

IoT Experimental Learning

Week 5 Journal

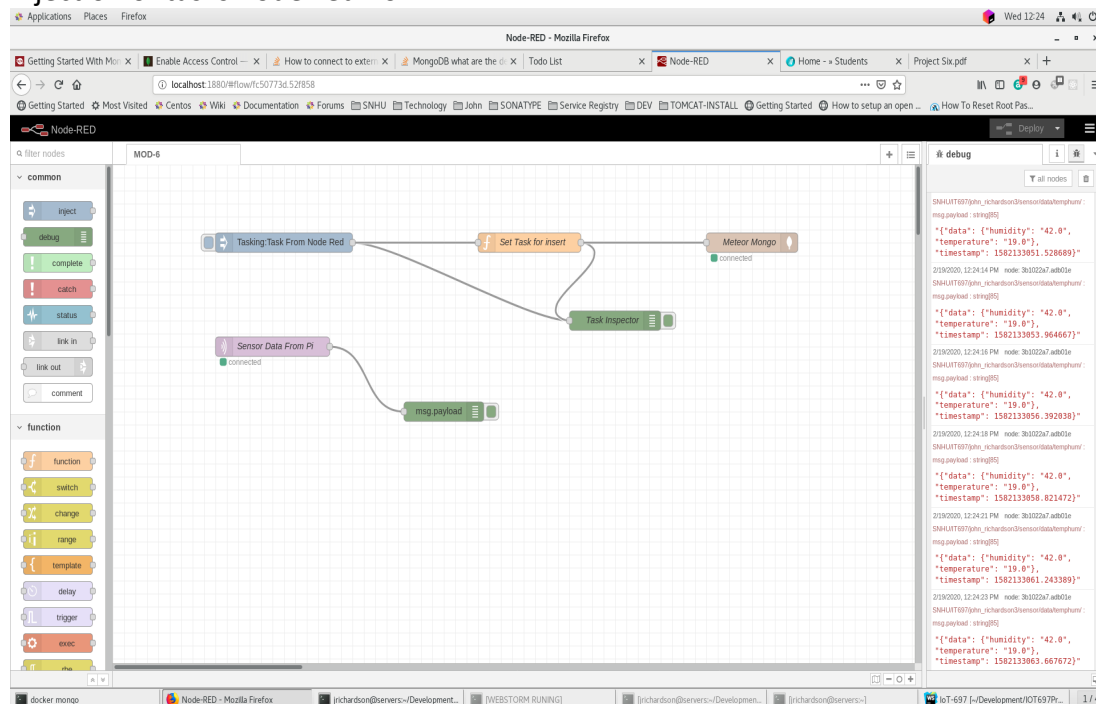
1. Describe the experience and what you hope to gain from participating in the experience.

- This week's assignment was a very easy for me, I found no issues while trying to complete assignment.
 1. Reviewed weekly assignment material.
 2. Completed Coding assignment

2. Provide an overview of tasks and key activities (training, discussions, labs, assessments, etc.) in which you were engaged during the week.

For week 6 I accomplished the following tasks in chronological order;

- ❖ **Wednesday February 19, 2020**, completed reviewing this week's assignment for the project.
 - I reviewed the Module 6 weekly assignment on the SNHU Brightspace.
<https://learn.snhu.edu/d2l/le/content/343560/Home>
 - Started the Module 6 Coding assignment at:
 - <https://learn.snhu.edu/content/enforced/343560-IT-697-X3325-OL-TRAD-GR.20TW3/Project%20Six.pdf>
 - Injection of tasks node-red flow



- Injected tasks in Docker Mongo Instance

```
File Edit View Search Terminal Help

2020-02-19T15:00:57.430+0000 I CONTROL [initandlisten] ** We suggest setting it to 'never'
2020-02-19T15:00:57.430+0000 I CONTROL [initandlisten]
...
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

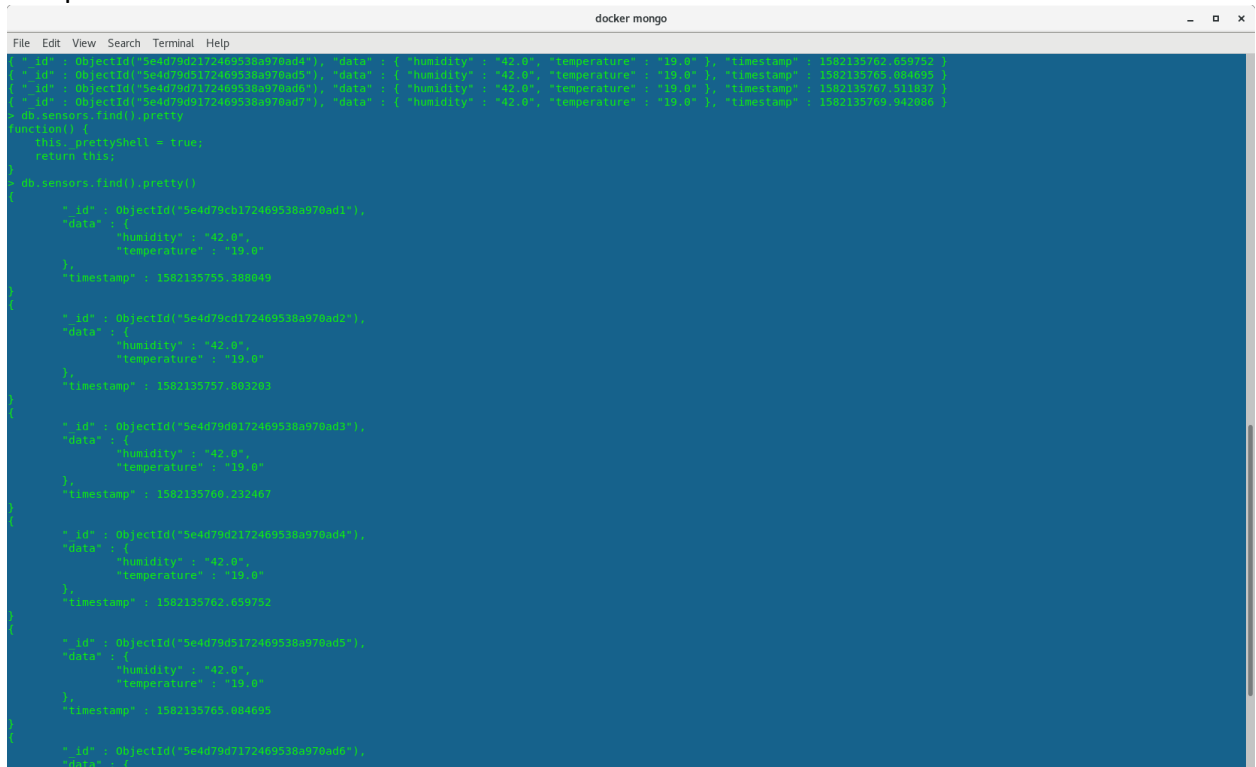
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
...

> show dbs
admin 0.00000
config 0.00000
local 0.00000
> show dbs
admin 0.00000
config 0.00000
local 0.00000
> use meteor
switched to db.meteor
> show collections
meteor.accounts loginServiceConfiguration
tasks
tasks
> db.tasks.find()
{ "_id" : "7ajghdXDH4QnSY7P5", "text" : "Task For External MongoDB", "createdAt" : ISODate("2020-02-19T15:55:26.226Z"), "owner" : "KhocSV7DyB9rYr3h", "username" : "JhRichardson" }
> db.tasks.find()
{ "_id" : "7ajghdXDH4QnSY7P5", "text" : "Task For External MongoDB", "createdAt" : ISODate("2020-02-19T15:55:26.226Z"), "owner" : "KhocSV7DyB9rYr3h", "username" : "JhRichardson" }
{ "_id" : "objectid('944d3e7c172449338a978ac9')", "text" : "Task From Node Red", "createdAt" : ISODate("2020-02-19T18:12:44.759Z"), "owner" : "KhocSV7DyB9rYr3h", "username" : "JhRichardson" }
>
```

- Updated node-red flow with pi injection from MQTT for temp-hum sensor

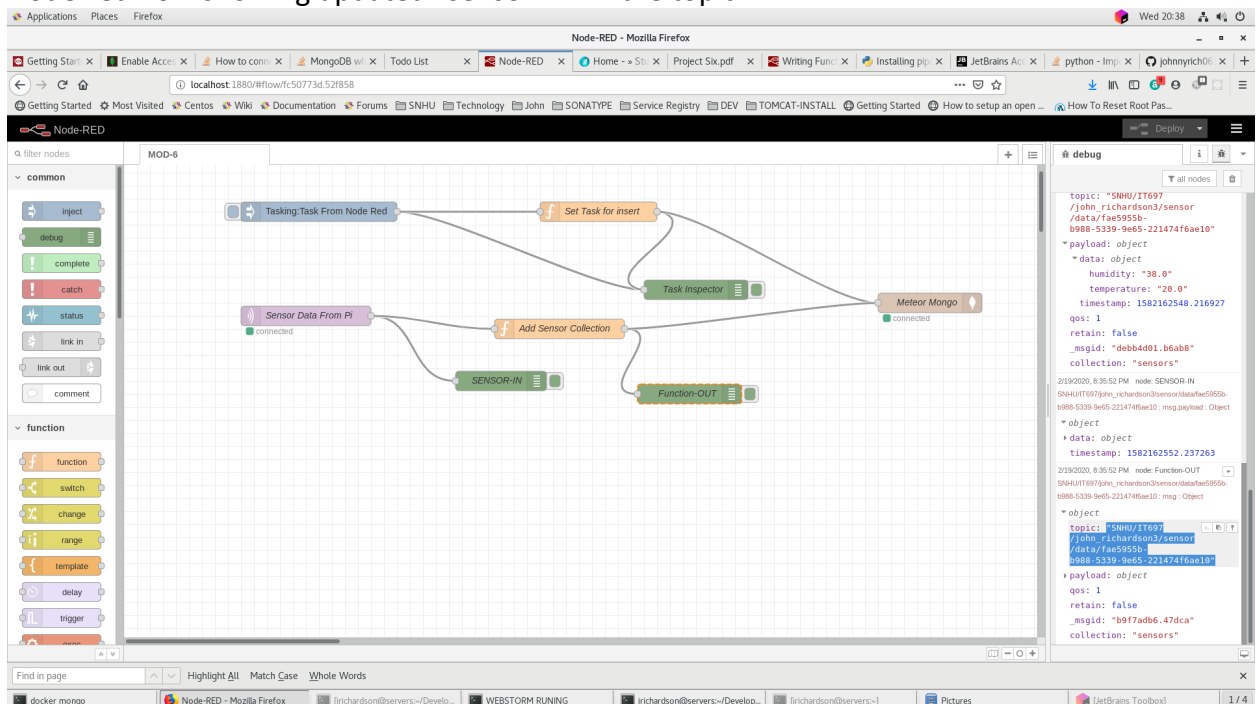
The screenshot displays the Node-RED web interface within a Mozilla Firefox browser. The browser's address bar shows the URL 'localhost:1880'. The interface includes a top navigation bar with various tabs and a main workspace for creating and editing flows. On the left, there are panels for 'common' and 'function' nodes. The central workspace contains a flow diagram with several nodes: 'Tasking-Task From Node Red', 'Set Task for insert', 'Task Inspector', 'Sensor Data From Pi', 'Add Sensor Collection', 'Meteor Mongo', 'SENSOR-IN', and 'Function-OUT'. The right sidebar features a 'debug' console showing a series of JSON messages, including data objects with timestamps and topics. The bottom of the browser shows the taskbar with several open applications, including Docker, Node-RED, and a terminal window.

➤ Temp-Hum data in sensors collection

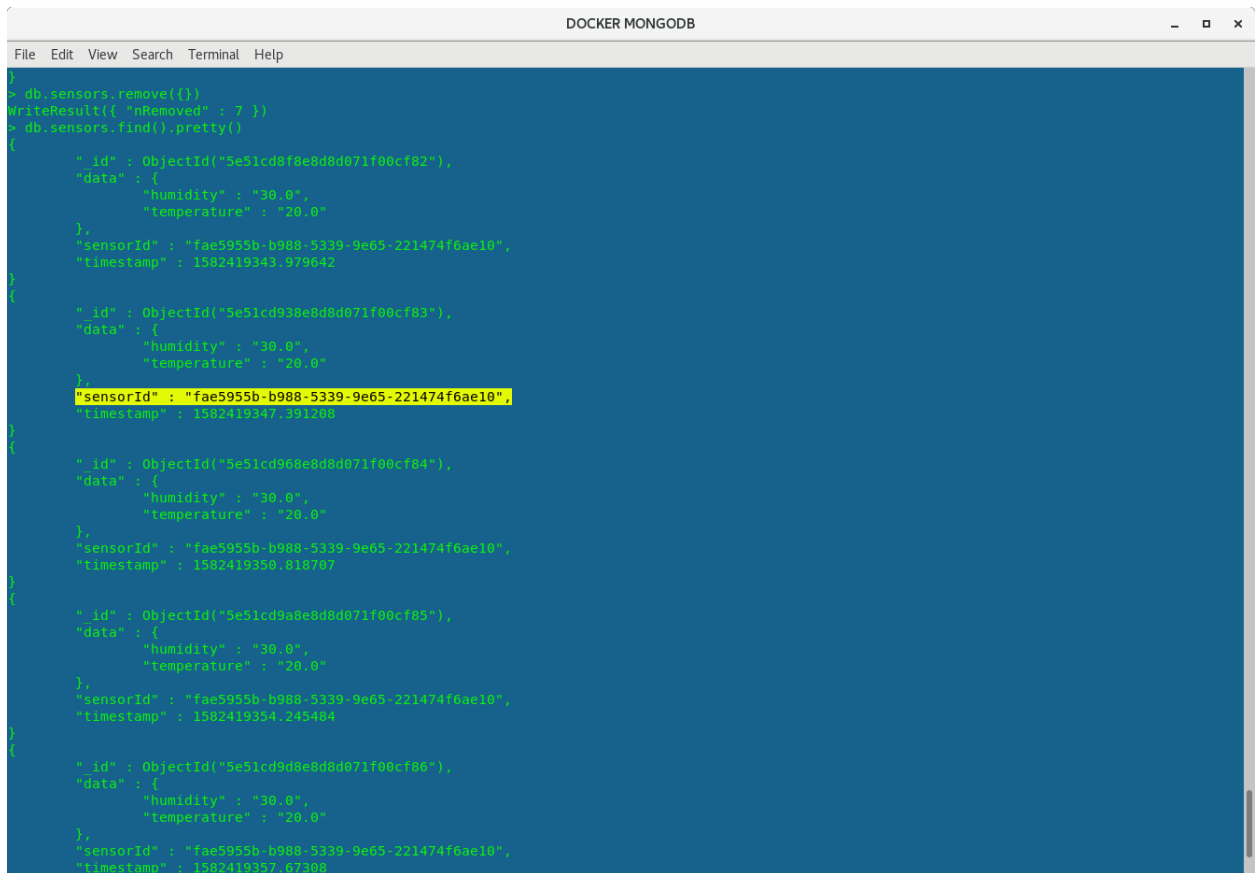


```
File Edit View Search Terminal Help
docker mongo
{ "id" : ObjectId("5e4d79d2172469538a978ad4"), "data" : { "humidity" : "42.0", "temperature" : "19.0" }, "timestamp" : 1582135762.659752 }
{ "id" : ObjectId("5e4d79d5172469538a978ad5"), "data" : { "humidity" : "42.0", "temperature" : "19.0" }, "timestamp" : 1582135765.084695 }
{ "id" : ObjectId("5e4d79d7172469538a978ad6"), "data" : { "humidity" : "42.0", "temperature" : "19.0" }, "timestamp" : 1582135767.511817 }
{ "id" : ObjectId("5e4d79d9172469538a978ad7"), "data" : { "humidity" : "42.0", "temperature" : "19.0" }, "timestamp" : 1582135769.942888 }
> db.sensors.find().pretty()
function() {
  (this.prettyshell = true);
  return this;
}
> db.sensors.find().pretty()
{
  "id" : ObjectId("5e4d79d172469538a978ad1"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135755.388849
}
{
  "id" : ObjectId("5e4d79d172469538a978ad2"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135757.805293
}
{
  "id" : ObjectId("5e4d79d172469538a978ad3"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135760.212467
}
{
  "id" : ObjectId("5e4d79d2172469538a978ad4"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135762.659752
}
{
  "id" : ObjectId("5e4d79d5172469538a978ad5"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135765.084695
}
{
  "id" : ObjectId("5e4d79d7172469538a978ad6"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135767.511817
}
{
  "id" : ObjectId("5e4d79d9172469538a978ad7"),
  "data" : {
    "humidity" : "42.0",
    "temperature" : "19.0"
  },
  "timestamp" : 1582135769.942888
}
```

➤ Node-red flow showing updated “sensor ID” in the topic.



- I also added an attribute for collecting “sensor_id” in the msg.payload, came in handy in mod 7.



```
DOCKER MONGODB
File Edit View Search Terminal Help
> db.sensors.remove({})
WriteResult({ "nRemoved" : 7 })
> db.sensors.find().pretty()
{
  "_id" : ObjectId("5e51cd8fb8e8d8d071f00cf83"),
  "data" : {
    "humidity" : "30.0",
    "temperature" : "20.0"
  },
  "sensorId" : "fae5955b-b988-5339-9e65-221474f6ae10",
  "timestamp" : 1582419343.979642
},
{
  "_id" : ObjectId("5e51cd938e8d8d071f00cf83"),
  "data" : {
    "humidity" : "30.0",
    "temperature" : "20.0"
  },
  "sensorId" : "fae5955b-b988-5339-9e65-221474f6ae10",
  "timestamp" : 1582419347.391208
},
{
  "_id" : ObjectId("5e51cd968e8d8d071f00cf83"),
  "data" : {
    "humidity" : "30.0",
    "temperature" : "20.0"
  },
  "sensorId" : "fae5955b-b988-5339-9e65-221474f6ae10",
  "timestamp" : 1582419350.818707
},
{
  "_id" : ObjectId("5e51cd9ae8d8d071f00cf83"),
  "data" : {
    "humidity" : "30.0",
    "temperature" : "20.0"
  },
  "sensorId" : "fae5955b-b988-5339-9e65-221474f6ae10",
  "timestamp" : 1582419354.245404
},
{
  "_id" : ObjectId("5e51cd9d8e8d8d071f00cf83"),
  "data" : {
    "humidity" : "30.0",
    "temperature" : "20.0"
  },
  "sensorId" : "fae5955b-b988-5339-9e65-221474f6ae10",
  "timestamp" : 1582419357.673008
}
```

- Final node-red flow for Module 6

