AT&T M2X & Flow tutorial

This guide will show you how to get your device talking to the cloud based time-series data storage (M2X) and AT&T Flow application service.

Getting started

Create an AT&T IoT Services account

IoT Developer Services account is free and takes moments to setup up. It will give you access to IoT Starter Kit, M2X, AT&T Flow, and cellular connectivity API access, if you purchase an IoT Starter Kit hard- ware.

IoT Starter Kit

You have a choice between purchasing a SIM Card only, or a complete hardware kit with cellular shield and LTE module built-in. Both options come with a data plan and a connectivity API.

All you need to get up and running fast.
To purchase IoT Starter Kit, or sign-up for an AT&T IoT Services account, go to: starterkit.att.com

M2X

Navigate to https://m2x.att.com.
It comes to you free for up to 10 devices.
At its heart, M2X is a time-series data store – but it’s also more, including a device manager, authorization manager, and dashboard widget builder. We’ll come back to it soon.

AT&T Flow

Select “Flow” from the “AT&T IoT Services” dropdown in the top-left, or navigate to https://flow.att.com. Flow is GUI-based IoT development tool based on NodeJS with many ways to get data in and out of it, as well as implement business rules – all with version control, teams, and scalable application containers.

AT&T Flow gives you up to 100,000 data points per month free of charge.

If all this is new to you might check out https://flow.att.com/start. Login with the same credentials you used for your AT&T Developer account.

Setting up M2X

The AT&T M2X Data Service can receive, store, and react to data from a data source. In order to prepare the service for your data source, the first step is to set up a Device, which is a way to prototype your Device before launch.

Follow these steps to set up a Device:

1. Go to https://m2x.att.com/onboarding.

2. You will be asked to set up your first Device as either a physical device or a virtual device. If you choose physical, it will assist you in finding the correct client library. However, we will provide you with client library for this event. Therefore, click on Virtual Device.

3. The first time you log in to the M2X website, you can be led through the first-time process of creating a new Device and Stream. However, for this exercise, we will use the more standard procedure. In the top right corner, click on Skip Setup.

4. Click on Create Device.

5. For Device Name, type “test”. Leave everything else the same. Click Create.

6. Your Device will have a Device ID and API Key associated with it. You will use these later.

7. Let’s say that you were collecting information on ambient temperatures, and you wanted to keep track of the temperature for each asset as a function of time. Let’s add a data stream for device. Scroll down and click on the Add Stream button.

8. Type in “temp” for the Stream ID, “Temperature” for the Display Name, leave the Steam Type as Numeric, and then click on the **Aa** button.

9. Click on the **D**, then scroll down and select “degrees Celsius”.

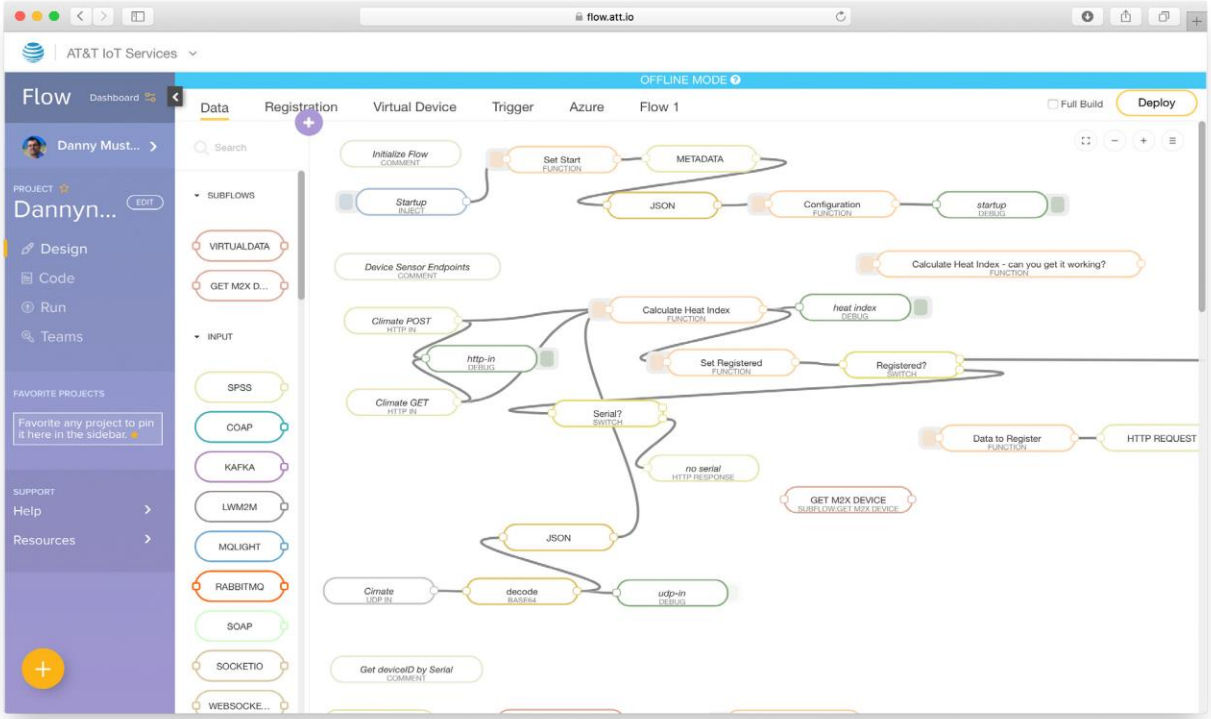
10. Click on Save to create the stream.

11. In the Overview tab you’ll see a link to Streams followed by Triggers. Click on Triggers and then on the “Add Trigger” button.

12. Give the trigger a name like “Hot Temp”, select Temperature from the Stream drop-down, “>” from the Condition drop-down, enter a threshold value of 40 (remember, this is in Celsius), and reset condition of 37. Leave the Notification Frequency set to Single and click on the Save button at the bottom.

Setting up AT&T Flow

You now have a Device, a stream, and a trigger. Next, you'll send some data to that Device. Follow these steps to create a flow in Flow Designer to talk to M2X. Projects contain the flows that together make your application.



Let’s start by creating your first project: This step is to make a simulated value of a specific stream.

1. In a new tab navigate to https://flow.att.com/ or select Flow from the AT&T IoT Services drop-down at the top of the page.

2. Click Login. As long as you are signed in with M2X, you will be signed into Flow Designer. (If not, use your M2X username and password.) If this is the first time you’ve logged into Flow Designer, a new Flow Designer account will be created for you.

3. Click the + button to create a new project.

4. Give your project a name, something like ‘My first project’. You can also add a description if you like, ‘Testing Flow Designer’ for example.

5. Keep Visibility set to private (you can always change it later). Internal or Public visibilities allow others to see your project, clone it, etc. and is great for collaborative work. For now, we’ll keep this first project Private.

6. Click “Create”. The system will import the required template files and will set the environment for you. Once done, your new project is in ‘Design’ mode, ready for you to write your first flow.

7. Your project was created with a default first flow called Flow 1. From the node palette on the left grab an Inject node from the Input section (you can find it easily by scrolling down or typing in “inject” into the Search box at the top of the palette) and drop it onto the canvas. By default this simply sends a Unix timestamp to the next node. Inputs for nodes connect on the left and outputs connect on the right so that the flow, itself, reads from left to right.

8. Grab a Function node from the Function section and drop it to the right of your Inject node. Click and drag on the output connector (white box on the right) to create a line connecting it to the input of the Function node. Now, the Inject node will send a msg object to the Function node with msg.payload set to the timestamp.

9. Double click on the Function node to open it. Name the function node something like “M2X Lead-in” and add the following code:

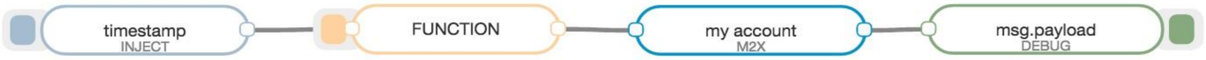
   ```
   msg.topic = “devices”;
   msg.action = “postMultiple”;
   msg.topic_id = “your_device_ID_not_name”;
   var timestamp = new Date(msg.payload).toISOString();
           // change timestamp to Date format
   msg.payload = {
        “temp”: [// this should be same as stream name
               { “timestamp”: timestamp, “value”: 34 }]
   };
   return msg;
   ```

10. Set the msg.topic_id to your *device ID* (not name) from M2X (found at the top of the device’s detail page).

11. Click on the OK button at the bottom to save your function.

12. Take a M2X node from the Storage section and drop it onto the canvas. Double-click on it to open it up. Next to the Feed click on the pencil icon to create a new M2X configuration. Name it “My M2X Account” and copy and paste your **Master API Key** from the M2X account page (https://m2x.att.com/account). Click Add to save it, then OK to save the M2X node. Link the Function node to the M2X node.

13. From the Output section, drop a Debug node to the right of the M2X node. Link the output of the M2X node to the Debug node.
    Your output should look like this:



14. Click on the Deploy button in the upper right hand corner. You'll see a progress report updated you on the status of the build.
    Once successfully completed, your flow has been built and deployed to the AT&T IoT Platform as a Service (PaaS), to a special default environment created for each project called 'sandbox'. Once finished, you may need to click on the 'Enter Online Mode'

    Note: 🔌 *Button in the upper right corner is used to change modes between 'Online Mode' and 'Offline mode.' Online mode means you are actually running your code on a designated 'Docker' (lightweight portable code execution container, equivalent to virtual machine, but much more efficient), as if you were running it on your own machine. Offline mode in contrast, means as if you are looking at an architectural drawing of your house (the flow), but unable to actually walk up the stairs to 'test' or debug their pitch.*

    Note: The Online button icon indicates the state that you will be in once you've clicked it, not the state that you are currently in. So you will see a "connected" icon when you are offline and a "disconnected" icon when you are Online.

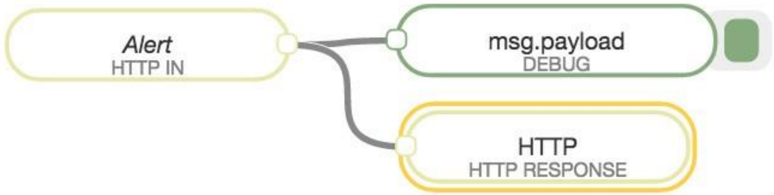
15. Once deployed, click on the tab to the left of the Inject node to initiate the flow. Click on the Debug panel at bottom to pull it up

and you should see an accepted message. If you look at your Temperature stream in M2X you'll see a single data point of 34 °C.

16. A function node is written using NodeJS and could check the values to make sure they're in your given bounds, or you can rely on M2X to send triggers. M2X integrates with many services, including IFTTT. We'll use the M2X triggers here. Grab an HTTP node from the Input section and drop it below the Inject node. Double-click on it to open it and name it "Alert", change the Method to POST, then set the URL to "/alert". Click OK to save it.

17. Connect up a Debug node to the right of the HTTP In node. Deploy your project.

18. Grab an HTTP RESPONSE node from the Output section and drop it below the Debug node. Connect the output of the HTTP node to the input of the HTTP RESPONSE node. The HTTP RESPONSE is required to close the trigger sender's connection. Your new nodes should look like this:



19. Deploy your project.

20. Pull up the Endpoints panel at the bottom and you'll see the URL for sending data to the Alert node labeled, "HTTPS (Alert)". Copy that URL. Then, open up M2X and the trigger you created by click on the pencil icon on the same line as the trigger. Paste that URL into the Callback URL text field and then click on the Save button.

21. Modify the Function node to set the temperature higher than 40, deploy the project, and then execute the Inject node by clicking on its tab. In the Debug panel in Flow you'll see a JSON object sent from M2X. You can modify the Function node to go back to a value less than 38, which will reset the trigger.

22. You can stop this node, preserving resources, by clicking "Run" in the left hand column. For example, you may need to do this because your Developer account supports a limited number of executing projects.

23. Click on "Sandbox"

24. Click on the "Stop" button on the upper right-hand corner. This will stop your flows, while retaining your code in Flow Designer's repository.