

i-UG Open Source  
Education for IBM i

# Node-Red

Master Lesson Control





i-UG Open Source education for IBM i  
September 2023  
Node-RED

Contents

1) Introduction	5
2) The tools	6
a. POWER	6
b. DATABASE	6
c. Node.js	6
d. Node-RED	6
e. The “Lights”	6
f. The Sensors	6
3) The environment	7
4) A Foreword on this course...	8
a. Simple stuff	8
b. Node-RED understanding and discipline	8
c. “There is more than one way to skin a cat”	8
d. Neat or not:	9
e. Notes and documentation	10
5) Getting in to the course	11
6) Base Lesson – Basics of Node-RED	13
a. The Editor	13
b. Core Nodes	13
c. Working with Messages	13
d. Writing Functions	13
7) Starting Node-RED, working with Node-RED flows in a browser	14
a. Starting Node-RED	14
b. Working with Node-RED through your Browser	15
c. Core nodes	16
d. Managing nodes and flows	16
e. The Deploy button	17

f.	Basic Functioning	17
g.	Adding New Nodes	18
h.	The Node-RED UI - Dashboard	19
8)	The path we will follow	21
9)	i-UG Lesson 1 - Initial Lighting Control	22
10)	i-UG Lesson 2 – Extended Lighting Control	23
11)	i-UG Lesson 3 – Data Controlled Lighting	24
12)	i-UG Lesson 4 – Engaging with the Node-RED UI -Inputs	25
13)	i-UG Lesson 5 – Engaging with the Node-RED UI – Outputs	26
14)	i-UG Lesson 4 – Speaking to the world – e-mail	27
15)	i-UG Lesson 7 – Reading from the DB2 Database	28
16)	i-UG Lesson 8 – Writing to and Updating a DB2 Database	29
17)	i-UG Lesson 9 – MQTT – Connecting the world of IoT	30
18)	The Brewery – Bringing it all together	31
19)	<b>Appendix A</b> - Getting in to our Node-RED	32
a.	Starting Node-RED	32
b.	Working with Node-RED through your Browser	33
c.	Accessing the UI/Dashboard	34
20)	<b>Appendix B</b> - i-UG Potential Additional Modules	35
a.	SQL Refresh	35
b.	Talking to Watson AI	35

## 1) Introduction

Welcome to the i-UG program designed to bring new talent to the IBM i platform by leveraging the Open Source capabilities of the platform.

In this Node-RED based education, you will be introduced to a variety of tools which will enhance your ability to start producing useful applications that can be applied directly to real business.

Along the way, the education will focus your understanding of the Internet of Things (IoT), the use of databases, the decision-making process, the realisation of connectivity between people, data and things, as well as having an enjoyable journey through the Open Source world and conversing with an AI.

All of the processing power for the education and the database will be provided. You will be using an extremely powerful IBM POWER server, running the IBM i operating system and will store and recover data from the in-built DB2 Database. It is this area where there is a high demand for skills of this sort and opens the student into a very large world of large companies with requirements.

We will start this course by having you set up flows that switch on a light in a remote location (nr Manchester), in which you are in control. We will set the light to show random colours, then we will choose and manage the colours, and finish by making the light conform to some rules we set.

Whilst you will be familiar with the common Apps that provide this all as part of , say, a bedroom lighting kit, the difference will be that YOU will be in charge and you will understand how this App is working.

We will then move on to making decisions based on data within our database. As you move into industry, this is the single most key piece of knowledge you will need! It will start simply, but will escalate to more complex algorithms as you move forward. When you are in business, the sky is the limit when you are trawling through data in order to make the right decision. Indeed sometimes, you may need the help of a friendly AI to provide you with vital information that will help you and your business make profit and deliver goods.

Along the way, we will need to keep people informed and that will take the shape of reports, dashboards and e-mail (Maybe other communication and social media if the chance crops up).

Finally, you will be presented with a challenge... and it involves beer!

You will be presented with a basic brewery process line, a line that is, to say the least, difficult to manage. But, you will have learned what you need in order to change the database values and constantly monitor in order to make the beer flow smoothly. A good prize.

At the end of the course, providing the brewery is operational, you will be awarded a certificate from i-UG, the largest IBM i User group of companies in the UK, which will be presented before the member companies.

So let's start with an introduction to the tools.

## 2) The tools

### a. POWER

The main 'tool' is the incredibly powerful IBM POWER system. At this time, it is envisaged that the course will run on a POWER9 system, but things will progress.

Whilst much of what we will cover can be done on the regular Intel servers, it is a distinct feather in your caps to state that you did all this on an IBM POWER server running IBM i.

### b. DATABASE

Again, there are other databases, but none that are integrated with the hardware in the way that DB2 for i is. This strong link gives the IBM i operating system a wholly unparalleled advantage in security and delivery. Indeed, no viruses.

### c. Node.js

Node.js is one of the most widely used programming languages out there today. Whilst this whole education is not limited to just this language, we will be using this and JavaScript as the base. It may well be that you use some other language to fix problem, either in the future or in the crucial Brewery test, but that would be your decision. The Flows (See later) will still flow.

### d. Node-RED

Node-RED is basically a flow-based development tool for visual programming developed by IBM, originally for wiring together hardware devices, APIs and online services as part of the Internet of Things (IoT). Node-RED provides an excellent web browser-based flow editor, which can be used to show the 'Flow' of what you are trying to achieve. Node-RED has progressed into a much more powerful tool than just simple IoT. Now used extensively for real business, interrogating and working with established databases, linking with AI's and communicating with the world. It is an excellent way of educating programmers as well as users, and serves as a great tool to get started.

### e. The "Lights"

We have chosen to use the Phillips Wi-Fi system to take over the switching of the lights. It comprises a Hub and a Light Strip. This will have been configured already and you will just be sending instructions to the Hub over MQTT. However, we will show you how we can work with devices of this type, and of course, a switch is a switch. There are many similar devices available on the market that would be valuable to business to operate, say, a production line. Same technique...

### f. The Sensors

We will be using an NCD Industrial Temperature and Humidity sensor in the main, but we will also introduce a very small, amazingly cheap Raspberry Pi Pico setup too. As with the lights, we will converse with it over an MQTT Broker.

### 3) The environment

Each team will be presented with a running, operational environment and the credentials with which to log into this environment. In some ways, it doesn't matter where this environment is, but just for reference, it is currently in Birmingham (well, most of it is). You will enter into our controlled network(s), but you will also be given access to the devices (Lights, switches, cameras, sensors) through web connections and MQTT.

Each team will have a profile for the Open Source environment, a profile for the IBM POWER system and the DB2 Database, work areas for the Node-RED flows, a Library on the POWER system assigned to them containing Database tables. You will have a Master document (this document) with base information for the various steps in the education, and further documentation for each lesson in the course.

You will also be supplied with links to a GitHub account. This will have the individual lessons in PDFs, provide a way to report issues to us (i-UG) that you may run in to as you progress through the course, examples of what your flow should look like etc.

There will also be the answers... just in case you miss something.

A word of caution...

As you progress through the lessons, if you reach straight for the answers rather than working through the lesson, you will have NO CHANCE of understanding the Brewery, which is the object of the course and what you will be graded on.

## 4) A Foreword on this course...

### a. Simple stuff

During this course, you will cover some items that may seem very 'simplistic'. Indeed, when we start with switching on and off a light in a remote location... you may feel that this has been easily conquered already and can be ordered for a few pounds on e-bay and delivered tomorrow morning.

BUT, now you will see HOW this is achieved and will be able to tailor the usage to your own needs. Moreover, once you understand that this is simply a 'Switch' for the light, then this could easily be the switch that starts or stops a factory process, the switch that opens a door or restricts access... and on and on.

The cheap e-bay solution is now just as it seems, an enjoyable children's bedroom gimmick, whereas what you will be able to achieve under your own control can be applied to many situations and will be a valuable asset to business!

### b. Node-RED understanding and discipline

Throughout this course, you will be able to get help from Mr Google on just about ANY aspect of what we are doing, and that's fine. Node-RED is a widely used application and there are many contributors that have stumbled over all of the issues you will face on this course. Just keep in your mind that Google answers are not always totally reliable and can be contradictory at times. It is worth looking at a few 'answers' before you deploy their 'solution', but it is likely that this is all good information.

### c. "There is more than one way to skin a cat"

Now, as you follow this course, you will see guidance and examples that do not always fit with what you see on Google. Sometimes this is just preferences that vary between programmers. This does not mean that what you see is wrong, just a slightly different preference for executing code to meet a goal.

At other times, we have taken a deliberate approach in order to get you thinking about, say, managing variables, in a way that you might do if you were coding but without Node-RED. It is designed to make you think about managing your working data and not always relying on Node-RED to manage these for you.



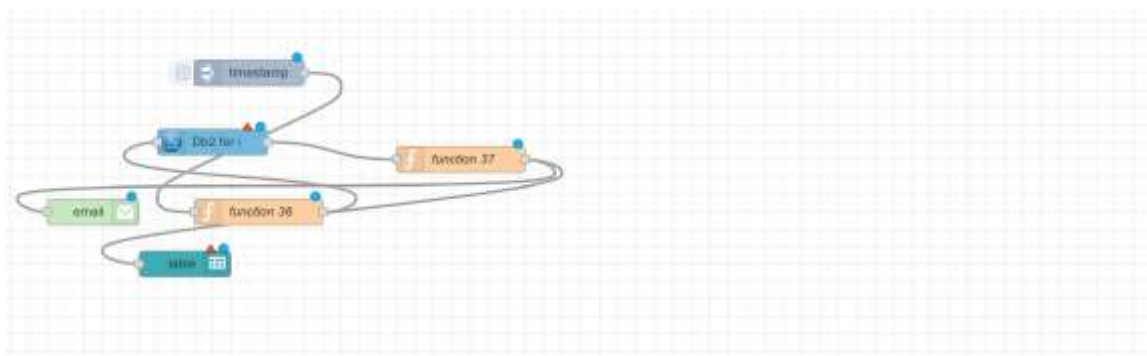
#### d. Neat or not:

As we build our flows, we will think about the 'Story' that this process is telling. If we construct this properly, it becomes immensely easier to explain the process to others, immensely easier to come back and revise or repair the process, and just easier to pass on to others. It is all about documentation.

Consider this: -

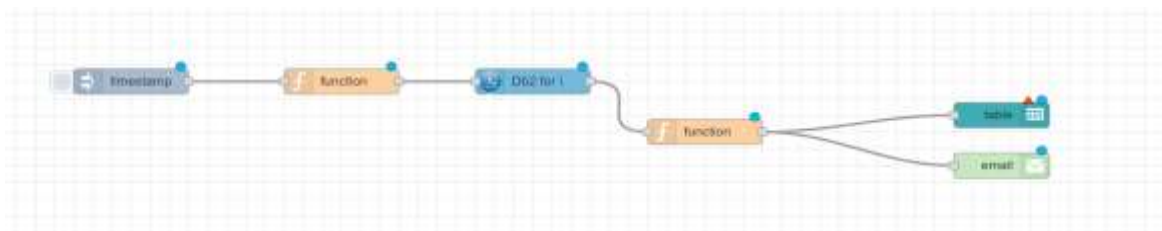
Here is an example where we want to reach into a database, pick up some information, decide if it meets our criteria and advise the correct user about the situation. A normal and useful process.

Using Node-RED, a flow like this actually works: -

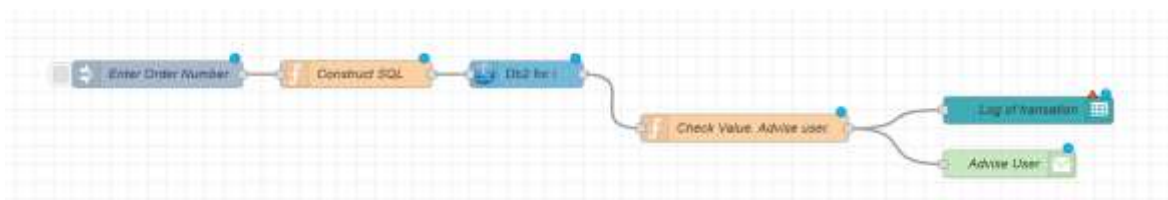


...but it is not easy to follow.

Making it more logically laid out (Very simple in Node-RED), it starts to resemble a sensible flow. Believe it or not, this is exactly the same as above, but in a sensible layout.



If we also keep our documentation clear by filling in the 'name' boxes - again this is an easy thing to do in Node-RED (and in any language really) - we can let any non-technical person see just what is going on! :-



Seriously, these are ALL the same process, just a little documentation makes it all so much easier to understand.

### e. Notes and documentation

It is OK to make a note of what you are doing.

It helps when you look back, it helps when others are looking at it... and it looks good.

In Node-RED, blank lines are like spaces. They do nothing. But, if anything sits in any of these blank spaces or lines, then the editor is looking to syntax check it etc.

Everything except for a pair of slashes... `//`. `//` is the standard to define a comment, so everything that comes after that for line, until you hit return, will be turned green and will be free-format text. Take a look at this example: -



```
1
2 // Define the variables for the 'SOIL' station to be
3 // used throughout the flow. Set to start values.
4
5 var valuef = flow.get('valuef') || 0;
6 var lossf = flow.get('lossf') || 0;
7 var valueex = flow.get('valueex') || 0;
8
9 flow.set('valuef', 0)
10 flow.set('lossf', 0)
11 flow.set('valueex', 10)
12
13 let payloadout = `(select * from mike.brewctl where PROCESS`
14
15 msg.payload = payloadout
16 return msg;
```

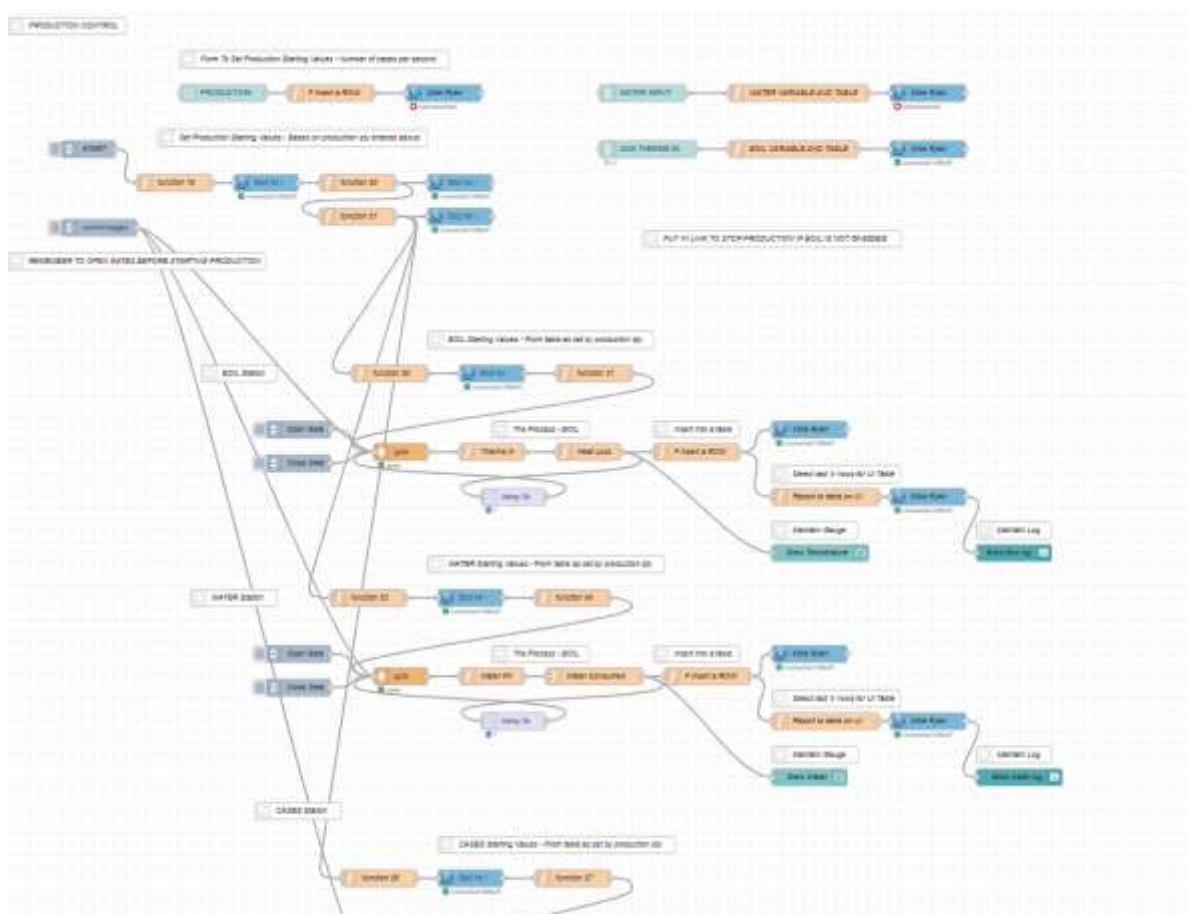
## 5) Getting in to the course

You will have been given a set of credentials for this education set. In the examples, you will see in the documentation, we will be using User: STUDENT and Password: STUDENT to log in, but you will use your own credentials of course. A section on setting up the environment is also available in Appendix A – Node-RED Education Environment Setup.

Just to keep you interested... let's look ahead...

Whilst we have a number of building-block steps covering basic principles and simple tasks, these are there so that you can learn to handle Node-red adequately before you move ahead onto more complex work. It does eventually get VERY interesting.

Here is just part of the flow that you will eventually be working on: -



And just a sneak preview at the Dashboard...:



This is where you will control a whole process – in this case, brewing beer!!! More on all of this later as we start the project 'Challenge'.

...but for now...

We are going to start with some introductory videos and some basics, then we will move on to the Internet of Things (IoT) by switching lights on and off...

## 6) Base Lesson – Basics of Node-RED

Let's take a look at some of the basics of Node-Red.

Here are some Stock video's and education sets for basically using Node-RED: -

### a. The Editor

First of all, the editor environment itself: -

[Editor Guide : Node-RED \(nodered.org\)](https://nodered.org/docs/guides/editor) (1 hour of video's)

### b. Core Nodes

There are a number of nodes that form the very basis of using Node-Red. We will embellish these and expand other nodes later in the documentation, but you need to understand these first: -

[The Core Nodes : Node-RED \(nodered.org\)](https://nodered.org/docs/guides/core-nodes)

### c. Working with Messages

A fundamental aspect of Node-RED is the way that it passes information – a message (msg.) between Nodes. We will look at many ways of doing this as we progress, but again, these are fundamentals: -

[Working with messages : Node-RED \(nodered.org\)](https://nodered.org/docs/guides/working-with-messages)

### d. Writing Functions

It will soon become apparent that some of the real workload and flexibility is delivered through one particularly useful node, the Function node. So, lets' explore just some of its capabilities: -

[Writing Functions : Node-RED \(nodered.org\)](https://nodered.org/docs/guides/writing-functions)

## 7) Starting Node-RED, working with Node-RED flows in a browser

### a. Starting Node-RED

We will be running Node-RED in a Secure Shell in the IBM i PASE environment. This means that we will initiate it from a SSH which we can drive from a Terminal Application or a PowerShell application. The result will be the same, so you choose the application that you have.



Open your terminal app:

Terminal,



PowerShell

First, we need to gain access to the IBM POWER system, so we will enter your User security credentials. Our IBM i POWER system is at **iug3. rowton. it** so at the prompt, we enter: -

```
ssh [STUDENT]@iug3. rowton. it    [Use your Student User Name]
```

We will be prompted for a Password, so enter your password. Here we will enter 'Student'

```
password : [STUDENT]    [Use your Student Password]
```

If you have been accepted (If you have put in the correct User and Password), you will now be logged in to the IBM POWER System. The Prompt will change to **-bash-5. 1\$**

We can now start Node-RED running. At the -bash-5.14 prompt, type: **node-red**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Nryan> ssh student_02@iug3.rowton.it
student_02@iug3.rowton.it's password:
-bash-5.1$ node-red
22 Sep 10:24:28 - [info]
Welcome to Node-RED
=====
22 Sep 10:24:28 - [info] Node-RED version: v3.1.0
22 Sep 10:24:28 - [info] Node.js version: v18.17.1
22 Sep 10:24:28 - [info] OS/Arch: 7.4 ppc64 BE
22 Sep 10:24:30 - [info] Loading palette nodes
22 Sep 10:24:32 - [info] Settings file : /home/STUDENT_02/.node-red/settings.js
22 Sep 10:24:32 - [info] Context store : 'default' [module=memory]
22 Sep 10:24:32 - [info] User directory : /home/STUDENT_02/.node-red
22 Sep 10:24:32 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Sep 10:24:32 - [info] Flows file : /home/STUDENT_02/.node-red/flows.json
22 Sep 10:24:32 - [info] Server now running at http://127.0.0.1:18802/
22 Sep 10:24:32 - [warn]

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
```

Take note of  
this port  
number...

Your Terminal should now look something like this: -

## b. Working with Node-RED through your Browser

Now that Node-RED is running in the server shell, we can access it through a browser.

Open your browser. We will key in a URL that looks very like the URL in the picture below, but we will use the Socket that has been assigned to us. The Socket number will be in the highlighted section of the Terminal above. In this case, we would use-

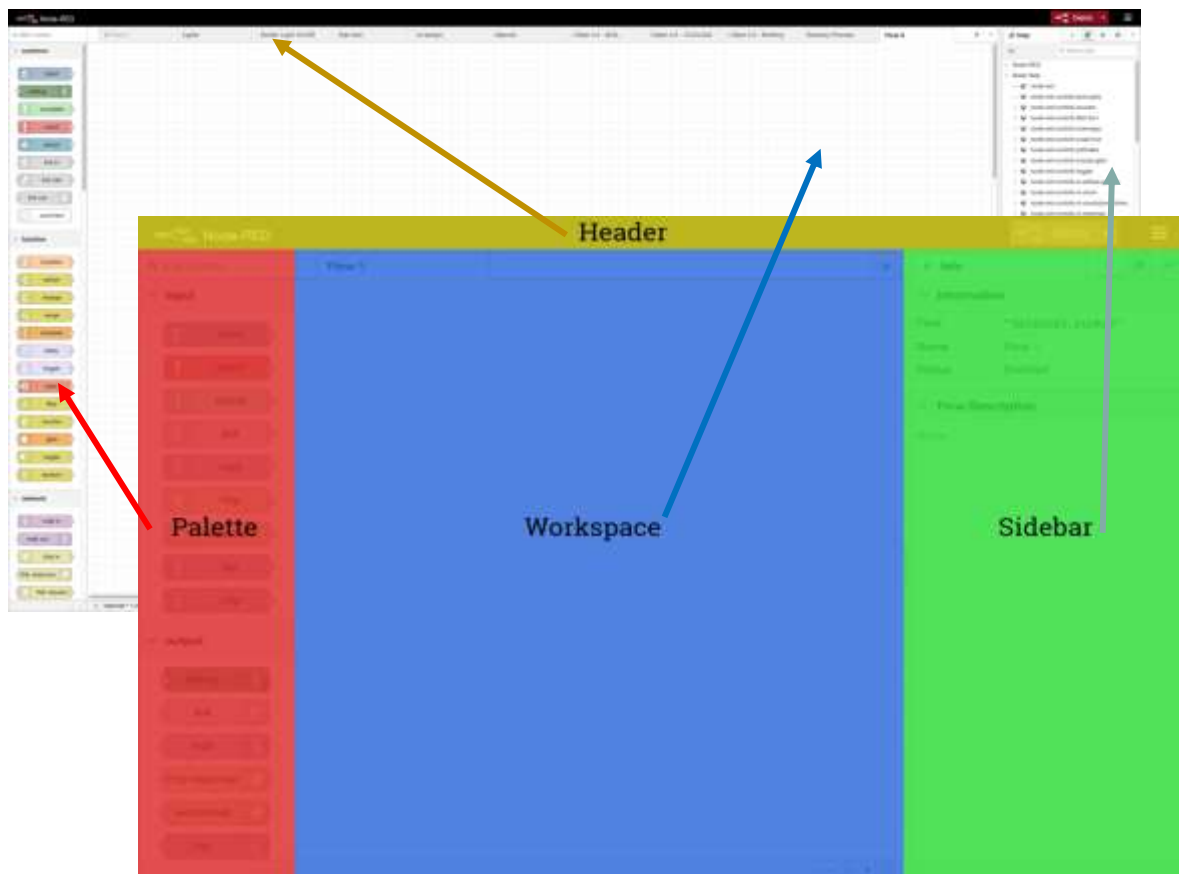
<http://iug3.rowton.it:18802>

iug3.rowton.it:18802

You will then be presented with the main Node-RED screen.

*N.B. The port number is basically '188' + your Student Number.  
So, for Student\_06, it would be 18806*

The Node-RED browser interface consists of 4 main elements. The Header, the Workspace, the Palette and the Sidebar: -



This next section of the documentation is now just a reminder/overview of what you have learned from the introduction Video's. It can serve as a reference point and is worth just reading through before you start with the lessons. Having the browser open will help understanding, but we will only be using it for reference just now.



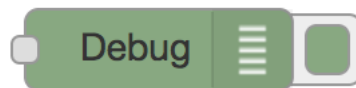
### c. Core nodes

All of the nodes below can be found in the palette section and dragged onto the workspace.

- **Inject** – This is a manual trigger for a flow. It can be configured to automatically trigger in intervals, and it can have its PAYLOAD and TOPIC properties changed across a flow or globally.



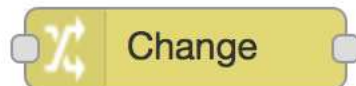
- **Debug** – This will display messages within the sidebar under the debug tab. These messages will consist of the message received from the node it's attached to.



- **Function** – This node allows JavaScript to be run against the messages that are passed through it.



- **Change** – This node can be used to modify a message's properties. They can use several operators such as: Set, Change, Move and Delete.

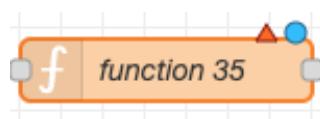


- **Switch** – this node allows messages to be routed to different branches of a flow based on a set of rules against each message: Value, Sequence, Expression (JSONata) and Otherwise.

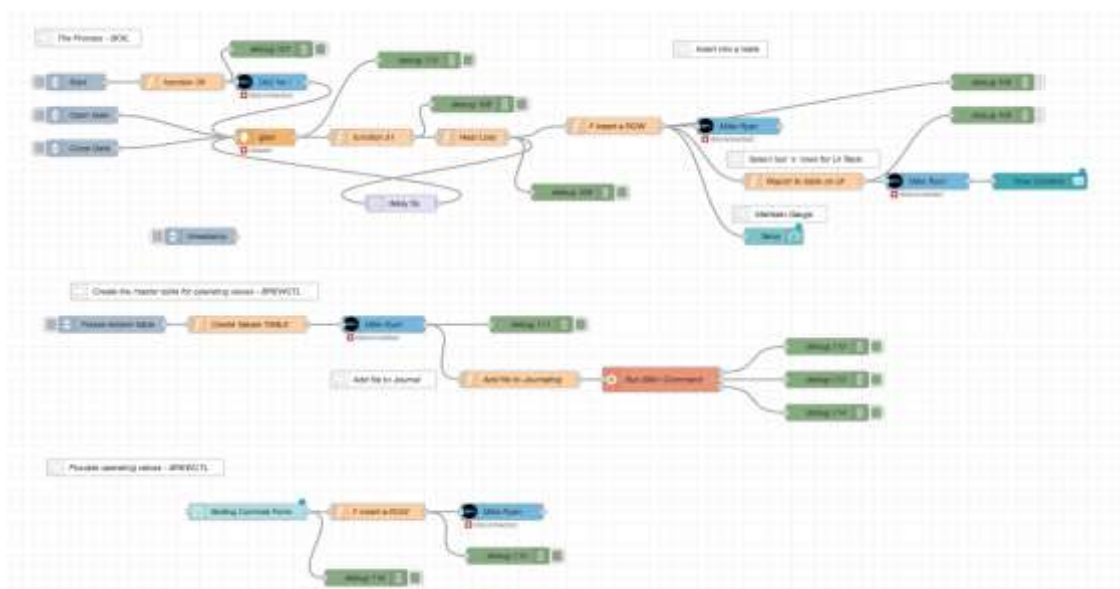


### d. Managing nodes and flows

If a node has a red triangle above it, it means that the code within has an error causing it to not function. The blue circle means that the node has not yet been deployed.




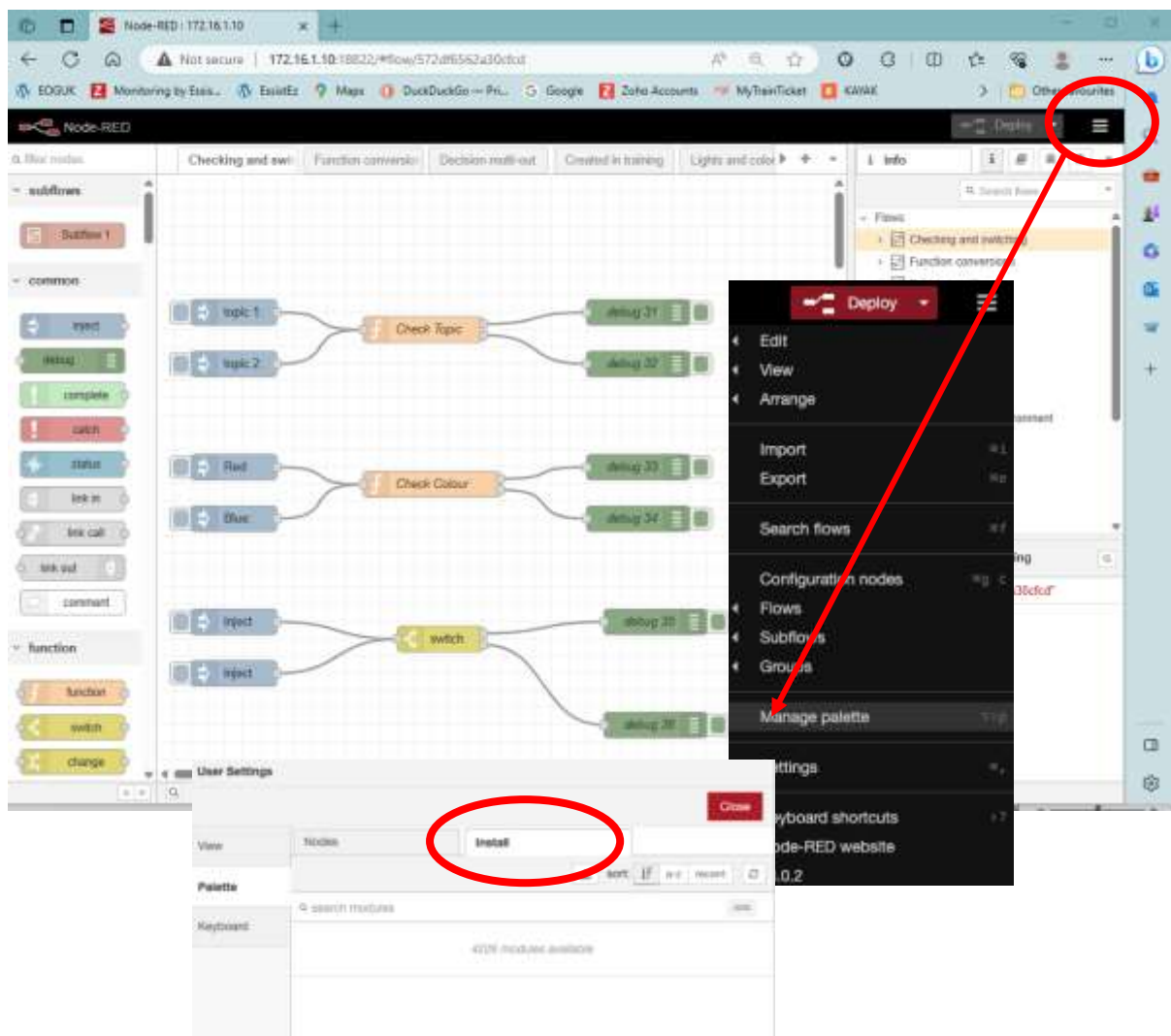




### g. Adding New Nodes

Node-RED comes with a wide selection of nodes already installed and in the Palette on the left side of the screen. Occasionally, you may want to solve a problem by importing a new node that has the functions you need.

If you need to add a new node, you can easily install it by clicking the menu icon , then clicking **manage palette**, then click on the **install** tab and type in the node you need.



A vast selection of nodes and their functions can easily be found on <https://flows.nodered.org>.

## h. The Node-RED UI - Dashboard

The Dashboard functions as the UI (User Interface) of Node-Red, and can be used as an easier to navigate and operate interface for using your flows... if you know how to make one. It can also work as a quick visual indicator of how your flow or work process is running.

You can have several types of dashboard node, all of which can be found by searching the library for what you are looking for on [Node-Red's](https://nodered.org/) site, however many come with the **node-red-dashboard** package. Some examples are:



The form node can be useful for inputting data from the Dashboard, which, with the correct coding within a function node, can change and edit the parameters of your task.



The gauge node is a great visual indicator for progress or item amounts, such as fluid levels or temperature read-outs. They can be edited to fit your exact needs.

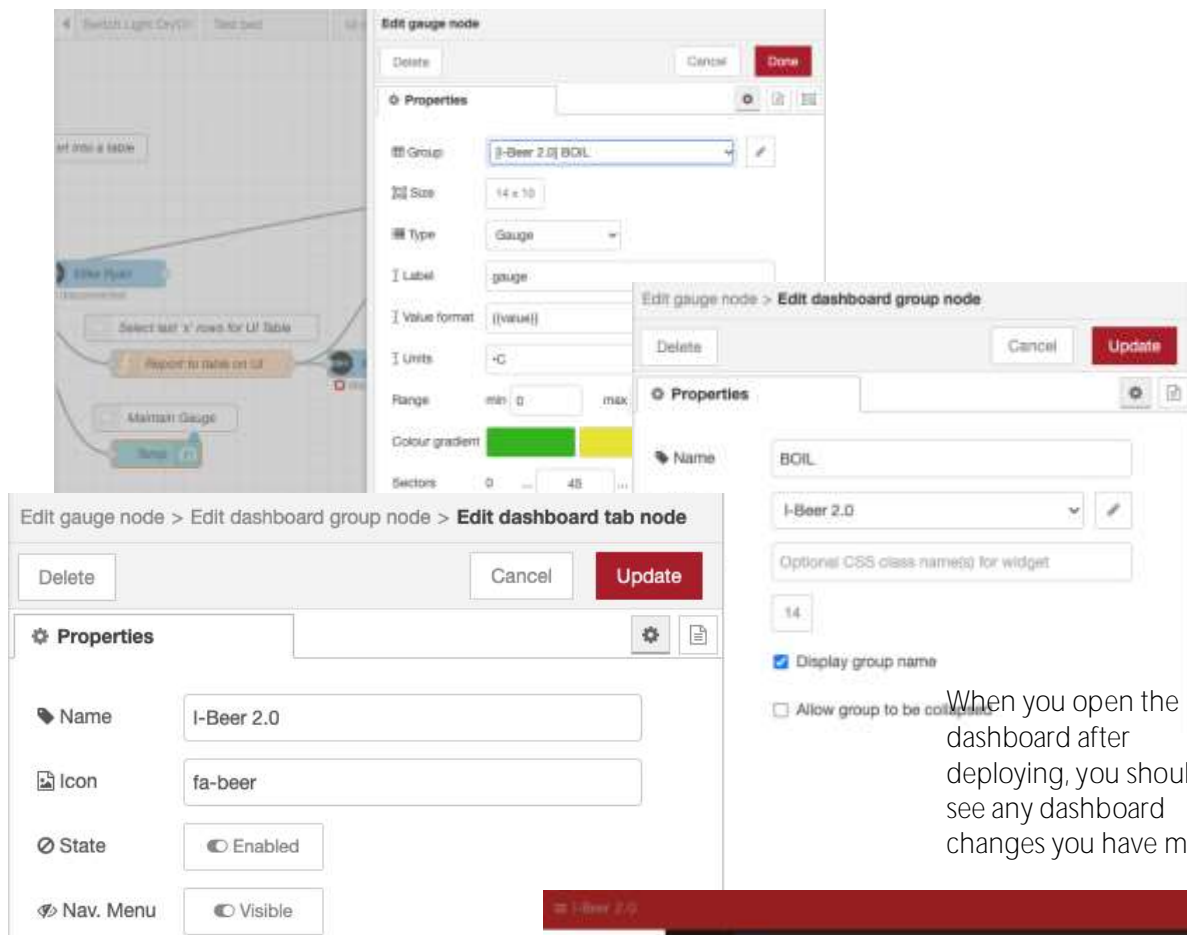


When you add a dashboard node such as a gauge, you can decide where in the dashboard it will appear by deciding what group it will be in.

When you add a dashboard node such as a **gauge**, you can decide where in the dashboard it will appear by deciding what group it will be in.

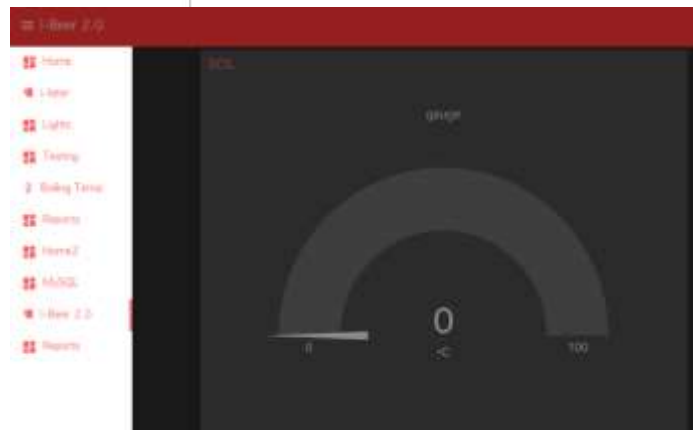
By clicking the pencil next to the group menu, you can create a new group, and by clicking the pencil next to the Tab menu, you can change which Tab in the UI its under.

You can also make changes to the appearance of the node within the dashboard.



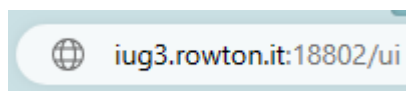
When you open the UI dashboard after deploying, you should see any dashboard changes you have made.

The different UI tabs you have created will appear here, under the menu icon.



You access the Dashboard by using the same URL that you have used to get to the Node-RED alette, but adding '/ui'

<http://iug3.rowton.it:18802/ui>



## 8) The path we will follow

The course is set up in a way that each lesson teaches about a particular element of Node-RED which in turn will lead up to enough education to take on the Project at the end – The Brewery.

Initially we will cover the basic nodes and make things like a light bulb switch on and off by sending a direct code to the next node.

As we progress, we will learn about JSON objects and parameters so that we can send a more complex instruction to the next node.

We will cover variables and maintaining them, see the power of the Function node and then progress on to basing a decision on the data that we will read directly from an IBM DB2 Database. This in turn leads on to adding and changing the Database to keep a record of what we are doing.

We will integrate the Node-RED UI (User Interface) into our flows which will allow us to represent the output data or our actions as, say, a Graph, a Chart or a Gauge. The UI will also work as input. We will set up graphical switches, sliders and selectors which will pass data back that we can use to, say, change the colour that our light strip shows.

As we start the Brewery Project, we will understand the UI that will be displayed, the ways in which to use the slider switches, see clearly how we will receive information into our flows from the graphical Forms and/or the DB2 Database, which we will maintain.

I think it is time to learn by example now. So, let's move on to the lessons.

## 9) i-UG Lesson 1 - Initial Lighting Control

In this lesson, you will see how we can construct a very simple flow that allows a Light in Manchester to be switched on and off. Later we will progress on to colours and hue's, but here we are just setting the scene.

N.B. As the main systems are in Birmingham but the Light Strip is in Manchester, we will be using a link over MQTT to carry the switch information between the two sites. You will set up and use MQTT in this lesson, but the explanation proper of MQTT and how it will help in your future is explained in a later lesson.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples. on occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 10) i-UG Lesson 2 – Extended Lighting Control

In this lesson, you will see how we can get a more detailed flow that allows us to manipulate the Light Strip to change the colours, hue's and the timing of the lights.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 11) i-UG Lesson 3 – Data Controlled Lighting

In this lesson, you will see how we can get a more detailed flow that allows us to manipulate the Light Strip to change the colours, hue's and the timing of the lights.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.



## 12) i-UG Lesson 4 – Engaging with the Node-RED UI -Inputs

In this lesson, you will see how we can get some input from users and some other facilities through the Node-RED U.I.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 13) i-UG Lesson 5 – Engaging with the Node-RED UI – Outputs

Carrying on from the previous lesson – Input - in this lesson, you will see how we present information to users in a graphical way and show data content that the user can request.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 14) i-UG Lesson 4 – Speaking to the world – e-mail

In this lesson we extend our information to the outside world by sending e-mail messages based on the processing in our flow.

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 15) i-UG Lesson 7 – Reading from the DB2 Database

Interacting with the large databases that we will eventually be dealing with is a key part of IT and is crucial to the decision making of any business. Here we will read a simple orders database and present the results to the user in the UI/Dashboard. Note the SQL in this lesson, it will be useful later...

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 16) i-UG Lesson 8 – Writing to and Updating a DB2 Database

Data is important and is always being added to. We also regularly need to change some of the data, like maybe a change of address. Here we allow a user to enter some data and change the contents of the data in the database. We will also just add new information; slightly different. Note the SQL in this lesson, it will be useful later...

[Link to Lesson](#)

Follow the Lesson plan on the link above.

[Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

[Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 17) i-UG Lesson 9 – MQTT – Connecting the world of IoT

In the world of IoT (Internet of Things), you can imagine that there are MILLIONS of devices trying to let people know what they are up to and allowing users to change what they do. This means lots... and lots... and lots of information – Data flying around. MQTT (Message Queueing Telemetry Transport) is a very lightweight, simple method of getting this data from point A to point B. You will find it incredibly useful.

### [Link to Lesson](#)

Follow the Lesson plan on the link above.

### [Link to GitHub and Supporting Documentation](#)

Our GitHub account holds additional information and all the source you should need. It also has additional examples.

### [Link to the answers](#)

On occasion, you may struggle. This is the link to what you SHOULD be seeing. It is also a good place to confirm that what you have done in the lesson is close to what was intended.

## 18) The Brewery – Bringing it all together

[Link the i-BREW Brewery](#)

Follow the route to the iBREW factory here...

GOOD LUCK!

## 19) Appendix A - Getting in to our Node-RED

### a. Starting Node-RED

We will be running Node-RED in a Secure Shell in the IBM i PASE environment. This means that we will initiate it from a SSH which we can drive from a Terminal Application or a PowerShell application. The result will be the same, so you choose the application that you have.



Open your terminal app: Terminal,



PowerShell

First, we need to gain access to the IBM POWER system, so we will enter your User security credentials. Our IBM i POWER system is at **iug3. rowton. it** so at the prompt, we enter: -

```
ssh [STUDENT]@iug3. rowton. it [Use your Student User Name]
```

We will be prompted for a Password, so enter your password. Here we will enter 'Student'

```
password : [STUDENT] [Use your Student Password]
```

If you have been accepted (If you have put in the correct User and Password), you will now be logged in to the IBM POWER System. The Prompt will change to **-bash-5. 1\$**

We can now start Node-RED running. At the -bash-5.14 prompt, type: **node-red**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\mryan> ssh student_02@iug3.rowton.it
student_02@iug3.rowton.it's password:
-bash-5.1$ node-red
22 Sep 10:24:28 - [info]

Welcome to Node-RED
=====
22 Sep 10:24:28 - [info] Node-RED version: v3.1.0
22 Sep 10:24:28 - [info] Node.js version: v18.17.1
22 Sep 10:24:28 - [info] OS: 400 7.4 ppc64 BE
22 Sep 10:24:30 - [info] Loading palette nodes
22 Sep 10:24:32 - [info] Settings file : /home/STUDENT_02/.node-red/settings.js
22 Sep 10:24:32 - [info] Context store : 'default' [module=memory]
22 Sep 10:24:32 - [info] User directory : /home/STUDENT_02/.node-red
22 Sep 10:24:32 - [warn] Projects disabled : editorTheme.projects.enabled=false
22 Sep 10:24:32 - [info] Flows file : /home/STUDENT_02/.node-red/flows.json
22 Sep 10:24:32 - [info] Server now running at http://127.0.0.1:1880/
22 Sep 10:24:32 - [warn]

Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
```

Take note of  
this port  
number...

Your Terminal should now look something like this: -

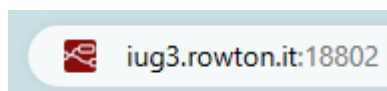


## b. Working with Node-RED through your Browser

Now that Node-RED is running in the server shell, we can access it through a browser.

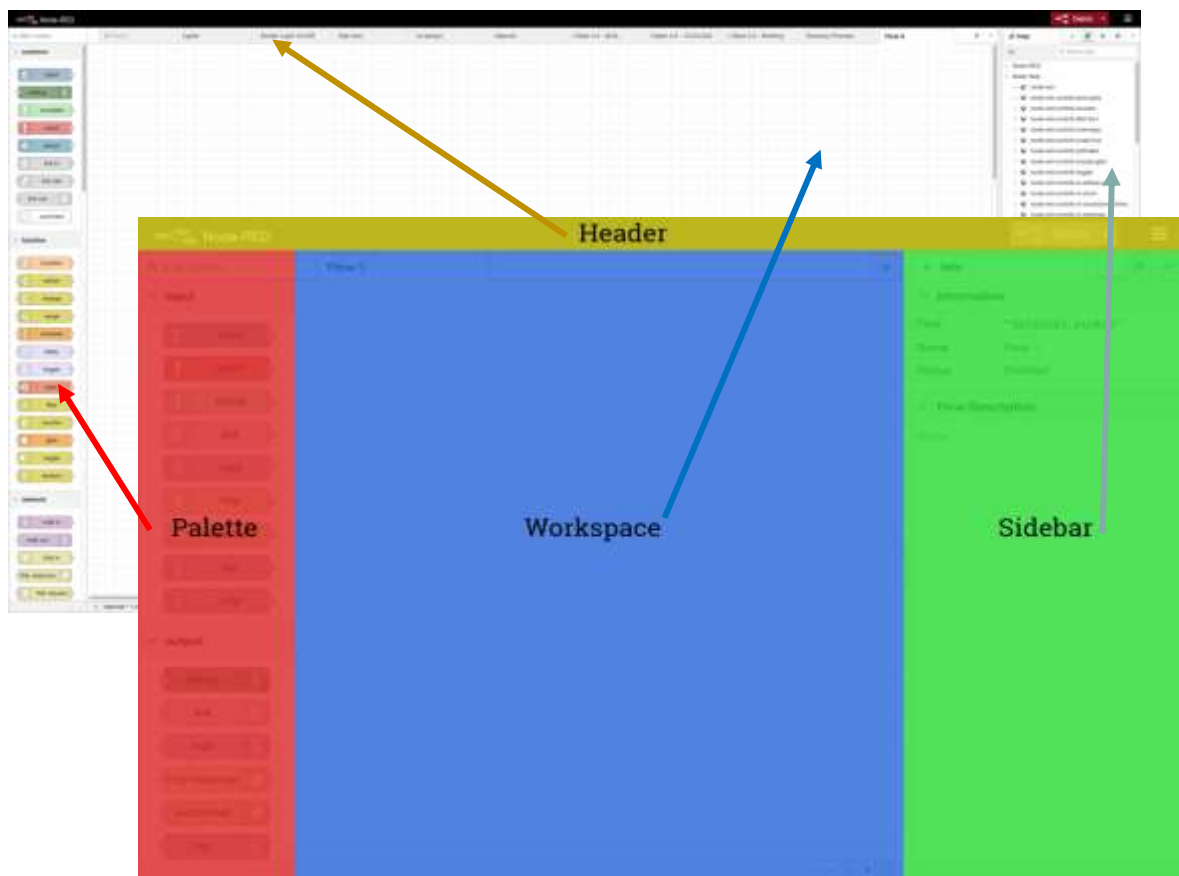
Open your browser. We will key in a URL that looks very like the URL in the picture below, but we will use the Socket that has been assigned to us. The Socket number will be in the highlighted section of the Terminal above. In this case, we would use-

**http://iug3.rowton.it:18802**



You will then be presented with the main Node-RED screen.

It consists of 4 main elements. The Header, the Workspace, the Palette and the Sidebar: -

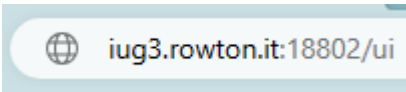


c. Accessing the UI/Dashboard

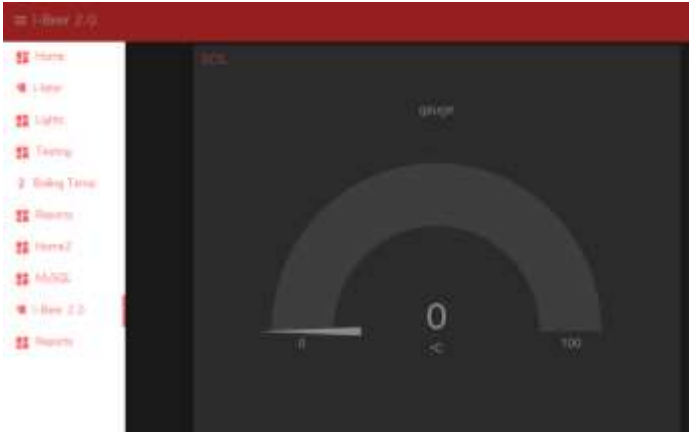
The Dashboard functions as the UI (User Interface) of Node-Red, and can be used as an easier to navigate and operate interface for using your flows... if you know how to make one. It can also work as a quick visual indicator of how your flow or work process is running.

You access the Dashboard by using the same URL that you have used to get to the Node-RED alette, but adding '/ui'

<http://iug3.rowton.it:18802/ui>



The different UI tabs you have created will appear here, under the menu icon.



Forms Example (Input)



Gauge Example (Output)



## 20) Appendix B - i-UG Potential Additional Modules

a. SQL Refresh

b. Talking to Watson AI