Programming Assignment: Using the IoT APIs in a Bluemix application

You have not submitted. You must earn 2/2 points to pass.

Deadline Pass this assignment by May 28, 11:59 PM PDT

Instructions

My submission

Discussions

Assignment goals

This assignment builds on the experience from lesson 3. Instead of using NodeRED, we will create a Node.js-based application that uses the IoT Foundation APIs. The application will run on Bluemix to receive temperature data from the SenseHAT device and issue commands when the threshold is crossed.

The solution to this assignment should be able to replace the Bluemix application you created in Node-RED during week 3, so the same message format is used:

```
{"d":{"temperature":32.8,"humidity":51.9}}
```

Your application should send a display command back to the device when the threshold temperature is crossed with the content:

```
{"screen" : "on"}
{"screen" : "off"}
```

How to complete the assignment

Create a Bluemix application using the Node.js buildpack. Use the attached skeleton for this lesson as a starting point. Remember to edit the manifest.yml by adjusting the name and host property. You may also copy the manifest from a Bluemix starter application.

Before submitting your solution, in order to obtain the credentials to the IoT platform service, you should use the VCAP_SERVICES environment variable. The grader environment has a service called "iotf-service" that you can extract them from.

bluemix-app-skeleton.zip

In server.js, implement an express application that connects to IoT Platform with the Node.js API for application developers. In service.js, create the control code for the SenseHAT device. Register for device events and keep track of temperature changes. If the temperature rises above or falls below the threshold of 29C, issue a device command to turn the warning on or off.

The relevant sections in the skeleton are commented as "TODO". Please stick to "display" as the command name and use the same JSON payload as in the previous lesson: { "screen": "off" } and {"screen": "on" }.

You need to retain the previous temperature reading sent from a device to ensure you only send a command when the temperature crosses the threshold value. In a production system you would typically use one of the services available on your chosen Cloud platform. In Bluemix you could use services such as the Session Cache, Redis or any of the database services to store the previous reading for devices. However, the grader used to mark the assignment does not have access to any external services, so your solution to this assignment should use an in memory solution to retain the previous reading. For this assignment It is acceptable to assume a single device will be using the application, so a single variable can be used to hold the previous temperature reading - however, you should understand that such a solution would be of limited use in a real world scenario.

When testing your solution the grader will use the following values:

deviceType : **SenseHAT**

device ID : **senb827eb7ddd6d**

event Type : **environment**

event format : **json**

You should ensure your solution works with events containing these values otherwise the grader will fail your solution.

What to submit

Submit your server.js as assignment6.js and your service.js as assignment7.js. Make sure to follow the skeleton code, as the grader uses the defined APIs to test your code. Note that the grader will not install new packages—you are limited to the ones listed in the skeleton’s package.json.

How to submit

When you're ready to submit, you can upload files for each part of the assignment on the "My submission" tab.