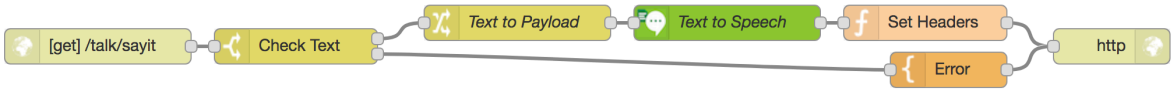
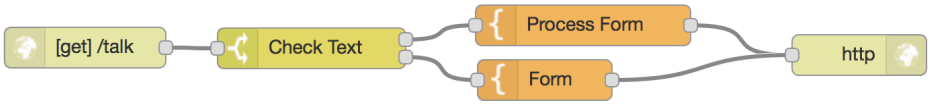
Text to Speech in Node-RED

Hands-On Lab

JeanCarl Bisson | jbisson@us.ibm.com | @dothewww



Add a web endpoint to convert text to audio
(see *Add Text to Speech in Node-RED*)



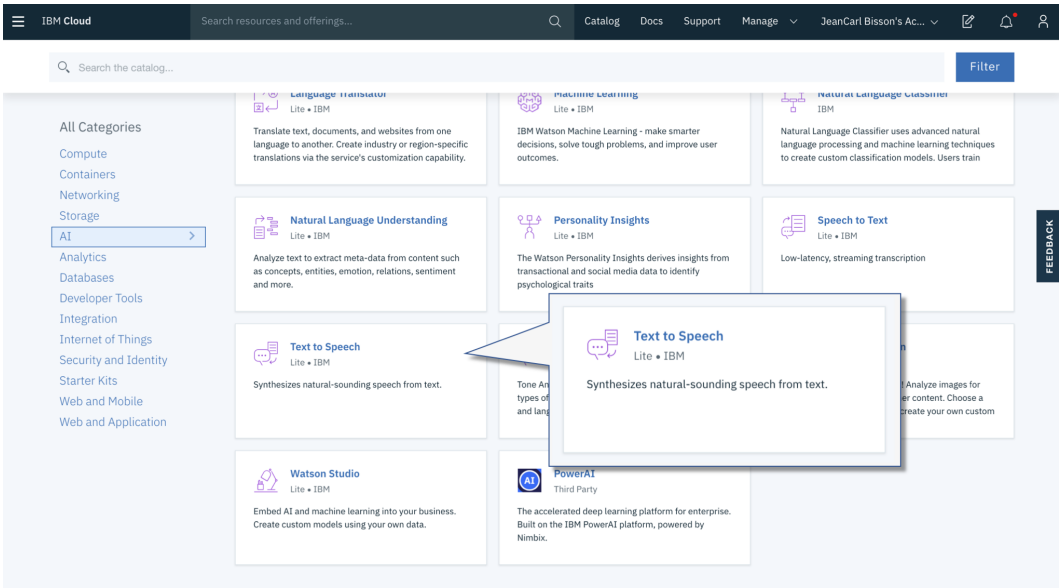
Create a webpage to input text and play audio
(see *Creating an Interactive Web UI*)

A digital copy of this lab and code snippets can be found at:
`http://ibm.biz/node-red-text-to-speech`

Add Text to Speech Service in IBM Cloud

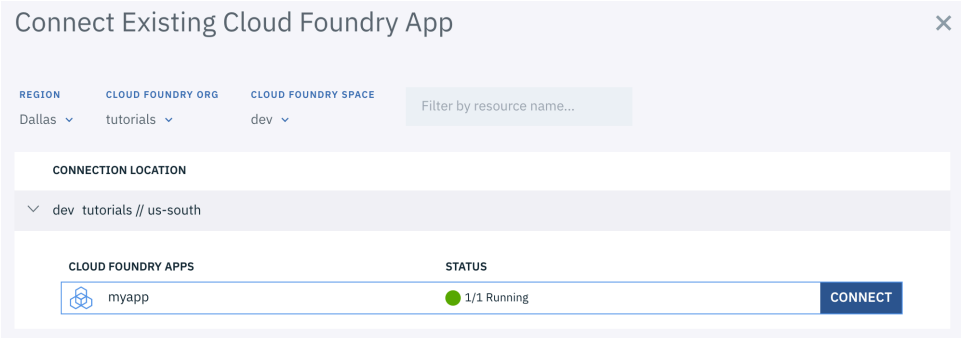
The Text to Speech node in Node-RED requires API credentials. In this section, we will create and bind the IBM Watson Text to Speech service to the Node-RED application.

1. Click on the **Catalog** link at the top of the IBM Cloud Dashboard. Under the AI section, click on the **Text to Speech** tile.

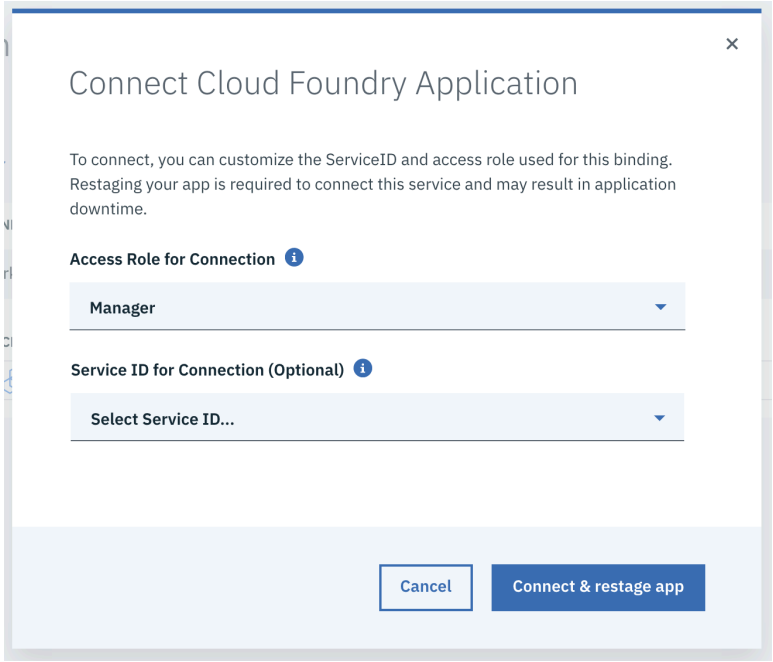


2. You can optionally give the service a custom name or leave it as the one given. Click **Create**.
3. Click on **Connections** in the menu on the left.
4. Click **Create connection** on the right.

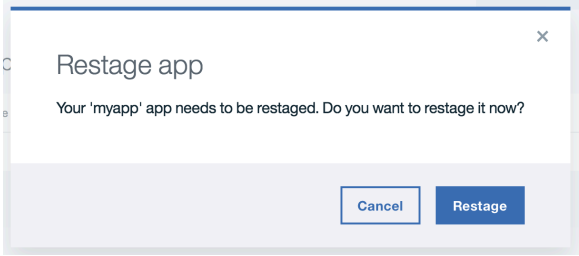
5. Click **Connect** next to the Node-RED application you created earlier.



6. IBM Cloud will prompt to configure access role for the new connection and service ID. Click **Connect & restage app**.



7. IBM Cloud will prompt to restage the application. Click on **Restage**. The application will restart and include the new service credentials in the environment.



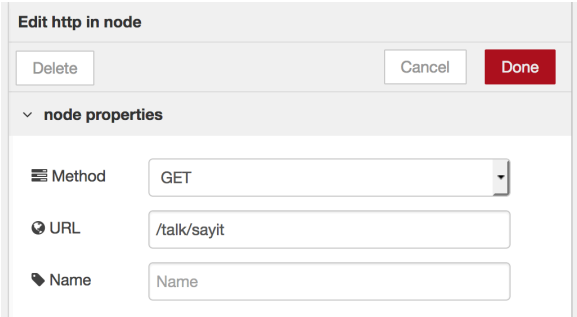
8. When the application has finished restaging, open the Node-RED Flow Editor. If you already have Node-RED open, refresh the page.

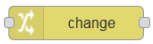
Add Text to Speech in Node-RED

In this section, we will use the IBM Watson Text to Speech service to produce a .wav audio file from input text through a simple web endpoint generated using a Node-RED flow.

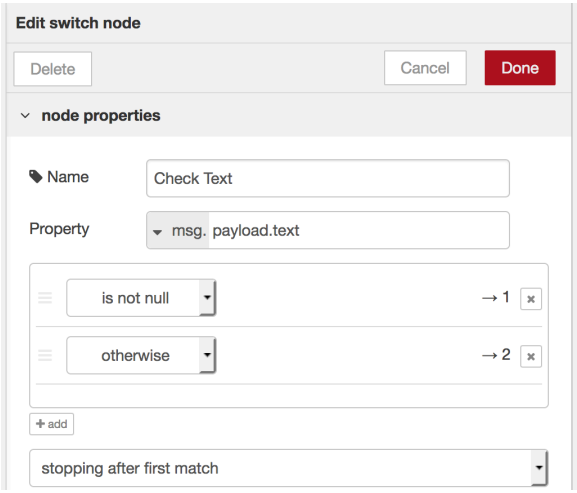
1. Add a 

node as shown below to collect the incoming speech request.



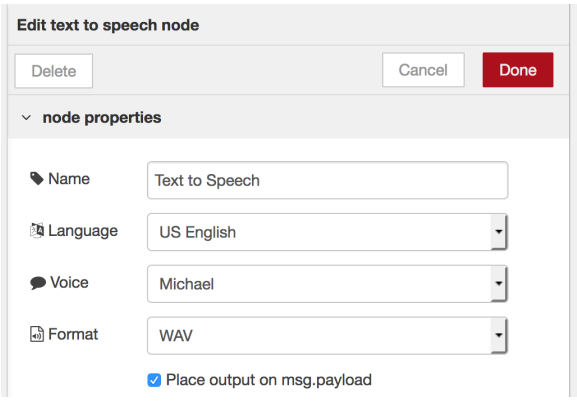
2. Add a 

node as shown below to extract the query parameter `msg.payload.text` and set it as the msg.payload. When invoked with query parameters such as `?text=Hello`, the text will be placed into the `msg.payload` object.



3. Now add a 

node as shown below. This node will generate the binary wav stream content and put it in the `msg.payload` property.

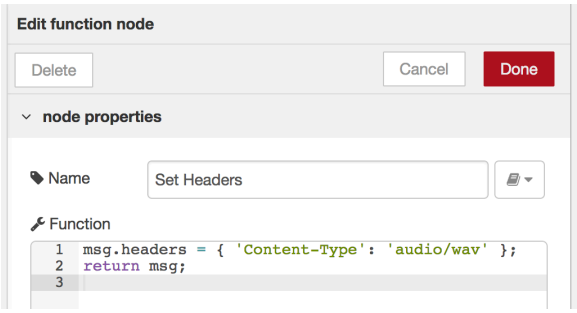


You can change the language, the voice, or the format of the audio that is returned.

4. Add a 

node as shown below to add the appropriate audio HTTP headers to the response.

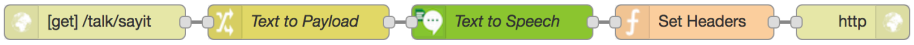
```
msg.headers = { 'Content-Type': 'audio/wav' };
return msg;
```



We set the HTTP response headers by setting the **msg.headers** to the Content Type of audio/wav. This is required in order to let browsers know that this is an audio file and not HTML.

5. Finally, add a 

node. This node will simply return what's in **msg.payload** to the HTTP response. The completed flow should look like the flow shown below:

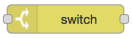


6. Click on 

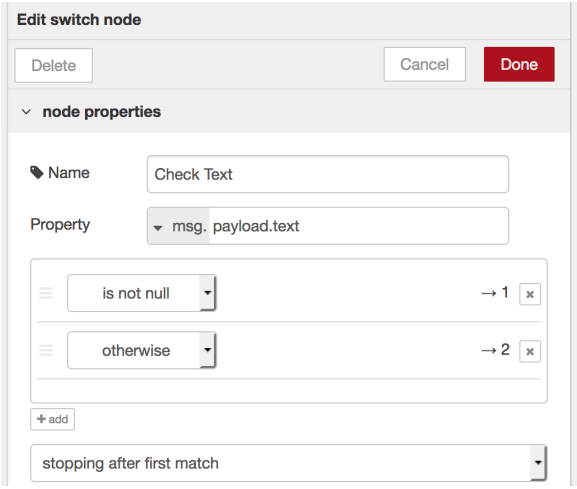
to save the changes. Test the flow by opening the application URL, appended with /talk/sayit?text=Hello as shown below.

```
https://<<MY-APP>>.mybluemix.net/talk/sayit?text=Hello
```

This should prompt you to save a file. Depending on how your browser is configured, it may download the audio file or play it within the browser. Either way, play it and you should hear the text.

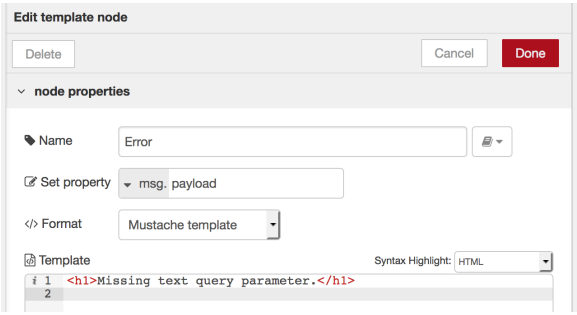
7. This flow has a caveat, however. The flow will fail when the text query parameter is not set. Add a 

node as shown below to check if the text query parameter is present.

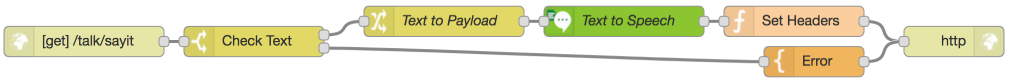


8. You'll notice that adding the second otherwise rule has created a second output handle for the switch node. Add a 

node with the error message shown below.



9. Connect the nodes together as shown below.



10. Click on 

to save the changes. Test the flow by opening a browser and going to the application URL, appended with the following endpoints:

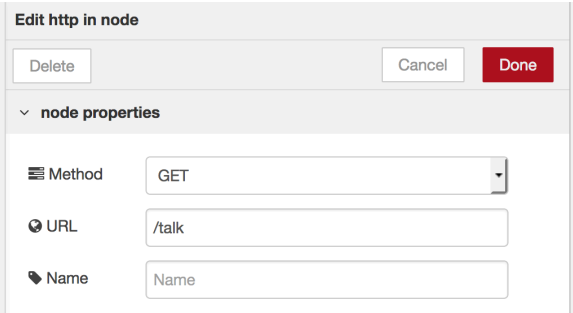
`https://<<MY-APP>>.mybluemix.net/talk/sayit?text=Hello`	(plays the audio with Hello)
`https://<<MY-APP>>.mybluemix.net/talk/sayit`	(displays error message)

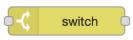
Creating an Interactive Web UI

In this section, we will create a simple webpage that displays an input textbox for the user to enter text. When the form is submitted, the text will be played via audio.

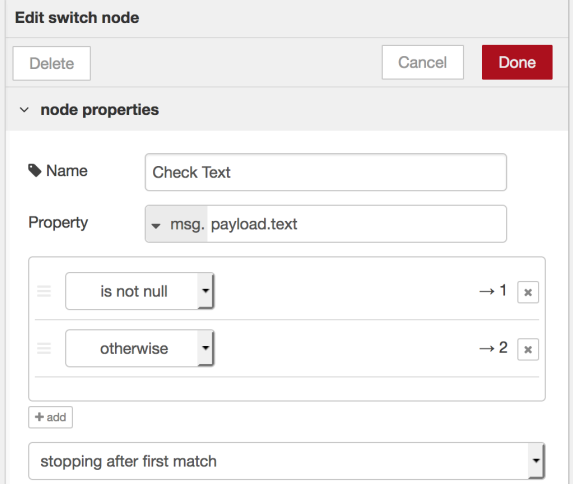
1. Add a 

node as shown below.



2. Add a 

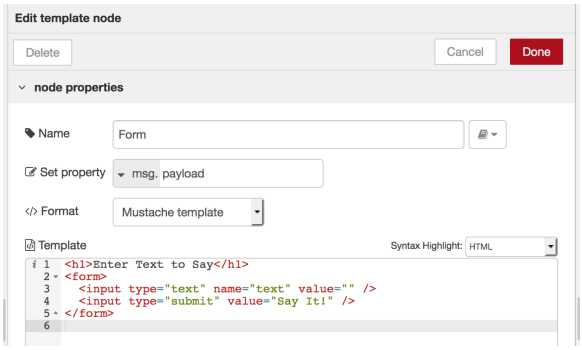
node to branch the flow depending whether text input is submitted.



When the form is submitted and the text parameter is present, proceed with the first flow. Otherwise, the blank form (flow #2) will be used.

3. Add a 

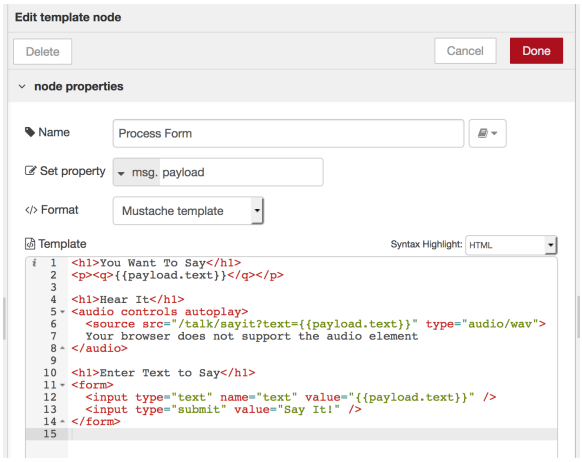
node with the HTML shown below.



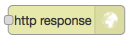
Get the code:
`ibm.biz/Bd4vtc`

4. Add a 

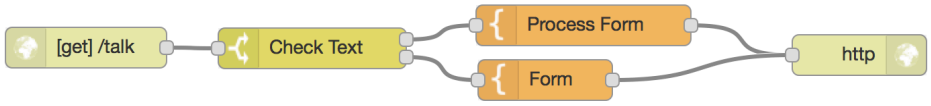
node with the HTML shown below.



Get the code:
`ibm.biz/Bd4vtx`

5. Add a 

and connect the nodes together as follows:



6. Click 

to save the changes. Test the flow by opening the application URL in the browser, appended with /talk.

`https://<<MY-APP>>.mybluemix.net/talk`