

TECH3812

MQTT, Node-Red, and MySQL

Lab #11

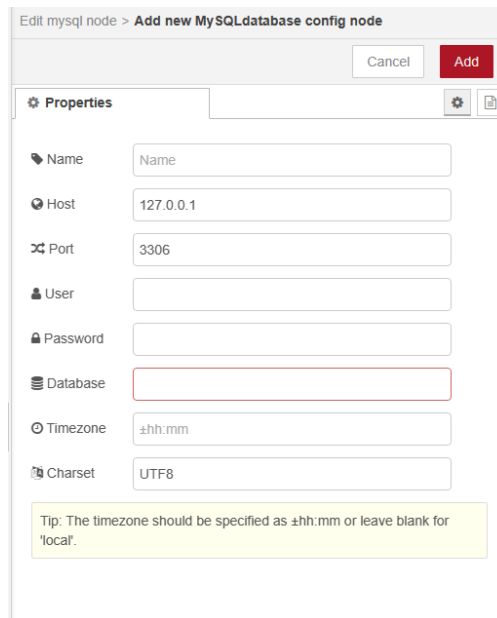
Background: We have seen how MQTT messages can be used with Node-Red but, other than the graphed data, no data is stored long term. To store data long term, we can connect Node-Red with MySQL database to store data for future analysis.

Setup: first we must add the *node-red-node-mysql* pallet to Node-Red. To do this go to your node-red flow and click on the ≡ at the upper left of the screen, then *Manage Pallet* then click on the *Install Tab*. In the search bar, type in *node-red-node-mysql* and click on the *install* box.

Once completed, this will add the following icon to the pallet:



Procedure: Start a new flow by using ≡ | Flows | Add. On the new Flow, add the mysql icon. In the window, add a new mysql database. When you do so, the following screen will appear:

The screenshot shows a dialog box titled "Edit mysql node > Add new MySQLdatabase config node". At the top right, there are "Cancel" and "Add" buttons. Below the title bar is a "Properties" section with a gear icon and a refresh icon. The form contains several input fields: "Name" (empty), "Host" (127.0.0.1), "Port" (3306), "User" (empty), "Password" (empty), "Database" (empty), "Timezone" (±hh:mm), and "Charset" (UTF8). At the bottom, there is a yellow tip box that reads: "Tip: The timezone should be specified as ±hh:mm or leave blank for 'local'."

Make the host, the IP address of the Raspberry PI (192.168.0.14), leave the port number as 3306 (default port of the MySQL Server), the User will be mqtt, the password for the MySQL user will be given by the instructor. Lastly the database is named mqtt. (all other boxes can be left as their default/blank).

Now a little bit on the MySQL Database we will be using. The name of the database is "mqtt" and within the data base is a table called "student". This database is defined as follows:

Server: localhost:3306 » Database: mqtt » Table: student

Table structure Relation view

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	index	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	student	text	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	3	topic	text	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	4	value_num	float		Yes	NULL		
<input type="checkbox"/>	5	value_text	text	utf8mb4_general_ci	Yes	NULL		
<input type="checkbox"/>	6	timestamp	timestamp		No	current_timestamp()		

There are 6 items stored:

- index – is a number of the record that is auto generated by mysql (and counts sequentially)
- timestamp – is the time and date when the record was created (again auto generated by mysql)
- student – will be filled in by you and should be your email name (eg my email is dekohn@memphis.edu, so I would use dekohn).
- topic – will be the mqtt topic
- value_num and/or value_text – will be the data sent by the mqtt broker (we need to save a text response in the value_text, and numeric values need to be stored in value_num)

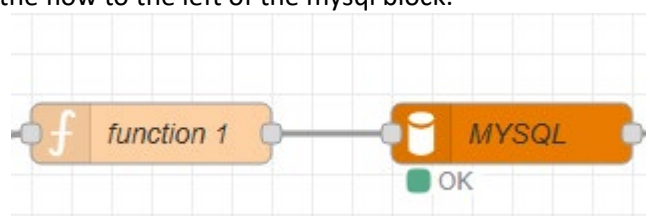
A typical 'store info into the database' will look like:

```
INSERT INTO student (value_num,value_text,student,topic) VALUES (-1,NULL,'dekohn','acel/x');
```

Notes: The “INSERT INTO student” tells MySQL to store the data in the student table. The (value_num, value_text, student, topic) define the order which you will send the information to the server and what column each piece of data will be placed in (order is irrelevant, so long as the values are in the same order as the column names). The keyword VALUES says put these values into those columns. Then the values, in order are placed within () and text must be in single quotes and the values are separated by commas. Note NULL is saying store nothing in this column (since acel/x sends back a value, not a string we are leaving the value_text blank).

Since we are going to store all the MQTT broker data we previously displayed in the database, we will need to get to pull the topic name and the value from each message received. To do this we will use a function block and a little bit of java script.

Place the function block on the flow to the left of the mysql block:



In the configuration of the function block, go to the “On Message” tab. This is where we will type the java script to create the command to send to the mysql database to store the new data.

```

var value1=msg.payload;
var value2=msg.topic;
if (isNaN(value1))
// it's a string
{
  msg.topic = "INSERT INTO student (value_num,value_text,student,topic) VALUES (NULL,'" + value1 +
  "','dekohh','" + value2 + "'");
}
else
// it's something else
{
  msg.topic = "INSERT INTO student (value_num,value_text,student,topic) VALUES (" + value1 +
  ",NULL,'dekohh','"+value2+"')";
}
return msg;

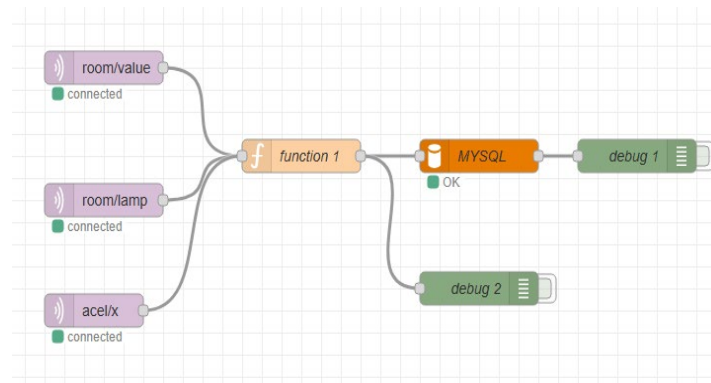
```

This code takes the message payload from the mqtt in blocks and breaks out the value (payload) and the topic (mqtt topic name) and places them in variables. Then it builds the string of the mysql command to place that data into the table called "student". Please replace the string 'dekohh' with your email as discussed above.

Watch the double and single quotes, it is important not to mix them up, or you will get an error when you send the command.

Since the javascript takes any topic and value, you can tie all the mqtt points into the one function box.

Here, is a sample of the flow:



the debug boxes are there to help fix issues that might occur.

For your's please include all of the following mqtt topics:

- acel/x
- acel/y
- acel/z
- room/value
- room/lamp

Please have the instructor verify that you are placing values into the database (demo) and submit a json file containing ALL FLOWS to the class file submission system.