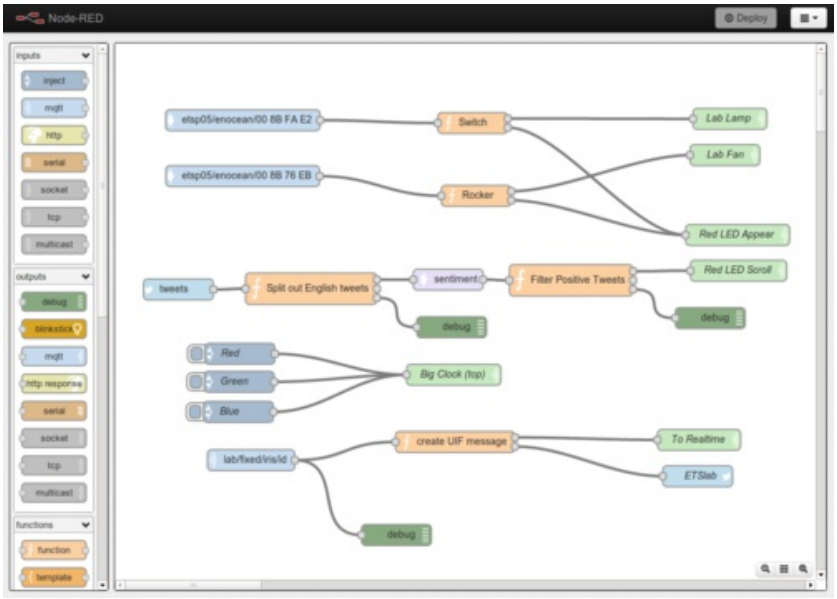
adafruit learning system

Raspberry Pi Hosting Node-Red

Created by Christopher Mobberley



Last updated on 2014-10-10 04:45:09 AM EDT

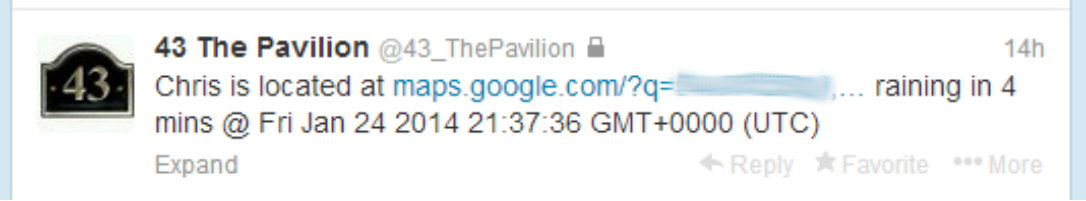
Guide Contents

What is Node-Red?

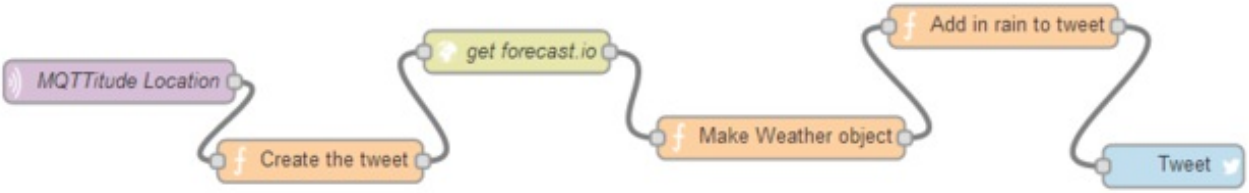
Node-Red in its simplest form is an open source visual editor for wiring the internet of things produced by IBM. What does this mean? Well for someone lazy like me it means I can spend more time making stuff "talk" to each other than worrying about all of the interfacing code I will need to write.

The system contains "Nodes" which look simply to be icons that you drag and drop on to the canvas and wire together. Each Node offers different functionality which can range from a simple debug node to be able to see what's going on in your flow, through to a Raspberry Pi node which allows you to read and write to the GPIO pins of your Pi.

To give an example of what can be done here is my home twitter account telling me where I'm currently located and whether it is going to rain any time soon. I even have a Google maps link thrown in for good measure.



The flow to produce something like this within Node-Red is shown below and it took me literally 15 minutes to have everything set up.



To summarise the flow:

1. An App on my phone called MQTTitude send my location to Node-Red
2. The basic framework of my twitter message is then created in a function node
3. I take my location and query Forecast.io for the weather at that location
4. The returned data is then formatted in to my tweet message using further function nodes
5. The message is then sent to a Tweet node and tweeted for me

Normally this would have taken me a good few hours to complete. But with Node-Red I am able to do it in minutes. I am not worrying about all of the interfacing to MQTT, HTTP or Twitters API. All of that is done for me by Node-Red.

So now I've sold you the idea.. lets get to the installation!

Setting up node.js

Firstly open up a secure shell into your Pi or open up a console session if you are working directly from the Pi.

We then need to get ourselves a working copy of Node.js. Node.js is a an event driven server side javascript environment. It is essentially the foundation that Node-Red will run on.

For the Pi it has been a pain in the past to get Node.js running but luckily there have been some tweaks made by the clever people out there that make our lives easier.

So go ahead and run the following commands in your terminal window, which will ensure our Pi is up to date and ready to go.

```
sudo apt-get update
sudo apt-get upgrade
```

We then jump in to downloading the latest Pi compatible version of Node.js.

```
sudo wget http://node-arm.herokuapp.com/node_latest_armhf.deb
```

Due to the download being in a convenient Debian package we can run the install by simply performing the following command.

```
sudo dpkg -i node_latest_armhf.deb
```

Once the installation has run through you can check to see if node.js is available and installed by calling its version as shown below.

```
node -v
```

We can now move on to installing Node-Red on to our Pi.

Setting up Node-Red

Now that we have node we can then jump in to downloading Node-Red. There is a tutorial on their git for the Pi which can be found here (http://adafru.it/d6B). But as always I want to take a different route as I want to use git to ensure that we always stay up to date with any new commits to the node-red repository.

We will now install Git. This allows us to clone git hub repositories. If you have never heard of git hub it is definitely something you should look into (http://adafru.it/d6C). In essence though it is a good way of adding some version control and open source collaboration to your projects.

To install Git carry out the following command.

```
sudo apt-get install git-core
```

Once Git has installed along with all of its dependencies we want to navigate to our home directory where we will install Node-Red.

```
cd ~
```

We will then clone in our Node-Red repository (repo) to the directory with the following command.

```
git clone https://github.com/node-red/node-red.git
```

You will then find that you have a folder cloned into the directory called “node-red”. We will now install everything using “npm” which is known as the Node Package Manager.

```
cd node-red
sudo npm install
```

After a while everything should run through and install for you. Next we can get Node-Red running, we still have some bits to do but its worth testing first. Make sure you are still in the /home/pi/node-red directory for this one.

```
sudo node red.js
```

You will see a lot of information print out on to the screen, errors about missing packages and all of the good stuff. However the one line you are looking for is related to Node-Red being available on the default 1880 port.

So next you need to navigate to your Pi's IP address and port to see Node-Red in all of its glory.

```
http://Pi-IP-Address:1880
```

You should be welcomed to a blank canvas with all of the "nodes" available on the left hand of the screen. You can drag and drop these to start wiring things together. We will next be discussing how to create your first "flow" using these nodes.

Managing Node-Red

Node-Red is great out of the box but it can be a bit manual to start, stop and run on boot. The following section will describe a simple init script to do all of the hard work for us.

Firstly we create a new init.d file, which is essentially a script for starting, stopping and restarting services under Linux.

```
sudo nano /etc/init.d/node_red
```

Then copy and paste the following code in to the file. To do this within nano copy the text below and right click in the nano window you should see the text start to appear.

The only thing you may need to change before saving is the directory where node-red is installed on your Pi. If you have been following this guide all the way through you do not need to change anything.

For those that have installed Node-Red in a different location just change the following line "cd /home/pi/node-red" to reflect the folder location where Node-Red is installed.

```
#! /bin/sh
# Starts and stops Node-RED
# /etc/init.d/node_red
### BEGIN INIT INFO
# Provides:     node_red
# Required-Start:       $syslog
# Required-Stop:        $syslog
# Default-Start:        2 3 4 5
# Default-Stop:         0 1 6
# Short-Description:    Node-RED initialisation
### END INIT INFO
# Note: this runs as the user called pi

PIDFILE=/var/run/nodered.pid

#Load up node red when called
case "$1" in

start)
        echo "Starting Node-Red.."
        su -l pi -c "cd node-red; screen -dmS red node --max-old-space-size=64 red.js
        echo `screen -ls red | sed -n '2p' | cut -f1 -d.` > $PIDFILE
# or
        #nohup node --max-old-space-size=128 red.js > /var/log/node-red.log &
        #echo $! > $PIDFILE
```

```
;;

stop)
        echo "Stopping Node-Red.."
        su -l pi -c "screen -S red -X quit"
# or
        #kill `cat $PIDFILE`
        rm -f $PIDFILE
;;

restart)
        echo "Restarting Node-Red.."
        $0 stop
        $0 start
;;
*)
        echo "Usage: $0 {start|stop|restart}"
        exit 1
esac
```

Once you have pasted and updated the above script hit CTRL+X and then Y to save changes. Next we will make the file executable so that it can be run.

```
sudo chmod +x /etc/init.d/node_red
```

Finally to ensure the script will start at boot and stop at shutdown we need to update the rc.d file.

```
sudo update-rc.d node_red defaults
```

Thats it! Nice and simple, as stated at the start the following commands will now work.

```
sudo service node_red start
sudo service node_red stop
sudo service node_red restart
```

The final part to managing Node-Red is keeping it up to date. Since we cloned the Node-Red repository originaly from Git Hub we can easily pull down the latest versions whenever we want.

To do this it is as simple as navigating to the home directory of your installation:

```
```

```
cd ~/node-red/
```

And to then run git pull

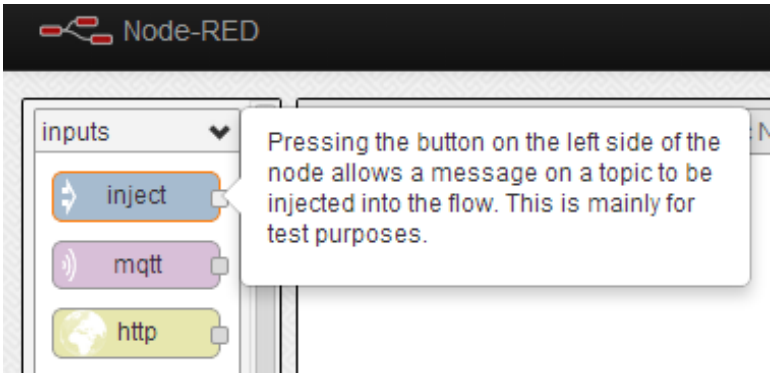
```
sudo git pull
```

If there are any changes you will see them being downloaded. You can then restart Node-Red using our service script and you are ready to go again.

Wiring your first Flow

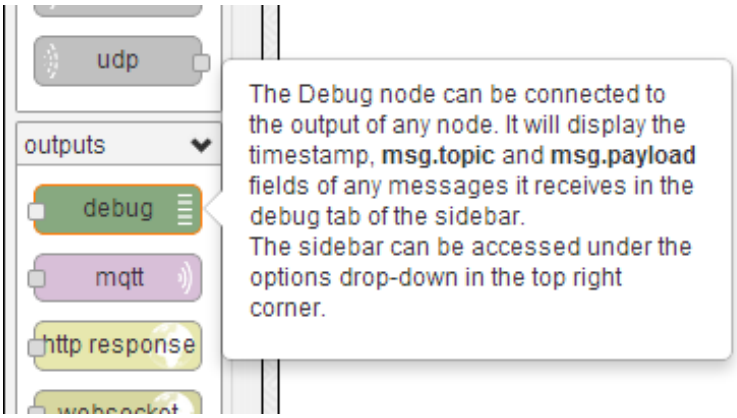
Before delving into the more advanced ways of using Node-Red it's worth understanding the fundamentals of how the systems works. I will now explain a simple "Hello World" setup using an inject node and a debug node.

Firstly open up your instance of Node-Red and locate the inject node. If you hover over any of the nodes a brief description will be given explaining its functionality.

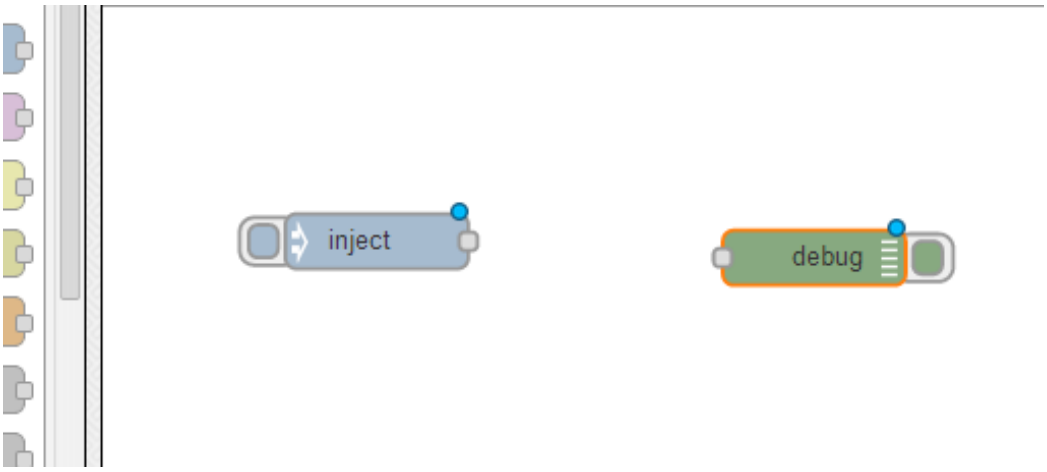


Click and drag the inject node anywhere on to the white canvas in the centre of the screen.

Next we find our debug node which is listed under the "outputs" section. Debug nodes are great for seeing what is happening within your flows and also can be used to read the outputs coming from sensors, GPIO pins and website fetches.



Drag the debug node on to the canvas near to your input node.



Now you can see that the inject node is an input and therefore has a small grey circle on its right hand side. Whilst the debug node is an output and has the small grey circle on its left. We just need to join these two together for them to work as one. Click and drag the inject nodes small grey circle towards the debugs.

As you begin to drag you are creating a "wire" between the two nodes. This is allowing them to pass content and interact with each other. Once completed the wire turns grey to show it is in place.

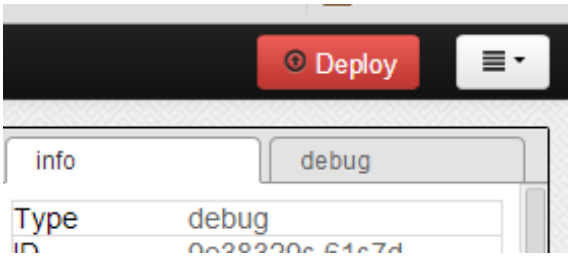
It is worth noting that you can have multiple wires going in to a node. As an example you can have multiple flows all going in to a debug node. We don't need lots of debugs we can just re-use the same one.

As soon as you change the canvas by adding in new nodes or adding/removing wires between nodes you will notice that the "Deploy" button at the top right of the canvas illuminates red. This shows that there are un-deployed changes.

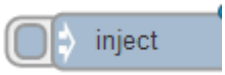
Deploying simply means allowing the system to update so that it is ready to execute the code within your nodes.

Click the Deploy button and you will see an alert stating that the deployment has been successful.

Once the deploy has completed click the debug tab directly below it. This is our debug window that will allow us to see any messages entering into the debug node.



We can now check the wiring is working OK by going to our inject node and clicking on the tab that pokes out on the left hand side.



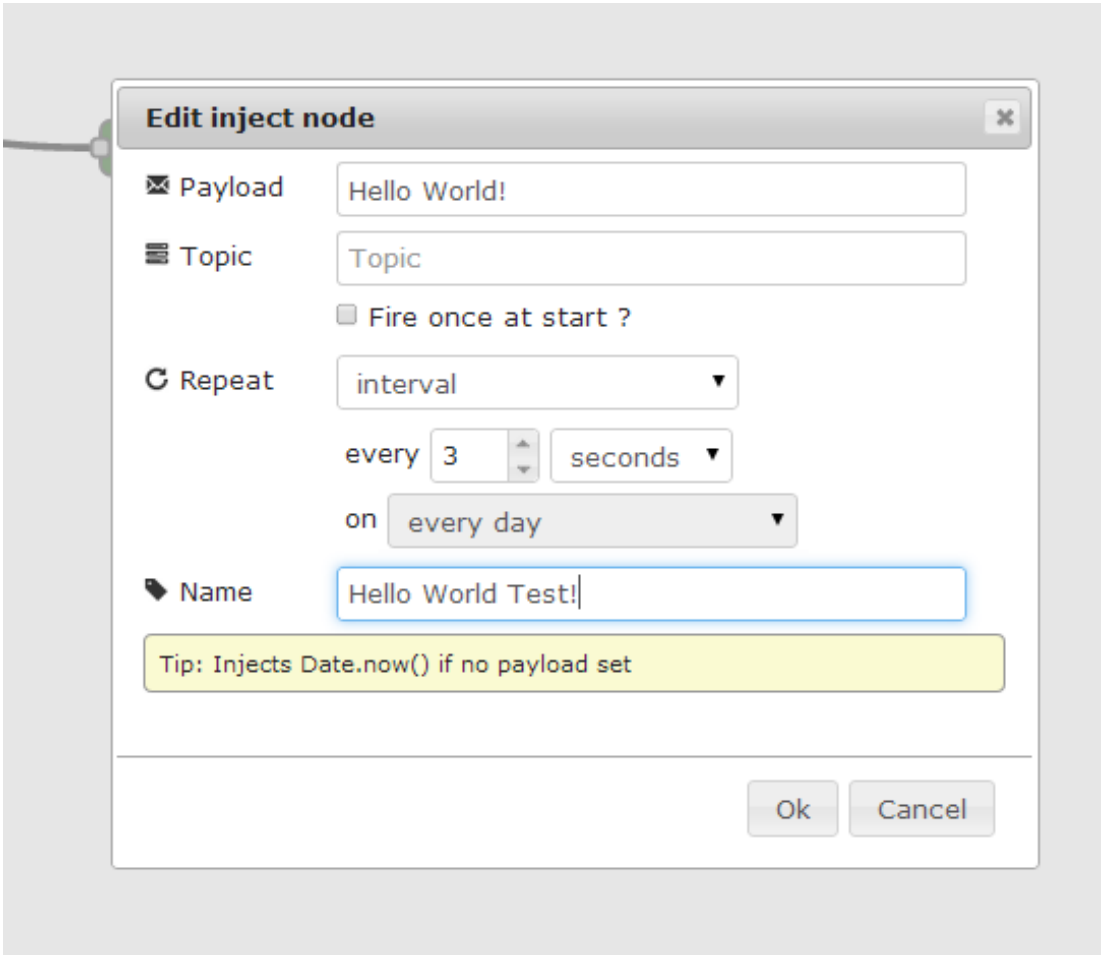
As soon as you click this tab you are telling the node to execute manually due to our intervention. Check the debug window and you will see a bunch of numbers which is actually the current time in milliseconds since 1970.

I promised though a Hello World example so double click on the body of the inject node which is highlighted in orange as shown below.

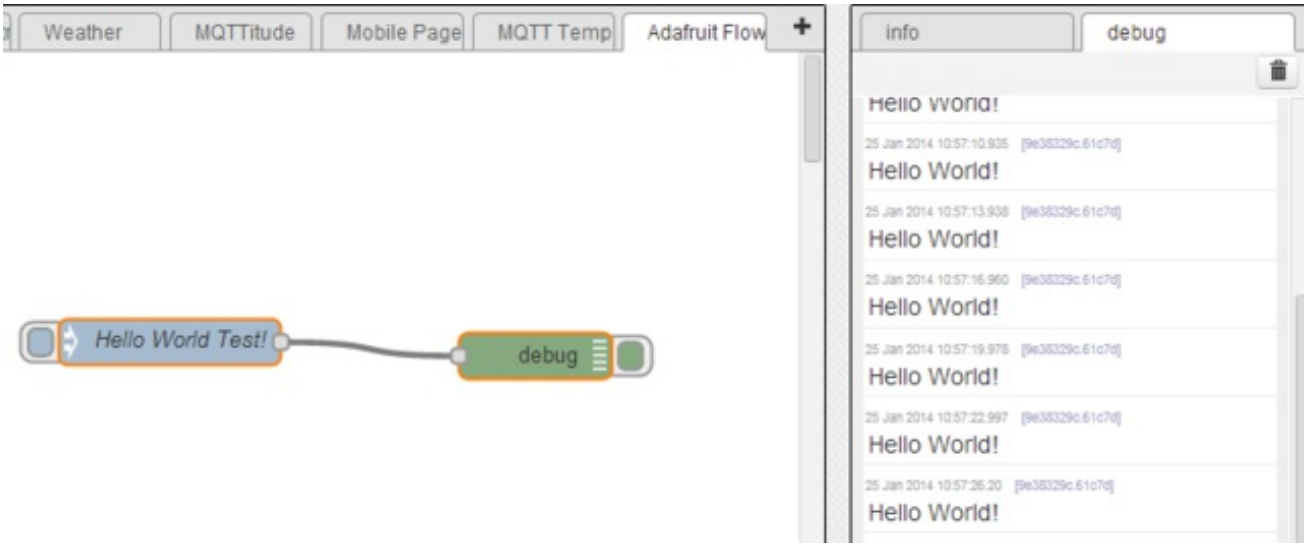
This opens up the nodes options panel. You can do this for any node to see further settings and to make changes.

I want our inject node to output the text Hello World every 3 seconds. In order to do this simply set your options as I have below.

The inject node is very versatile and you can schedule it to inject at various times of day which is really useful. I use it to automate fetching the weather to my twitter account just before I wake to go to work.



Once you have everything setup just click OK and the Deploy button at the top right of the screen. You should then start to see the Hello World text output to the debug window every three seconds.



I have explained quite a few fundamentals during this stage of the tutorial but the exact same process can be used with the other nodes. You simply drag and drop them on to the canvas, wire them together, change their settings and deploy.

The next stage of the tutorial will move in to making more nodes available and ensuring the environment is set up to run them.There are nodes such as the twitter node that require some further dependencies to be setup other than the ones discussed so far.

Installing extra Nodes

Node-Red comes with an excellent set of basic nodes within its palette however there are more out there and there is certainly nothing stopping you from writing your own!

Some of the nodes also have dependencies on external libraries and without these being installed will not function correctly. As an example for the twitter node to work correctly you need OAuth which allows you to authenticate with twitter.com to be able to send/receive tweets.

You may remember from the initial starting of node red we saw a list of these missing dependencies as shown below.

```
25 Jan 11:31:45 - [red] Loading palette nodes
25 Jan 11:31:45 - ---------------------------------------
25 Jan 11:31:54 - [61-email.js] Error: Cannot find module '/home/pi/node-red/../emailkeys.js'
25 Jan 11:31:54 - [imap] : Failed to load Email credentials
25 Jan 11:31:55 - [66-mongodb.js] Error: Cannot find module 'mongodb'
25 Jan 11:31:57 - [74-swearfilter.js] Error: Cannot find module 'badwords'
25 Jan 11:31:57 - [72-wordpos.js] Error: Cannot find module 'wordpos'
25 Jan 11:31:57 - [78-ledborg.js] Warning: PiBorg hardware : LedBorg not found
25 Jan 11:31:57 - [76-blinkstick.js] Error: Cannot find module 'blinkstick'
25 Jan 11:31:57 - [77-blink1.js] Error: Cannot find module 'node-blink1'
25 Jan 11:31:57 - [78-digiRGB.js] Error: Cannot find module 'node-hid'
25 Jan 11:31:57 - [103-hue_discover.js] Error: Cannot find module 'node-hue-api'
25 Jan 11:31:57 - [104-hue_manage.js] Error: Cannot find module 'node-hue-api'
25 Jan 11:31:57 - [103-hue_discover.js] Error: Cannot find module 'node-hue-api'
25 Jan 11:31:58 - [104-hue_manage.js] Error: Cannot find module 'node-hue-api'
25 Jan 11:31:58 - [101-scanBLE.js] Error: Cannot find module 'noble'
25 Jan 11:31:58 - [79-sensorTag.js] Error: Cannot find module 'sensortag'
25 Jan 11:31:58 - [60-wemo.js] Error: Cannot find module 'wemo'
25 Jan 11:31:58 - [26-rawserial.js] Advise: Only really needed for Windows boxes without serialport np
25 Jan 11:31:58 - [39-wol.js] Error: Cannot find module 'wake_on_lan'
25 Jan 11:31:58 - [69-mpd.js] Error: Cannot find module 'komponist'
25 Jan 11:31:58 - [57-notify.js] Error: Cannot find module 'growl'
25 Jan 11:31:58 - [57-prowl.js] Error: Cannot find module 'node-prowl'
25 Jan 11:31:58 - [57-pushbullet.js] Error: Cannot find module 'pushbullet'
25 Jan 11:32:00 - [56-twilio.js] Error: Failed to load Twilio credentials
25 Jan 11:32:00 - [92-xmpp.js] Error: Cannot find module 'simple-xmpp'
```

You may have more in your list as I have installed quite a few but the main point is that the "Cannot find module 'X'" is Node-Red telling us that we have a missing dependency.

Luckily it is very easy to install these and as an example if we wanted to install the swear

filter node, which filters out bad words we just simply have to carry out the following steps.

First stop node red if it is running using the management script from earlier on in the tutorial.

```
sudo service node_red stop
```

Then ensure you are within your Node-Red directory. If you have been following this tutorial yours will match mine as shown below.

```
cd ~/node-red/
```

Next we use node package manager to install the dependency for us, which is named exactly as the "Cannot find module" statement told us.

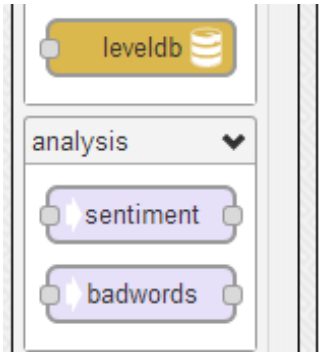
```
sudo npm install badwords
```

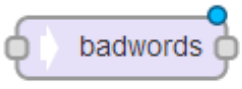
You will then see the package being installed.

```
npm http GET https://registry.npmjs.org/badwords
npm http 200 https://registry.npmjs.org/badwords
npm http GET https://registry.npmjs.org/badwords/-/badwords-0.0.3.tgz
npm http 200 https://registry.npmjs.org/badwords/-/badwords-0.0.3.tgz
badwords@0.0.3 node_modules/badwords
```

Once that has completed we can start Node-Red up again and you will be able to now use the Swear Filter node.

```
sudo service node_red start
```





To ease the pain of you having to go through the whole process each time for each module I have constructed a list of the common modules that you will need to install.

Simply stop Node-Red and follow the process as we did for the bad words example but replace sudo npm install badwords with the following line.

```
sudo npm install ntwitter oauth sentiment wordpos xml2js firmata fs.notify serialport feedparser push
```

Further information

You should by now be fully setup to use Node-Red and to keep up to date with the goings on for the projects why not take a look at the following links.

- Official Node-Red Website - http://nodered.org/ (http://adafru.it/d6E) - Always keep yourself up to date with what is going on and their documentation also explains different aspects of Node-Red
- Node-Red GitHub Repo - https://github.com/node-red (http://adafru.it/d6F) - Node-Red is open source and available on GitHub so why not take a look at how it all works.
- Node-Red Official Twitter Account - https://twitter.com/NodeRED (http://adafru.it/d6G) - Does what it says on the tin.
- My Twitter Account For Specific Questions - https://twitter.com/Hardware_Hacks (http://adafru.it/d7K)