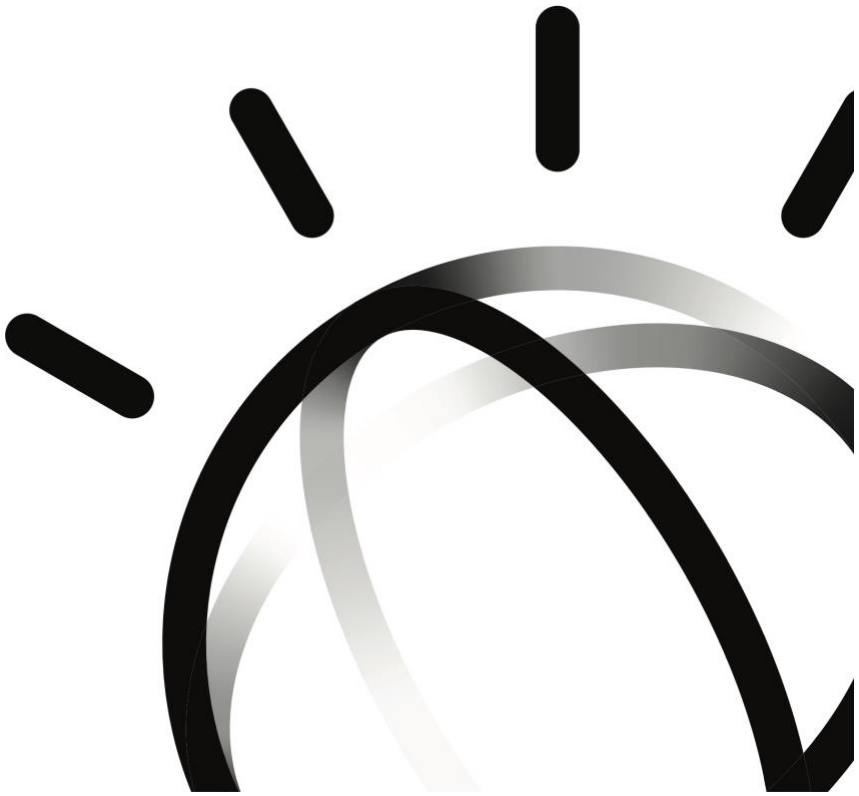
IBM Watson Solutions

Business and Academic Partners



Build a Node-RED App

That reveals Real-time Earthquakes

Prepared by: Armen Pischdotchian

Version 1.0 February 2018

Overview

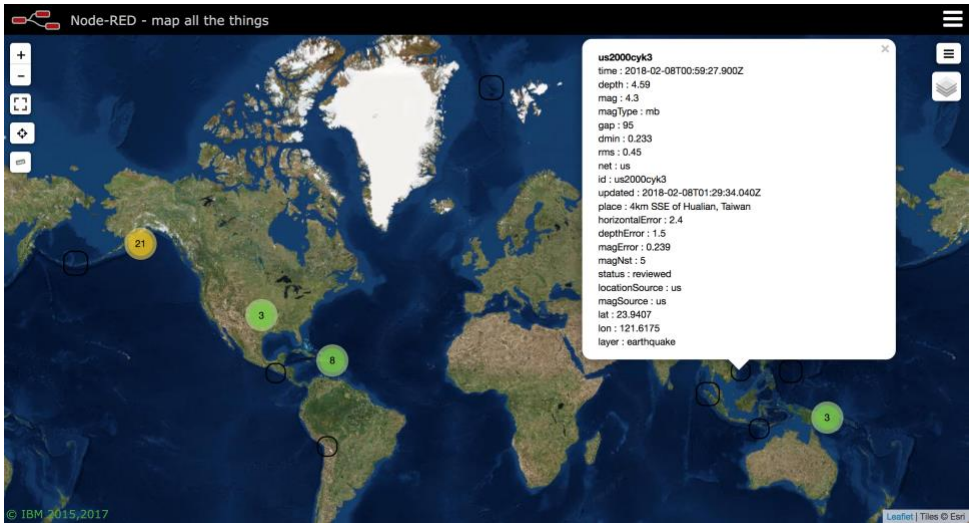
Node-RED is a visual tool for wiring the Internet of Things. It is easy to connect devices, data and APIs (services). It can also be used for other types of applications to quickly assemble flows of services. Node-RED is available as open source and has been implemented by the IBM Emerging Technology organization. Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click. While Node-Red is based on Node.js, JavaScript functions can be created within the editor using a rich text editor. A built-in library allows you to save useful functions, templates or flows for re-use.

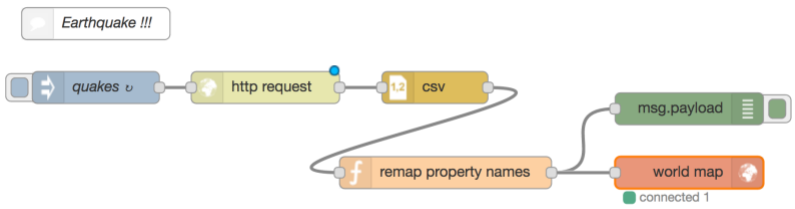
Node-RED is included in the Node-RED starter application in IBM Cloud but you can also deploy it as a stand-alone Node.js application. Node-RED is not just used for IoT applications, but it is a generic event-processing engine. For example, you can use it to listen to events from http, web sockets, TCP, Twitter and more and store this data in databases without having to program much if at all. You can also use it for example to implement simple REST APIs.

About your application

In this exercise, we will show how to generate an ESRI global map depicting active Earthquake location, in real time from csv data gathered from USGS government resources. The flow will present a separate tab in your browser by pressing `ctrl-shift-m`.

Use the flow below as a guideline as you drag and drop nodes onto the canvas. Connect the nodes at the end and always refer to the flow depicted below for logical placement of the nodes onto the canvas. You will be building the app from scratch using Node-RED as depicted in the image below:

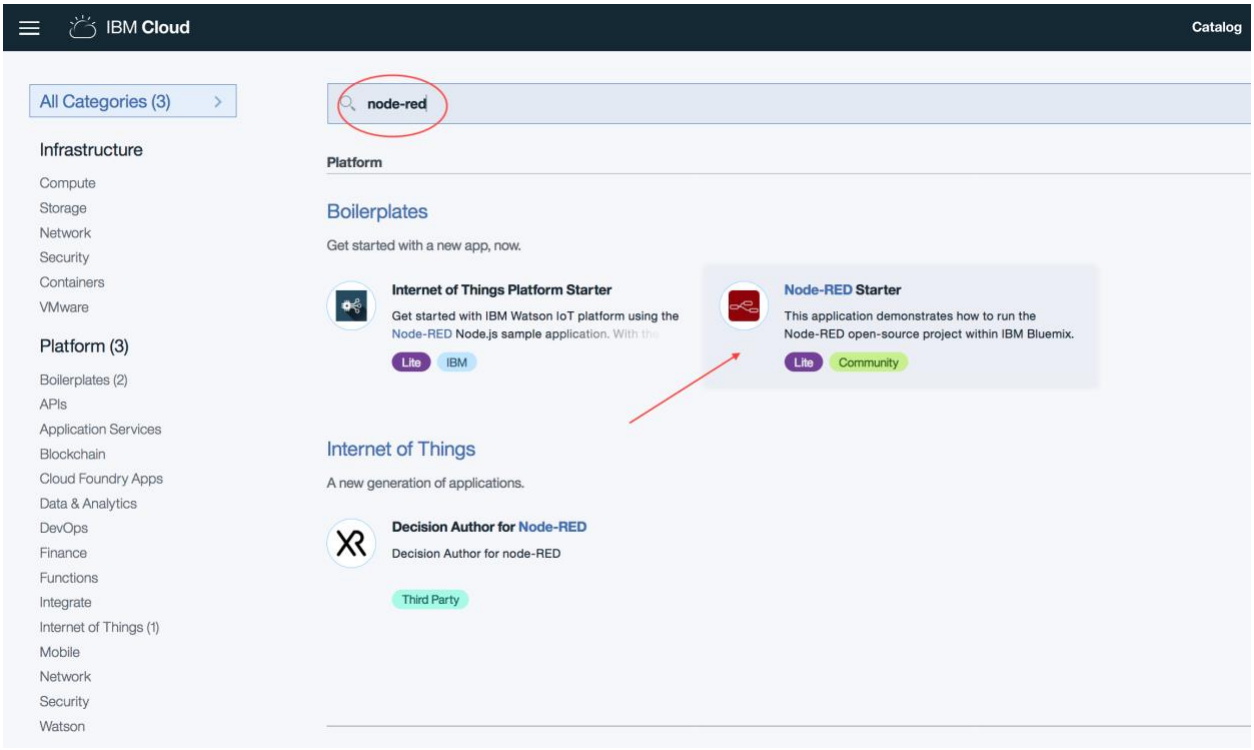




Build a Node-RED Starter Boilerplate

Let's begin by creating a Node-RED boilerplate app in IBM Cloud. This lab assumes that you have a IBM Cloud account and that you can sign in. You can register for IBM Cloud by clicking the **SIGN UP** button in the upper right corner of the page at console.ng.bluemix.net. After registering, you will receive an email message that requires you to confirm your registration.

1. Sign into IBM Cloud: https://console.bluemix.net/catalog/
2. From the IBM Cloud console, access the **Catalog** tab and search for **Node-RED Starter**.



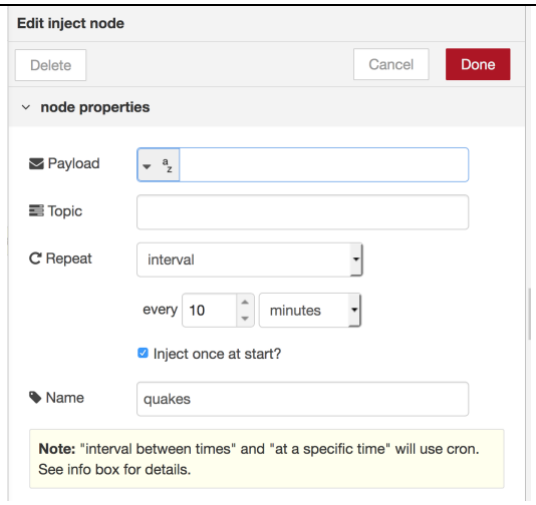
3. Specify a unique name for your app and click **Create**. Allow enough time for the app to stage and start. This may take a few minutes.
4. Once the app shows the status of *Running*, click **Visit App URL**.
5. In the ensuing page, advance through the wizard (give it a username/password) no need to check any other boxes underneath.
6. Click **Next** without selecting any of the provided starters; and click **Finish**.
7. In the ensuing page, click **Go to your Node-Red flow editor**.
8. Login with the username/password you created earlier in Step 6.

Populate the Node-RED canvas

The remaining steps that pertain to building your node-RED canvas are outlined in the table below.

Steps	Example screen capture

9. Drag and drop a **Comment** node. Keep an eye on the completed flow in the previous page as you add nodes. It can help you identify the nodes easily and roughly where to place them on the canvas.
10. For Title specify: Earthquake
11. In the Body copy/paste this:
 	
`Press the button on the `inject` node to get the list of earthquakes from the USGS public API.`
 	
`This returns a CSV - which we then parse using the `CSV` node into individual objects.`
 	
`These are then passed through a `function` to remap the property names into ones suitable for the map node, and passed to the `web-socket` out node.`
 	
`The map will be served at http://localhost:1880/worldmap`
 	
`You will need to zoom out :-)`
	

12. Drag and drop an **inject** node. 13. For the Repeat field, select **interval**. 14. Set the interval to every **10 minutes**. 15. Check the Inject once at Start? check box. 16. Click **Done**.	

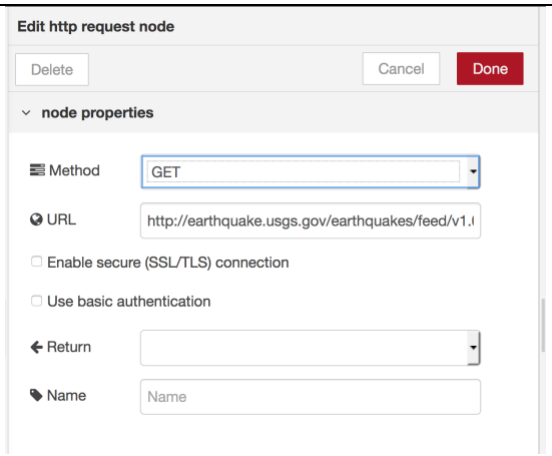
17. Drag and drop a **http request** node.
18. For Method, select **GET** (default) method.
19. For URL copy/paste this path (one line)

```
http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/2.5_day.csv
```

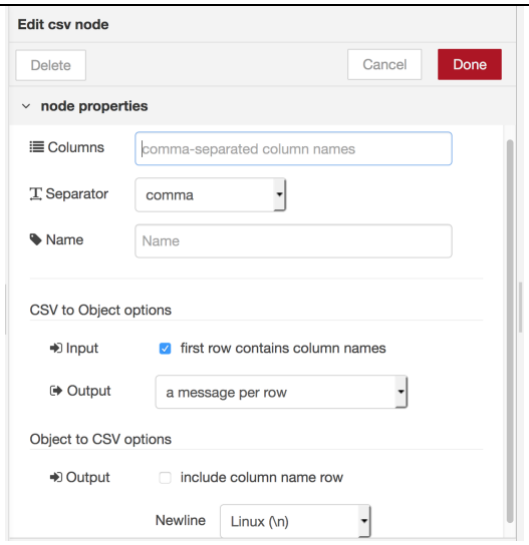
20. Click **Done**.

You can obtain other data sets from this website:

https://earthquake.usgs.gov/earthquakes/feed/v1.0/csv.php



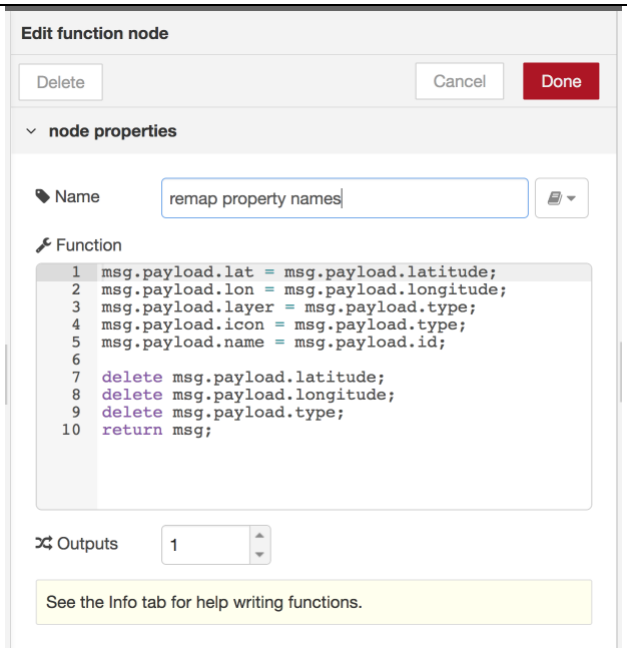
21. Drag and drop the **csv** node.
22. For the Input field check the **first row contains names.**
23. Leave all else per default and click **Done**.



24. Drag and Drop a **function** node to extract the query parameter from the web request and assign it to the payload to be provided as input to the Visual Recognition node.
25. Specify a name, for example: **remap property names**.
26. Copy and paste the following *replacing* everything that you see in the box:

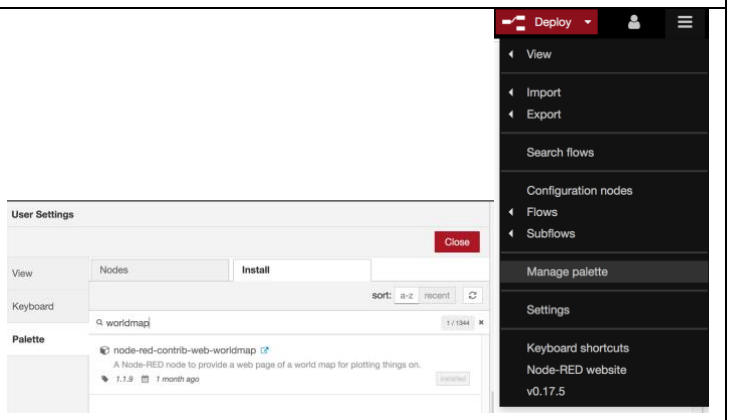
```
msg.payload.lat = msg.payload.latitude;
msg.payload.lon = msg.payload.longitude;
msg.payload.layer = msg.payload.type;
msg.payload.icon = msg.payload.type;
msg.payload.name = msg.payload.id;

delete msg.payload.latitude;
delete msg.payload.longitude;
delete msg.payload.type;
return msg;
```

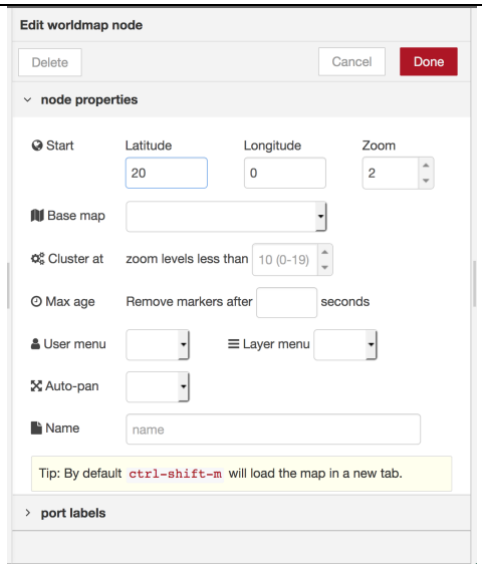


Before you proceed any further, you need to install the worldmap node. Not something that appears by default. This is a good opportunity to explore the numerous open sourced nodes that the community has developed for Node-RED. Continue with the steps below to install the worldmap node.

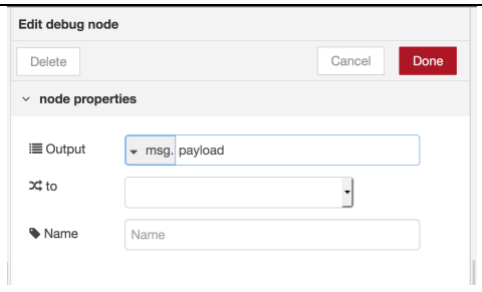
27. From the hamburger icon to the top left select **Manage palate**.
28. Click the **Install** tab.
29. Search for **worldmap**.
30. Install the node.
31. Click **Close**.



32. Drag and drop a **worldmap** node (not the worldmap in).
33. For Latitude select **20**.
34. For Longitude select **0**. These values are just starting points or place holders.
35. For Zoom, select **2**.
36. Leave all else empty (default) and click **Done**.



37. Drag and drop a **debug** node. No need to make any changes here. Incidentally, you can add debug nodes to any of your nodes and activate and deactivate them as you try to debug your more complex flows. This is akin to inserting a stepper in your code to see just which function calls what method and so forth.
38. Click **Done**.



39. Click **Deploy**.
40. Press `ctrl-shift-m`. This is the default key stroke to invoke another tab that has the map of the world.
41. Click the button to the left of the inject node to activate the inject node. Have fun with the map in the adjacent tab in your browser; try the ESRI view!