Setting Up an IoT Gateway to Connect to the DreamFactory* Open Source Backend Platform

Introduction

This tutorial demonstrates how to set up an Intel® IoT Gateway to write sensor data to any SQL or NoSQL database using DreamFactory* for the backend database access. DreamFactory is an open source backend that provides a ready-to-use REST API for easily storing sensor data in the cloud.

The gateway runs a Linux* image with Node-RED, which allows users to create flows for processing data. The tutorial adds a DreamFactory node to Node-RED and uses it in a flow to make HTTP calls to save the sensor data. As sensor data is created, it can be viewed in a simple Internet of Things (IoT) app that queries and charts the last hour's data. DreamFactory acts as the backend for all of this and provides the REST API for creating and retrieving the database records. This tutorial uses the default MySQL database, but it also describes how to use a NoSQL database such as MongoDB instead.

The main steps are as follows.

- Sign up for or create a DreamFactory Services Platform (DSP)
- Import the IoT app and schema into the DSP
- Install Node-RED on the Intel IoT Gateway
- Add a DreamFactory node to Node-RED
- Start Node-RED and deploy a flow to create sensor data
- Launch the IoT app
- See the sensor data populate the chart in the IoT app

Prerequisites

Start with an Intel IoT gateway, or similar device, that is running Linux and Node.js so that Node-RED can be installed. Serial port or SSH access to the device is required. Here are two links that may be helpful.

https://software.intel.com/en-us/iot/getting-started

https://software.intel.com/en-us/iot/downloads

Get a DSP

Those new to DreamFactory can create a free, hosted DSP at **www.dreamfactory.com** or use one of the Bitnami installers to install DSP. See the link below for details.

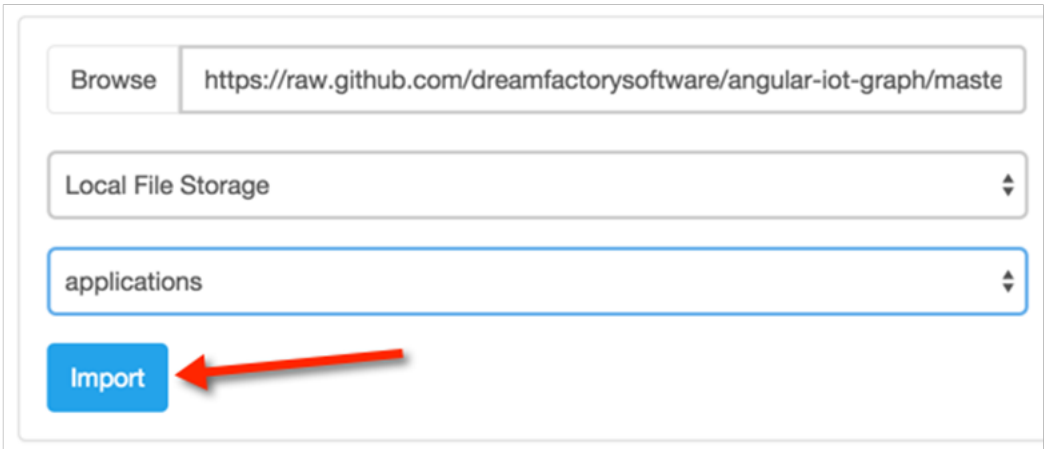
https://github.com/dreamfactorysoftware/dsp-core/wiki/Usage-Options

To verify Dreamfactory was installed correctly, make sure the DSP admin console can be accessed. For example, after creating a hosted DSP named 'dsp-foo,' the admin console would be accessible at https://dsp-foo.cloud.dreamfactory.com upon logging in with the admin email and password. For a hosted DSP, this is the email and password used to sign up for DreamFactory at **www.dreamfactory.com**.

Import IoT App and Schema into a DSP

In the DSP admin console, go to the Apps tab and select the Import option from the left sidebar. Paste the following URL in the input field to the right of the Browse button.

https://raw.github.com/dreamfactorysoftware/angular-iot-graph/master/iot.dfpkg



https://raw.github.com/dreamfactorysoftware/angular-iot-graph/maste

Local File Storage

applications

Figure 1. Import the IoT App and Schema

As shown Figure 1, select 'Local File Storage' and 'applications,' and then click the Import button.

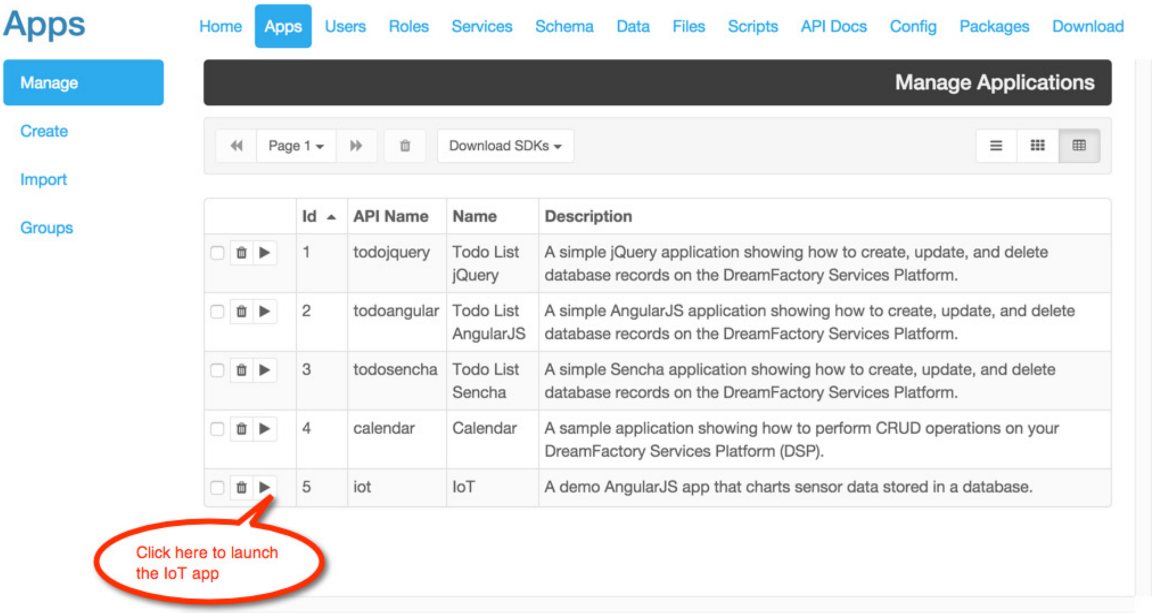
This will import both the IoT app and the associated schema for the 'iot' table. The schema is available from GitHub at:

https://raw.githubusercontent.com/dreamfactorysoftware/angular-iot-graph/master/schema.json

The table has four fields - id, created_date, device_id, and temp. When using a NoSQL database like MongoDB, the schema is not used.

Launch IoT App

Click on the Apps tab in the DSP admin console, as shown in Figure 2. Launch the IoT app by selecting it from the list. The app can be left running, but it won't display any data until the IoT Node-RED flow is deployed.



	Id ▲	API Name	Name	Description
□ 🗑 ▶	1	todojquery	Todo List jQuery	A simple jQuery application showing how to create, update, and delete database records on the DreamFactory Services Platform.
□ 🗑 ▶	2	todoangular	Todo List AngularJS	A simple AngularJS application showing how to create, update, and delete database records on the DreamFactory Services Platform.
□ 🗑 ▶	3	todosencha	Todo List Sencha	A simple Sencha application showing how to create, update, and delete database records on the DreamFactory Services Platform.
□ 🗑 ▶	4	calendar	Calendar	A sample application showing how to perform CRUD operations on your DreamFactory Services Platform (DSP).
□ 🗑 ▶	5	iot	IoT	A demo AngularJS app that charts sensor data stored in a database.

Click here to launch the IoT app

Figure 2. Launch the IoT App from the DSP Admin Console

Connect to the Intel® IoT Gateway

Now that DreamFactory and the IoT app are set up, the next step is to get Node-RED running on the gateway so it can generate some data. Log in as user 'root' and set the system date using the gateway's serial connection. This can prevent problems with SSH and other things. Issue the following command from the serial port.

```
date MMDDhhmmYYYY
```

For example Jan 24, 2015 17:43 would be:

```
date 012417432015
```

Switch over to SSH

The gateway should be on the network, making it possible to SSH to it. This is much better than using the serial port. To get the IP address, type ‘ip a’ from the serial port.

```
root@quark016cff:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
   inet6 ::1/128 scope host
     valid_lft forever preferred_lft forever
2: enp0s20f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast qlen 1000
   link/ether 98:4f:ee:01:6c:ff brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.39/24 brd 192.168.1.255 scope global enp0s20f6
   inet6 fe80::9a4f:eeff:fe01:6cff/64 scope link
     valid_lft forever preferred_lft forever
```

Create an SSH connection to that IP address as user ‘root’. For example, if the IP is 192.168.1.39, enter:

```
ssh root@192.168.1.39
```

The serial port is no longer needed since one or more SSH connections can be used instead.

Update npm and Node.js on the Intel® IoT Gateway

Node.js and npm should already be installed. Update npm since it will be used to install Node-RED. To see which versions are installed, type ‘npm -v’ and ‘node -v’. Then to update npm, type:

```
npm install npm -g
```

Type ‘npm -v’ again to see if a newer version was installed.

Install Node-RED

To install Node-RED, type:

```
cd ~
npm install node-red
```

Add DreamFactory* Node to Node-RED

It is necessary to clone a repo from GitHub to get four files that define the DreamFactory node. Copy these files over to the io directory by typing:

```
cd ~
mkdir repos
cd repos
git clone https://github.com/dreamfactorysoftware/node-red-contrib-dreamfactory.git
cd node-red-contrib-dreamfactory
cp *.js  *.html ~/node_modules/node-red/nodes/core/io
```

Start Node-RED

To start Node-RED, type:

```
cd ~/node_modules/node-red
node red.js

  Welcome to Node-RED
    ===================

  23 Feb 14:38:26 - [info] Version: 0.10.2
  23 Feb 14:38:26 - [info] Loading palette nodes
  23 Feb 14:38:49 - [warn] -----------------------------------------
  23 Feb 14:38:49 - [warn] Failed to register 4 node types
  23 Feb 14:38:49 - [warn] Run with -v for details
  23 Feb 14:38:49 - [warn] -----------------------------------------
  23 Feb 14:38:49 - [info] Server now running at http://127.0.0.1:1880/
  23 Feb 14:38:49 - [info] Flows file not found : flows_quark016cff.json
  23 Feb 14:38:49 - [info] Starting flows
```

Disregard the message ‘Failed to register 4 node types,' since those node types are not needed. Likewise, disregard the message ‘Flows file not found : flows_<device>.json,’ because this file will be created when the first flow is deployed.

Connect to Node-RED

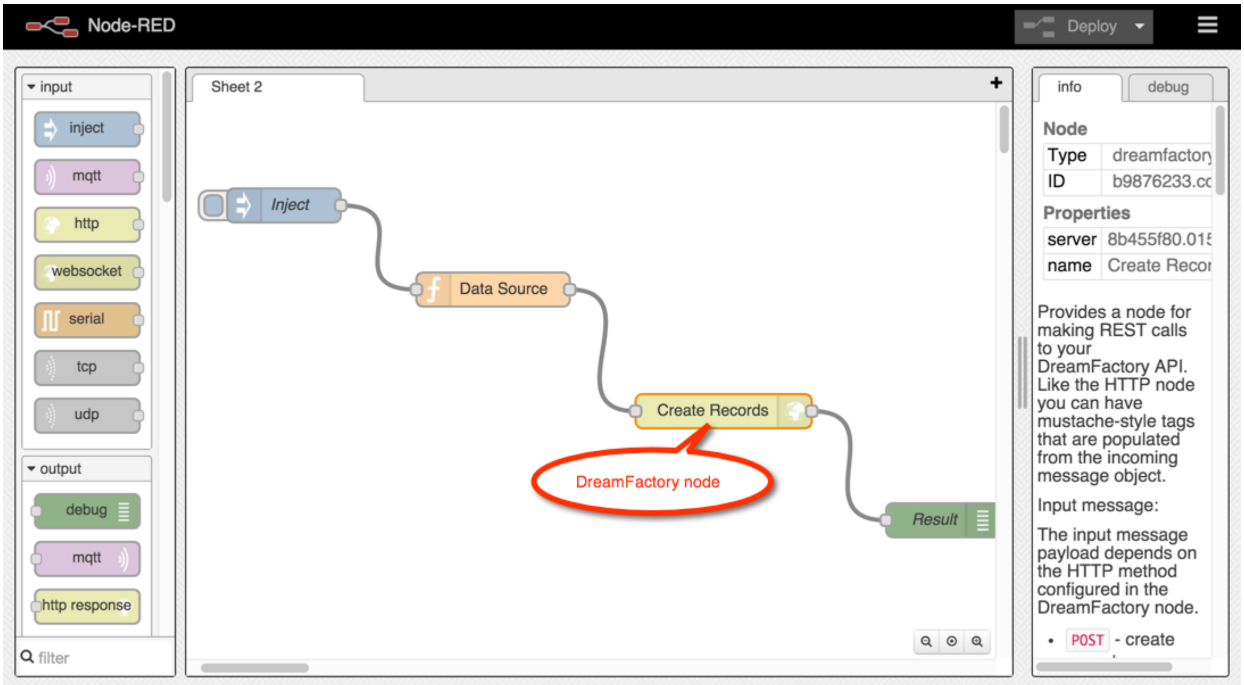
Node-RED is running, so point a browser to the IP for the gateway, port 1880 (e.g., http://192.168.1.39:1880). This should display the Node-RED user interface (UI).

Import IoT Flow into Node-RED UI

The JSON that defines the IoT flow in Node-RED is available from GitHub at:

https://raw.githubusercontent.com/dreamfactorysoftware/node-red-contrib-dreamfactory/master/iot-flow.json

Copy this data and select Import...>Clipboard from the menu in the Node-RED UI (Figure 3) and paste into the text area. Then click OK, and click again to place the flow on the workspace.



Node

Type	dreamfactory
ID	b9876233.cc

Properties

server	8b455f80.01
name	Create Recor

Provides a node for making REST calls to your DreamFactory API. Like the HTTP node you can have mustache-style tags that are populated from the incoming message object.

Input message:

The input message payload depends on the HTTP method configured in the DreamFactory node.

- POST - create

Figure 3. IoT application Flow Deployed in Node-RED

There are four nodes in this flow.

1. Inject - A node that triggers a new sensor reading every 60 seconds. Click the button to manually trigger a new sensor reading.
2. Data Source - A Function node containing the JavaScript for creating the simulated sensor data.
3. Create Records - A DreamFactory node that writes the payload received from the Function node to the database using the DreamFactory REST API. This example shows a DreamFactory node that creates records (HTTP verb = POST). The verbs GET, PUT, MERGE, and DELETE are also supported.
4. Result - A Debug node that displays the result of the REST API call to create the records.

Double click the third node, then click the pencil icon to edit the remote server configuration. Enter the DSP URL, DSP user email, and DSP user password fields.

- DSP URL: The DSP created earlier, (e.g., https://dsp-foo.cloud.dreamfactory.com).
- User Email: The email address for the DSP user, the one used to access the admin console.
- Password: The DSP user password.

Save these changes, and click the Deploy button. The flow creates a new record for each of the three sensors at startup and then once every minute.

View IoT App

Go back to the tab where the IoT app is running. If it is not visible, go back to the DSP admin console, and click the LaunchPad button in the upper right menu bar. Select the IoT app from the list to open it. The chart should begin to populate with sensor data as the records are created. The Inject node in Node-RED can be used to force the creation of more data points, as shown in Figure 4.

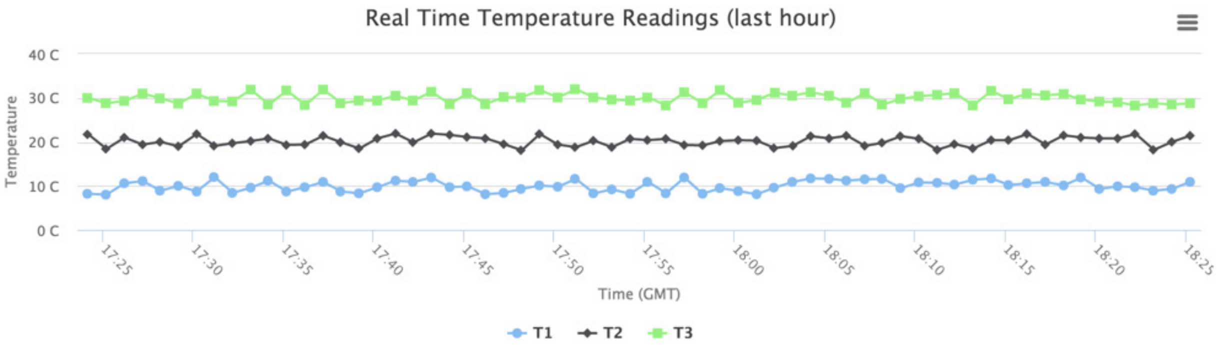


Figure 4. Live Data Generated from Node-RED

Tip: If the IoT app was launched from the DSP admin console, the session will eventually time out and the app will stop working. This can be prevented by adding the user DSP credentials to the app. Go to the Files tab in the admin console and edit the file app.js for the IoT app. If the session has timed out, it will use these credentials for basic HTTP authentication.

NoSQL Usage

Switching over to use a NoSQL database requires a few easy steps.

- From the Services tab of the DSP admin console, create a new service for a MongoDB account. Be sure to set the api name to 'mongo'.
- Create a new MongoDB collection named 'iot'. An easy way to do this is from the API Docs in the DSP admin console.
- In the Node-RED UI, edit the DreamFactory node named "Create Records". Click the pencil icon and change the path from /rest/db/iot to /rest/mongo/iot.
- In the Node-RED UI, edit the Function node named 'Data Source,' and set the useMongo variable to true. Save changes and deploy the flow.
- From the Files tab in the admin console, edit the file app.js for the IoT app, and set the useMongo variable to true. Save changes and reload the app.

Summary

To recap, this tutorial creates a DSP and imports the IoT app and schema. Next, the tutorial installs Node-RED, adds the DreamFactory node, and deploys the flow. At the end, the IoT app uses a date filter to retrieve the last hour's data from a SQL or NoSQL database.

Printed in USA 0415/MG/TM/PDF ♻ Please Recycle 332283-001US