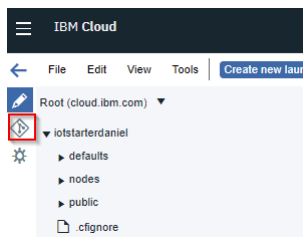
First you need to import the nodes `ibm iot in` and `ibm iot out` into your Node-RED application. These nodes allow you to easily connect to Watson IoT, but they are not in the palette by default. To do so, you need to create a continuous delivery toolchain and then add dependencies to the application.

NOTE: *There is also the possibility to directly import new nodes to the palette in the Node-RED editor. This would not require to set up a toolchain. But the free trial version of IBM Cloud does not provide enough memory for that option to work.*

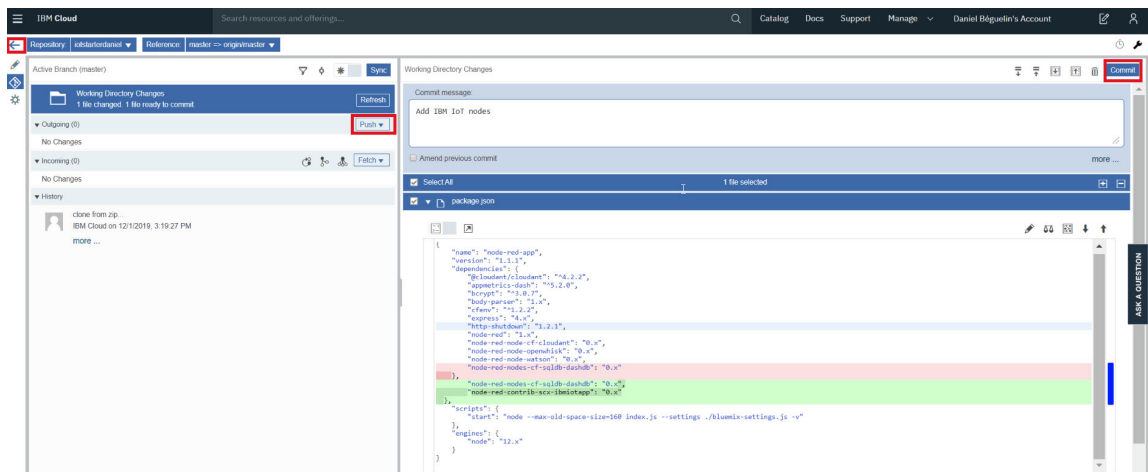
1. Open the Overview page of the Node-RED starter application, which you created in Step 1.
2. In the **continuous delivery** section of the dashboard, click **View Toolchain**.
3. Back on the Toolchain page, click the **Eclipse Orion Web IDE** icon. The IDE opens. Here you can change the code of your application.
4. Select the file **package.json** on the left side. In the editor on the right, add a line in the **dependencies** section: “node-red-contrib-scx-ibmiotapp”: “0.x”



5. Select menu File > Save. Then click the git icon in the left menu.

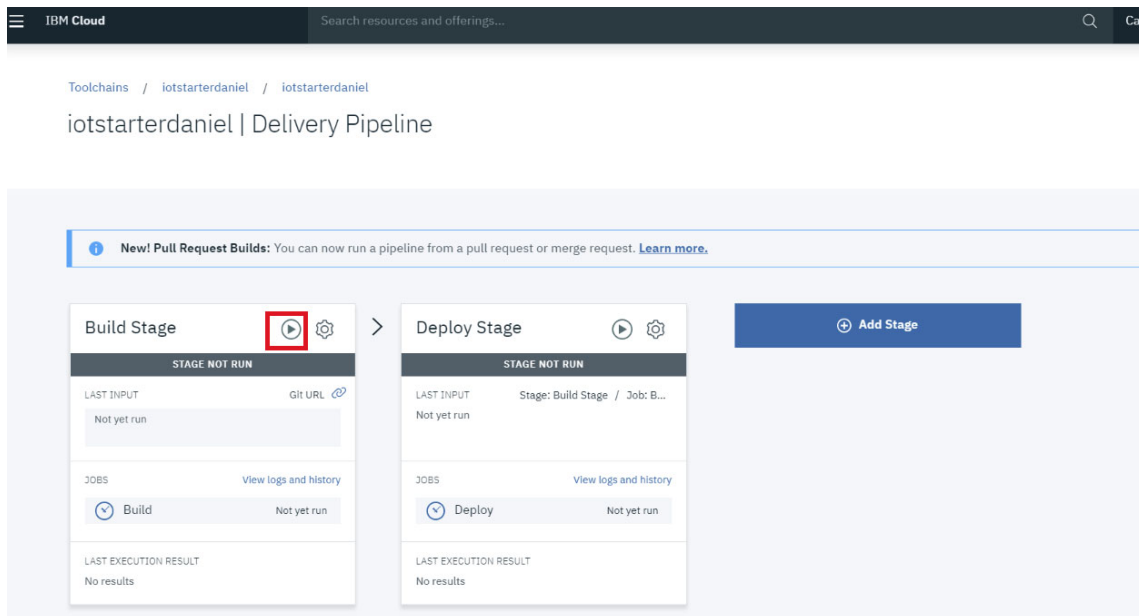


6. On the Git page, first click **Commit**, then **Push**. This pushes your change from the Git repository to the Application on the IBM Cloud.



7. Click the back arrow to get back to the Toolchain page. Now click **Delivery pipeline**.

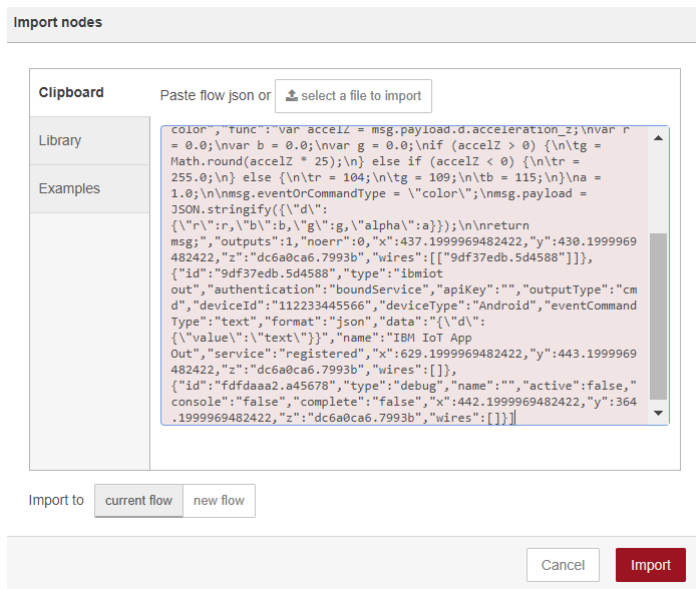
8. On the Delivery Pipeline page, stop your app, and then click the play icon next to Build Stage.



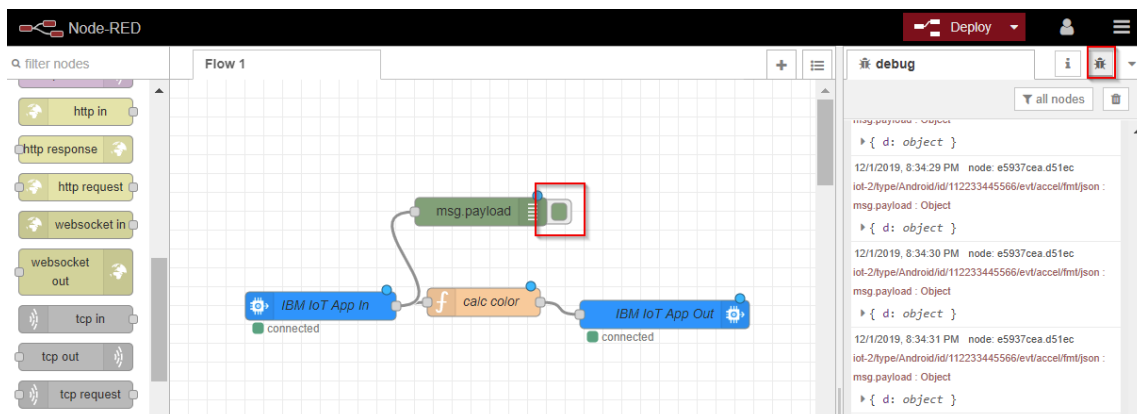
The build pipeline will take 5 – 10 minutes to complete.

Now you can create the actual Node-RED flow which processes messages.

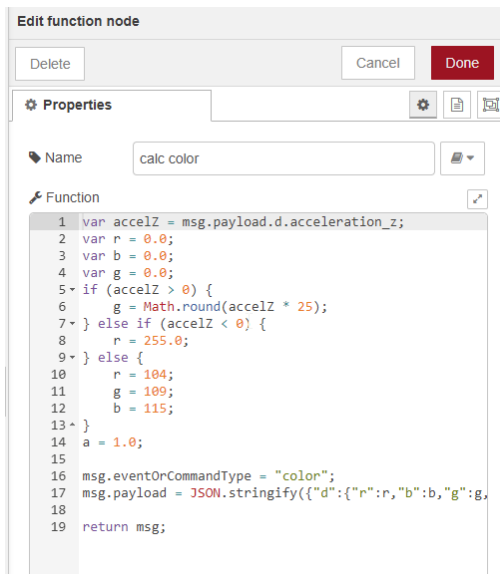
1. Open the Node-RED visual programming editor of your Node-Red starter app. (Review the tutorial you completed in Step 1 for details.)
1. Using the drag-and-drop features of this editor, you can plug together a flow of messages. You can create your own flow here, but now we will import the code. Download the following code from [GitHub](#) and save it as a text file.
2. Open the file in a text editor. Make sure that all the code is on a single line. Remove any line breaks. Copy the line of code.
3. In the Node-RED editor, press **Ctrl-I** to open the Import Nodes dialog. Paste the code, and click **OK**.



- Now you need to adapt the flow to your specific parameters. The only relevant parameter is the Device ID. Double-click the node **IBM IoT App out**. In the pop-up window, enter the Device ID that you used earlier (for example, 112233445566), and click **Import**.
- Click **Deploy** in the flow editor. The flow is deployed and should be active immediately.
- Move your smartphone around; flip and tilt it. The background color of the app on your phone should now change colors, depending on the orientation of the z-axis.
- In the Node-RED editor, click the rectangle next to the **msg.payload** node, and click the **debug** tab to enable debugging. You should see messages that are sent from your phone. The data is in JSON format.



- Inspect the flow. Double-click the **calc color** node. It calculates the red, green, and blue values based on the incoming z-acceleration value, and passes them on as JSON data.



You now have two-way communication between your smartphone and the first IBM Cloud IoT app.

Step 6 Create an IBM Cloud app to visualize sensor data

In this step, you create an additional app in the IBM Cloud, which receives the messages from your smartphone (now an IoT device) and visualizes the data.

[Get the code](#)

- Download the [rickshaw4iot-0.2.2.zip](#) file from the [rickshaw4iot](#) Github project for IoT Visualization. Extract the files to a local directory. Alternatively, you can clone or download the [Github repository](#).
- From the folder where you extracted the rickshaw4iot-0.2.2.zip file, open the manifest.yml file in a text editor.
 - Change the **host** and **name** parameters to a unique name. Again, it must be unique within the IBM Cloud, because it is used as a host name. For example, use

```
iotvisualizer<your name>.
```

2. Change the **domain** to the IBM Cloud location where you intend to run the app. In a free IBM Cloud account, all apps should run at the same location. Open your [IBM Cloud Resource List](#) to find your location. For example, if the Location is London, use `eu-gb.mybluemix.net` for the domain.

```
1 applications:
2   - disk_quota: 1024M
3     host: iotvisualizerdaniel
4     name: iotvisualizerdaniel
5     command: node app.js
6     path: .
7     domain: eu-gb.mybluemix.net
8     instances: 1
9     memory: 128M
10
```

3. Save the file.

3. In the preparation, you have installed the [IBM Cloud CLI and Developer Tools](#). Open a command or terminal window, and change to the directory where you extracted the `rickshaw4iot-0.2.2.zip` file.

4. Enter the following IBM Cloud CLI command: `ibmcloud login`
5. Deploy the application to IBM Cloud by entering the following command. Use the name you specified in the `manifest.yml`:

```
ibmcloud cf push your_application_name
```

This command looks for the file `manifest.yml` and uses the parameters from that file. It then uploads the code to IBM Cloud. After a while, you should see messages similar to the following ones:

```
Command Prompt

App iotvisualizerdaniel was started using this command 'node app.js'

Showing health and status for app iotvisualizerdaniel in org daniel.beguelin@umb.ch / space dev as daniel.beguelin@umb.ch...
OK

requested state: started
instances: 1/1
usage: 128M x 1 instances
urls: iotvisualizerdaniel.eu-gb.mybluemix.net
last uploaded: Tue Nov 19 14:33:22 UTC 2019
stack: cflinuxfs3
buildpack: sdk-for-nodejs

#0 state since cpu memory disk details
running 2019-11-19 03:34:36 PM 0.0% 23.3M of 128M 84.8M of 1G
```

6. In IBM Cloud, click **Dashboard**. You should see the app you just

deployed. Click it to see its details.

Resource list /

iotvisualizerdaniel ● Stopped [Visit App URL](#)

Org: daniel.beguelin@umb.ch Location: London Space: dev [Add Tags](#)

Runtime

- BUILDPACK**: SDK for Node.js™
- INSTANCES**: Stopped: 1 | Running: 0 | Health is 0%
- MB MEMORY PER INSTANCE**: 128
- TOTAL MB ALLOCATION**: 128 MB still available

Connections

No services are connected to this app
You can bind a service:

[Create connection](#)

Runtime cost

\$0.00
Current charges for billing period

\$0.00
Estimated total for billing period (Nov 1, 2019 - Nov 30, 2019)

Current and estimated cost excludes [connected](#) services.

- Click **Create connection**. Select the Internet of Things service that you defined in Step 1, and click **Connect**. Leave “Auto Generate” in the popup window and click **Connect** again.

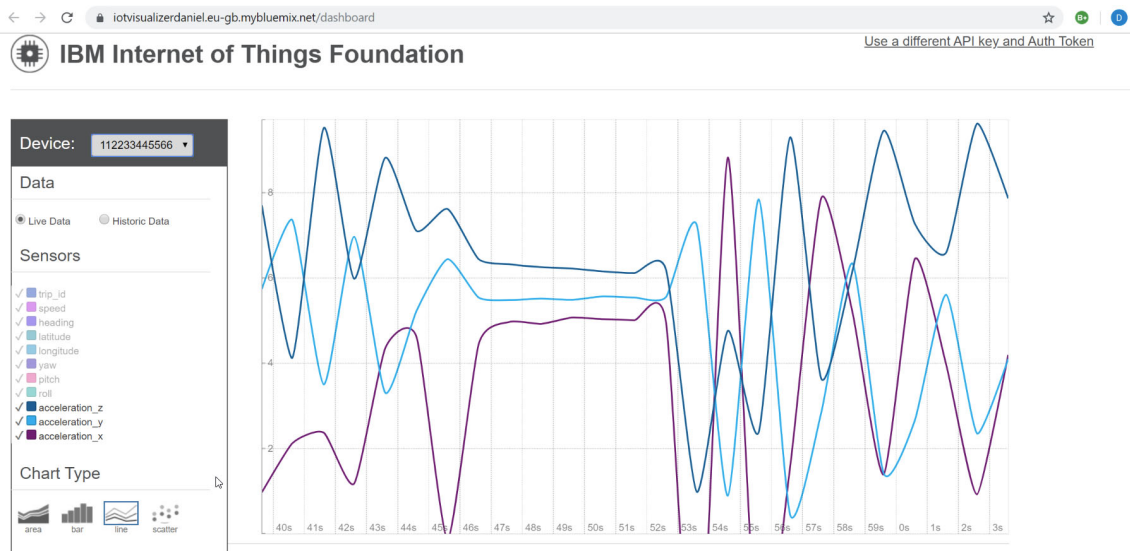
10 ▾ Items per page | 1-3 of 3 Items

1 of 1 pages < 1 >

SERVICES	RESOURCE GROUP	PLAN	SERVICE OFFERING
availability-monitoring-auto	--	Lite	Availability Monitoring
Cloudant-en	Default	Lite	Cloudant
Internet of Things Service 2	Default	Lite	Internet of Things Platform

Connect

- Click **Restage** to restage the app. Now the app can receive messages from the Watson IoT Platform.
- To verify the visualizer app, click the “Visit App URL” link in the app overview page. This opens `http://_your-app-name_. _your-location_.mybluemix.net`
- In the **Device** drop-down box, select your device id.
- Move your smartphone around. You should see the acceleration changes in the chart.



Conclusion

In this tutorial you learned how to easily turn your smartphone into a sensor device, connect it to the IBM Watson IoT Platform, and send and receive data. You also learned how to process and visualize device data on the IBM Cloud. With these two apps, you can recognize the value of IBM Cloud for the Internet of Things, and all you need is your own smartphone.