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 <u>2020 Call for Code Global</u> <u>Challenge</u>.

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 <u>free trial</u>. Learn more about IBM Cloud by reviewing the <u>Getting Started documentation</u>.
- 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 <u>create a Node-RED and</u> <u>Watson IoT Platform starter app in IBM Cloud</u>.

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.

IBM Wat	tson IoT Platform					0	daniel.t ID: I0y3	beguelin@umb.ch Bu1	8
	Browse Action Device Types In	terfaces					Add	d Device 🛨	
<u>°°</u>	Browse Devices								
Å	All Devices Diagnose								
6	This table shows a summary of all devices	hat have been a	dded. It can be filtered	l, organized, and searched on u	sing different				
~~~	criteria. To get started, you can add device:	by using the Ad	d Device button, or by	using API.					
6	Q Search by Device ID					Device Simulat	or 🌔		
ŵ	Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location			
$\oslash$									
				des					
				You don't have any devic	es.				
				Tou don't have any devic	co.				

- 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".

IBM	Watson IoT Platform						daniel.beguelin@umb.cf ID: l0y3u1	°
:::	Browse Action Device	Types Interfaces						
۵								
<u>°</u> 0							>	×
Å	Add Device							
\$	•	(	)	0	O			
~~~	Identity	D	evice Information	Security	Summary			
		Select a device type	for the device that you are addin	g and give the device a unique ID.				
8		Device Type	Android					
ŵ		Device ID	112233445566					
Ø								
						Ca	ncel Next	

- 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**.

IB M W	atson IoT Platform					?	daniel.beguelin@umb.ch ID: l0y3u1	9
:::	Browse Action Device	lypes Interfaces						
۰								
<u>°°</u>							×	
Å	Add Device							
8	\oslash				O			
~~	Identity	Dev	ice Information	Security	Summary			
		Auto-generated (default)	authentication token.	Self-provided auth	entication token			
6		Allow the service to get	nerate an authentication token for you.	Provide your own authenti token must be between 8 a	cation token for this device. The and 36 characters and contain a mix			
ŝ		Tokens are 18 characte characters and symbol	ers and contain a mix of alphnumeric s. The token is returned to you at the end	lowercase and uppercase I can include hyphens, unde	letters, numbers, and symbols, which rscores, and periods. Do not use			
0		of the device registration	on process.	predefined sequences.	nary words, user names, or other			
		Authentication Token	secret123	٩				
		Make a note of th tokens cannot be being stored.	ne generated token. Lost authentication e recovered. Tokens are encryped before Authentication token are encrypted before	we store them.				

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 <u>IoT Starter for</u> <u>iOS</u>. 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 <u>iot-starter-for-</u> <u>android GitHub project</u>.

If you are experienced in Android development, you can <u>download the code</u> 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-forandroid/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 <u>above</u>. For example, "112233445566" in this tutorial.
- Auth Token: The authorization token that you specified earlier.
- Make sure that Use SSL is checked.

14:35 🗖		ⓒ ▼ ◢ ∎
IoT Start	er	:
	Organization:	91.
	l0y3u1	
a O	Device ID:	
	112233445566	
	Auth Token:	
	Show Auth Token	Ż
Со	nnected to IoT: No)
	Activate Sensor	
		N 7 7 7 1

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.

IBM	Watso	on IoT P	latform	I.					G	Ð	daniel. ID: l0y	beguelini 3u1	@umb.ch	(e
		Browse	e Act	ion Device Types	Interfaces						Ad	d Devic	e 🕂	
۵			-											
<u>°°</u>		Bro	ows	e Devices										
Å		AI	ll Device	s Diagnose)									
\$		This tat	ble show	s a summary of all device	s that have been added.	It can be filtered, organiz	ed, and searched on	using different						
~~~		criteria.	. To get :	started, you can add devid	es by using the Add Devi	ce button, or by using AP	Ι.							
6		Q Se	earch by	Device ID					Device Simula	ator		101	$\nabla$	
ŝ				Device ID	Status	Device Type	Class ID	Date Added	Descriptive Location					
Ø		>		112233445566	Connected	Android	Device	Nov 17, 2019 2:12 PM				→		
		Items	per pag	e 50 ▼   1-1 of 1 iter	ns				1 of 1 pages	<	1	•	>	

- 3. 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.
- 4. 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.

≡	IBM Cloud	Search resources and offerings		Q	Catalog	Docs	Support	Manage 🗸	Daniel Béguelin's Account		Ľ	ጸ
÷	Repository. iotstarterdaniel 🔻 Reference: master	=> origin/master 💌									Ć	۶
1	Active Branch (master)	⊽ ¢ <b>* Sync</b>	Working Directory Changes						₹ ₹	+ +	1 Con	nmit
*	Working Directory Changes	Refresh	Commit message:									<u>^</u>
	▼ Outgoing (0)	Push 🔻	Add IBM IoT nodes									
	No Changes										11	
	▼ Incoming (0)	🕉 🐎 💩 Fetch 🔻	Amend previous commit								more	
	No Changes		Select All T	1 file set	lected						E E	1
	▼ History		🗹 🔻 🗅 package json									
	Cools franze		<pre></pre>	uenix-s	ettings.js -v	~			1 5	5 20 4	*	ASK A QUESTION

- 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.

Unitidad	Search	h resources and offerings				Q	Ca
Toolchains / iotstarte	rdaniel / iotstarterdan	iel					
intertextextextext		Dimeline					
lotstarterdan	iel   Delivery F	rpeline					
1 New! Pull Required to the second	est Builds: You can now ru	un a pipeline from a pull request	or merge request. Learn mo	re.			
•							
						í.	
Build Stage	<b>(ه)</b>	> Deploy Stage	(1) (1)	(	-) Add Stage		
				S.			
STAGE	NOT RUN	STAGE	NOT RUN				
LAST INPUT	NOT RUN	LAST INPUT Sta	NOT RUN ge: Build Stage / Job: B			I	
LAST INPUT Not yet run	NOT RUN	STAGE LAST INPUT Stay Not yet run	NOT RUN ge: Build Stage / Job: B			I	
STAGE I LAST INPUT Not yet run	not run	STAGE LAST INPUT Stay Not yet run	NOT RUN ge: Build Stage / Job: B				
STAGE I LAST INPUT Not yet run JOBS	Git URL @	LAST INPUT Stay Not yet run DOBS	NOT RUN ge: Build Stage / Job: B View logs and history			ſ	
STAGE I LAST INPUT Not yet run JOBS	Git URL @	LAST INPUT Stay Not yet run	NOT RUN ge: Build Stage / Job: B View logs and history Not yet run			ſ	
STAGE I LAST INPUT Not yet run JOBS OBS	Git URL @ Git URL @ View logs and history Not yet run	LAST INPUT Stay Not yet run JOBS Opploy	NOT RUN ge: Build Stage / Job: B View logs and history Not yet run				
STAGE I LAST INPUT Not yet run JOBS OBS LAST EXECUTION RESULT	Git URL @ Git URL @ View logs and history Not yet run	LAST INPUT Stay Not yet run JOBS Composition Result	NOT RUN ge: Build Stage / Job: B View logs and history Not yet run			ſ	

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.)
- 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 <u>GitHub</u> 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**.

Import nodes	
Clipboard	Paste flow json or 🔹 select a file to import
Library	<pre>color", "tunc": "var accel2 = msg.payload.d.acceleration_I;\nvar r = 0.0;\nvar b = 0.0;\nvar g = 0.0;\nif (accel2 &gt; 0) {\n\tg = Math.round(accel2 = 25):\n} els if (accel2 &lt; 0) {\n\tr =</pre>
Examples	255.0;\n} else {\n\tr = 104;\n\tg = 109;\n\tb = 115;\n}\na = 1.0;\n\msg.eventDrCommandType = \"color\";\nmsg.payload = JSON.stringfry((\"d\":
	<pre>{\r\":r\":r\":r\":r\":p\":g\":g,"alpha\":a}});\n\nreturn msg;","outputs":1,"noerr":0,"":437.199969482422,";":436.1999969 482422,":":"dc6a0ca6.7993b","wires":[["9df37edb.5d4588"]]}, {"id1:"9df37edb.5d4588","type":"ibmiot out","authentication:":boundService","apiKey":"","outputType":"cm d","deviceId":"112233445566","deviceType":"Android","eventCommand Type":"text","format":"josnof","data":"{\"d\": {\'value':":"text","format":"Signof","data":"{\"d\": {\'value':":"text","format":"Signof","data":"{\"d\"d\": {\'value':":"text","format":"signof","data":"["d\"d\": {\'value':":"text","format":"signof","data":"["d\"d\"d\": {\'value':":"text","format":"signof","data":"["d\"d\"d\"d\"d\"d\"d\"d\"d\": {\'value':":"text","format":"signof","data":"["d\"d\"d\"d\"d\"d\"d\"d\"d\"d\"d\"d\"d\"</pre>
Import to current	flow new flow
	Cancel Import

- 4. 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.
- 5. 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.
- 7. 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

- 1. Download the <u>rickshaw4iot-0.2.2.zip</u> file from the <u>rickshaw4iot</u> Github project for IoT Visualization. Extract the files to a local directory. Alternatively, you can clone or download the <u>Github</u> <u>repository</u>.
- 2. 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 <u>IBM Cloud Resource List</u> to find your location. For example, if the Location is London, use eugb.mybluemix.net for the domain.

1	applications:
2	- disk_quota: 1024M
3	host: iotvisualizerdaniel
4	<pre>name: iotvisualizerdaniel</pre>
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 <u>IBM Cloud CLI and</u> <u>Developer Tools</u>. 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:



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

#### deployed. Click it to see its details.

	Q Catalog Docs Supp	ort Manage 🗸 Daniel Bé	guelin's Account 🛛 🖉	Ŕ
Getting started Overview Runtime	Resource list / js iotvisualizerdaniel • stopped y Org: daniel.beguelin@umb.ch Location: London Space: dev	isit App URL Add Tags	Routes 💌	:
Connections Logs API Management Autoscaling Monitoring	Runtime js buildpack SDK for Node.js ^{**} J SDK for Node.js ^{**}	- 128 + 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)	
<		services.	si excludes <u>connected</u>	

7. 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.

.0    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

- 8. Click **Restage** to restage the app. Now the app can receive messages from the Watson IoT Platform.
- 9. 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
- 10. In the **Device** drop-down box, select your device id.
- 11. 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.