



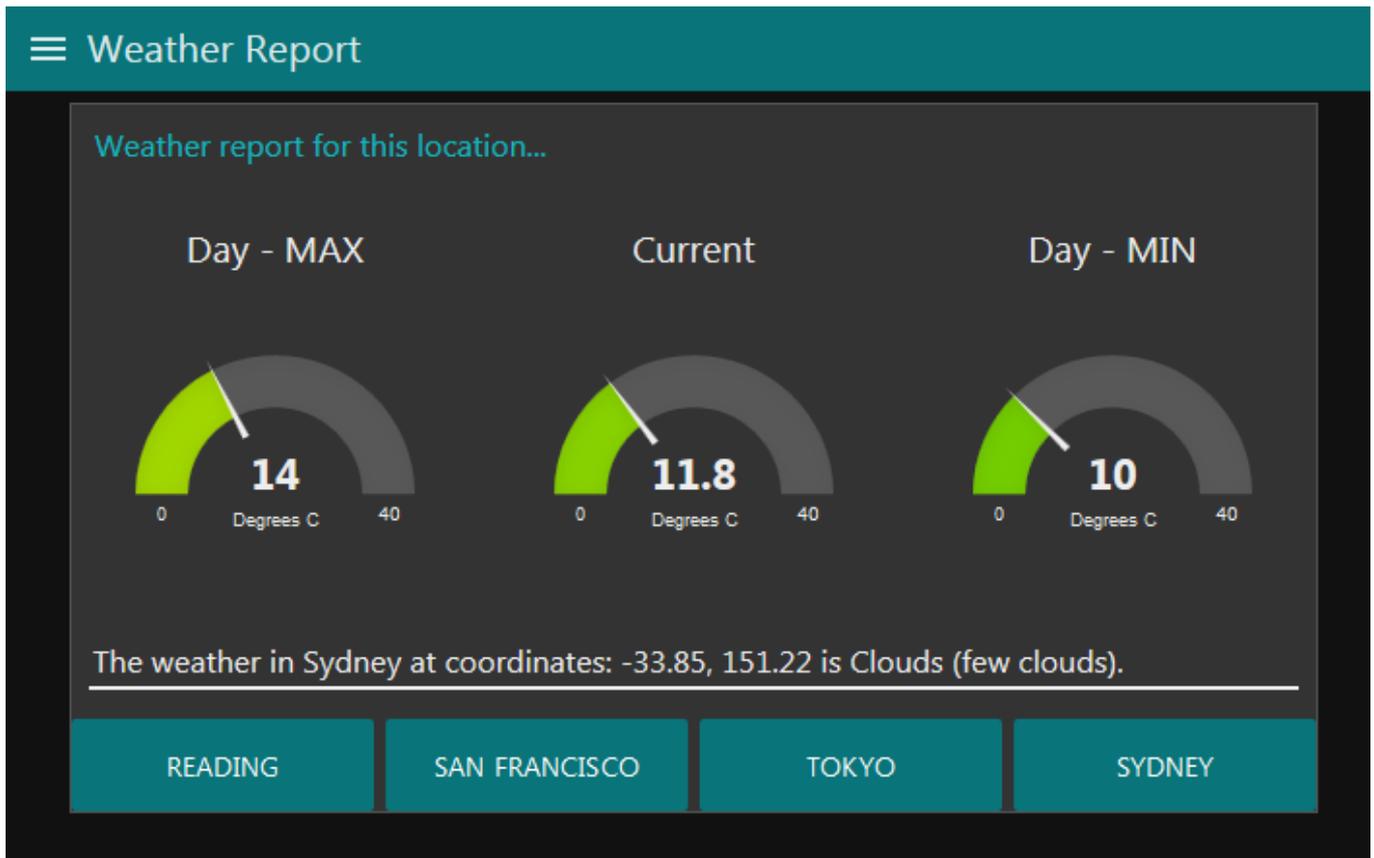
**Bohunt School (Wokingham)**  
**Internet of Things and Node-RED**



This session should be a bit of fun for you - creating a dashboard.

“*Session M1: Weather Reports*” showed you how to use OpenWeatherMap in a Node-RED flow to indicate weather report results in a set of debug nodes.

This session takes the idea further forward by showing you how to create a visually attractive, dynamic dashboard - like the image shown below.



The values displayed on the dashboard use a set of gauges to show the Day-Max temperature, Current temperature and Day-Min temperature for four different locations around the world (chosen by clicking the appropriate button).

The first thing to do is to plan-out the layout for the dashboard.

Shown on the next page is a rough sketch I made to plan the dashboard layout.

Running across the top of the dashboard is the name field (size: 12 by 1 units).

On the first row are three gauges, each 4 by 4 units in size.

The second row consists of a “text input” field where the description of the weather will appear. (Size: 12 by 1 units).

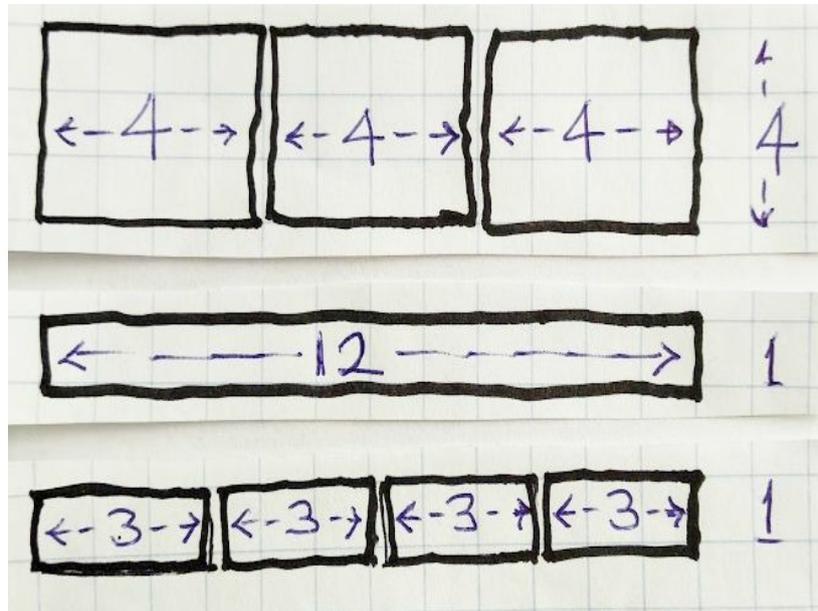
The last row consists of four buttons (size: 4 by 1 units) to select the location.



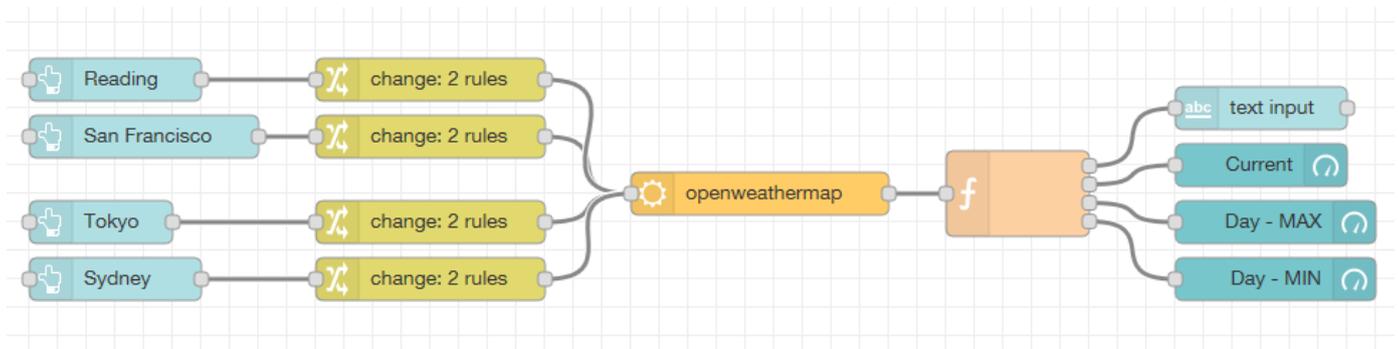
# Bohunt School (Wokingham) Internet of Things and Node-RED



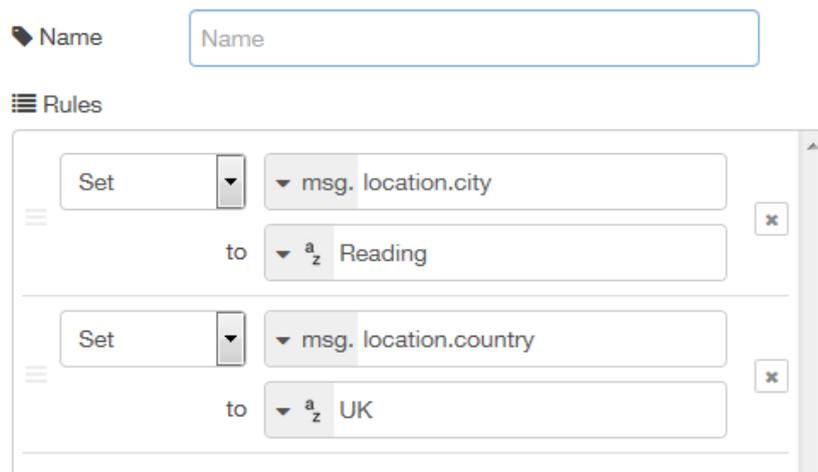
Here's my rough sketch.



The next thing to do is sort out the Node-RED flow (shown below) to interact with the dashboard.



On the left of the flow are four UI-buttons that are connected to a change node. Each change node sets a property for city and country as shown below.





## Bohunt School (Wokingham) Internet of Things and Node-RED



This information is sent to the “openweathermap” node which will return a JavaScript object for the city and country that was selected.

The “function” node that joins the flow to the “openweathermap” node is a little bit more complicated than function nodes you have used/created so far.

Name

Function

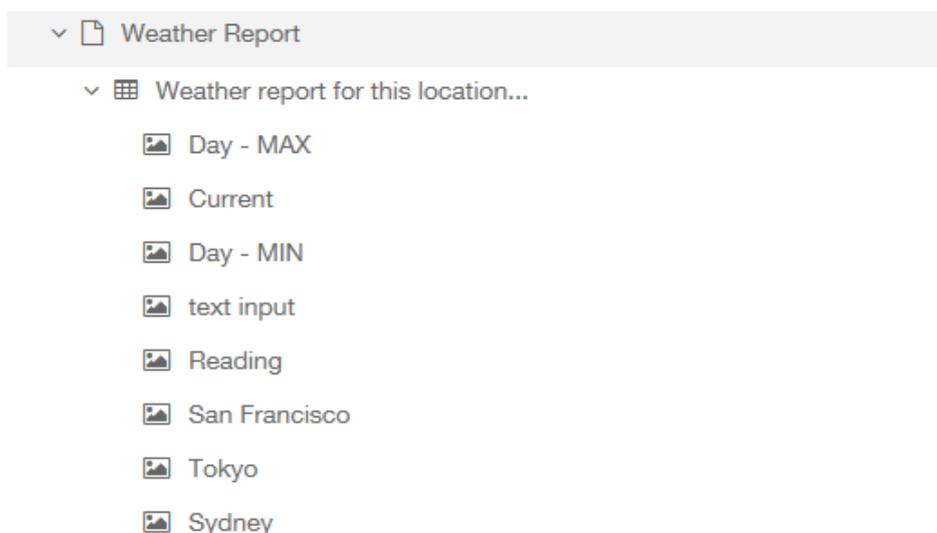
```
1 var tempDescription = msg.payload.description;
2 var tempC          = msg.payload.tempc;
3 var tempMaxC       = msg.payload.temp_maxc;
4 var tempMinC       = msg.payload.temp_minc;
5
6 return [{payload:tempDescription},{payload:tempC},{payload:tempMaxC},{payload:tempMinC}];
```

Each part of the JavaScript Object is filtered out and output to a dedicated pin on the function icon before being sent to the appropriate dashboard widget.

The screen shot below shows the stacking order for the various dashboard widgets used to create the desired layout.

If you have time (in this session) you could try:

- altering the stacking order and see the effect on the dashboard layout
- expanding the flow to include other pieces of data or other locations



If you get stuck or need some help, here’s a link to the Node-RED flow.

***Congratulations you now know how to create a simple dashboard.***