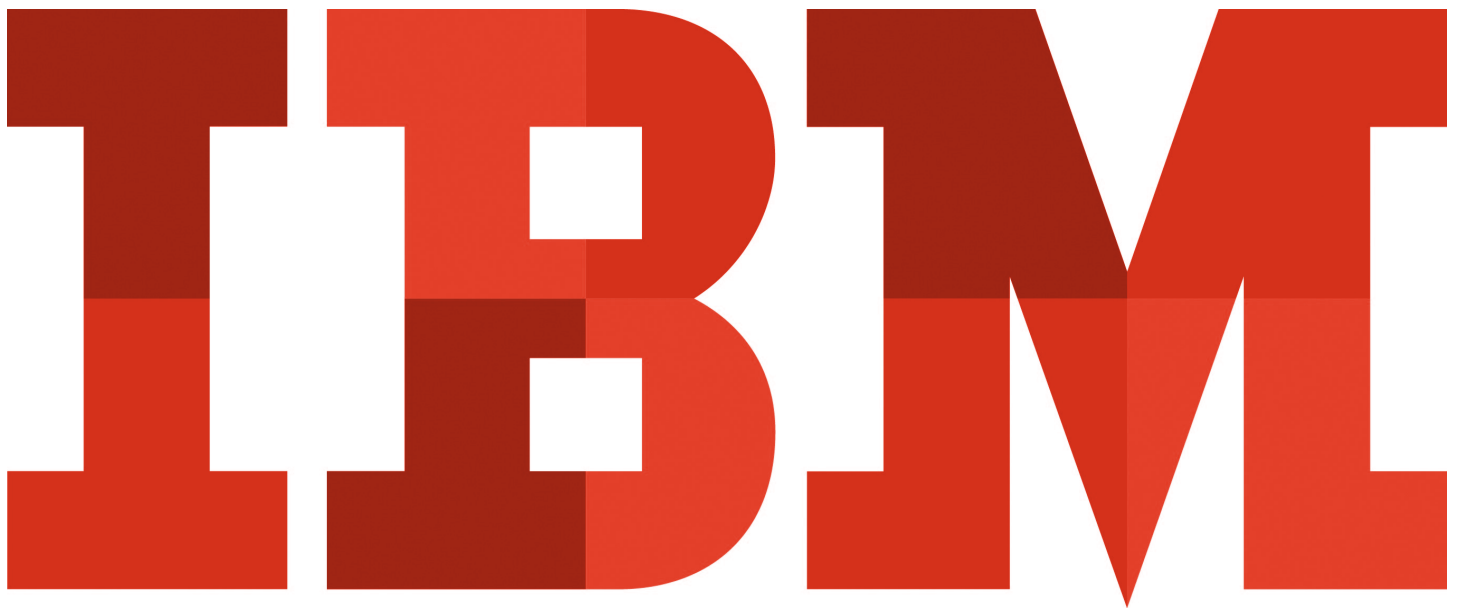


# Language Translation in Node-RED

## Hands-On Lab

JeanCarl Bisson | [jbisson@us.ibm.com](mailto:jbisson@us.ibm.com) | [@dothewww](#)



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



A digital copy of this lab and code snippets can be found at:  
<http://ibm.biz/node-red-language-translation>

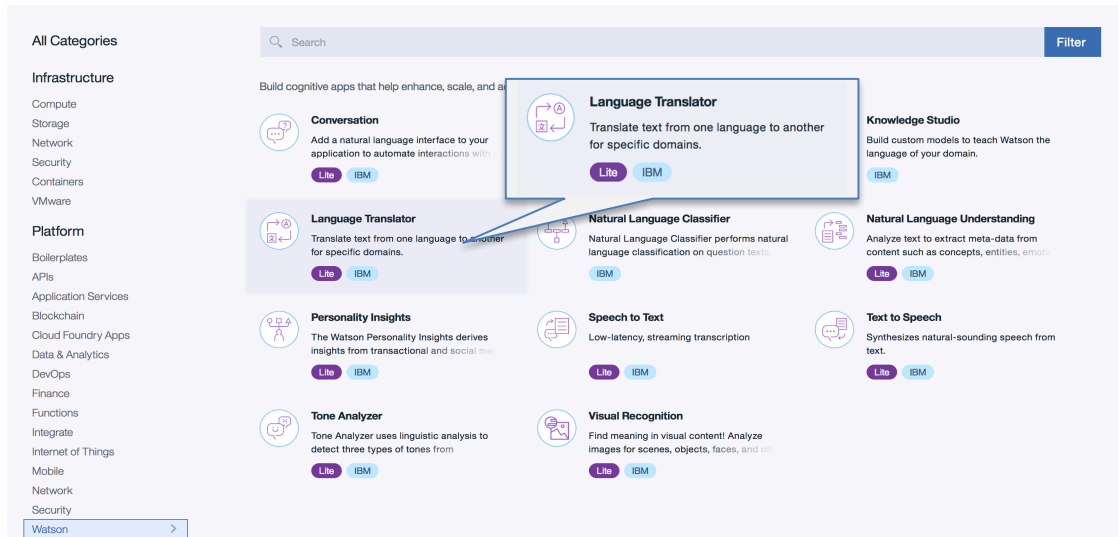


# Add Language Translator Service in IBM Cloud

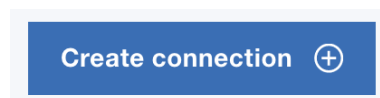
The IBM Watson Language Translator service enables you to translate text from one language to another. These domains are supported:

- The News domain - targeted at news articles and transcripts, it translates English to and from French, Spanish, Portuguese or Arabic.
- The Conversational domain - targeted at conversational colloquialisms, it translates English to and from French, Spanish, Portuguese or Arabic.
- The Patent domain - targeted at technical and legal terminology, it translates Spanish, Portuguese, Chinese, or Korean to English.

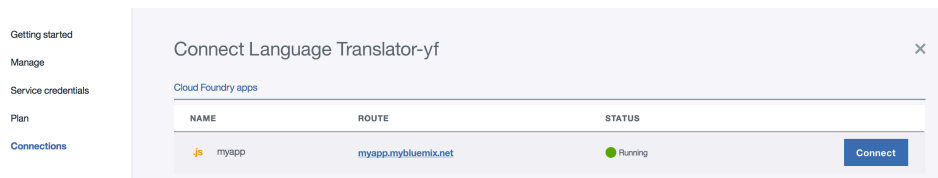
1. Click on the **Catalog** link in the top-right of the IBM Cloud Dashboard. Under the Watson section, click on the **Language Translator** tile.



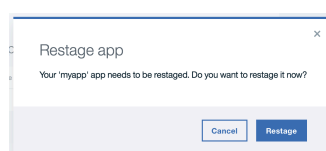
2. You can optionally give the service a custom name or leave it as the one given. Click **Create**.
3. Click on **Connections** in the menu on the left.
4. Click **Create connection** on the right.



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



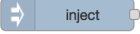
6. IBM Cloud will prompt to restage the application. Click on **Restage**. The application will restart and include the new service credentials in the environment.

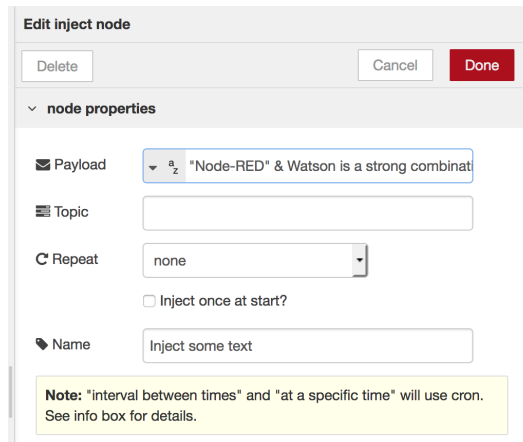


7. When the application has finished restaging, open the Node-RED Flow Editor. If you already have Node-RED open, refresh the page.

# Add Language Translator in Node-RED

In this section, we will add a Language Translator node and translate text input into French. In this example, some example text (in English in this case) is injected to French and the result outputted to the Debug tab. Using an Inject and Debug node can be a helpful first step to understanding how a flow might be constructed.

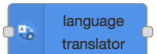
1. Add an  node as shown below (you can use any text for this):

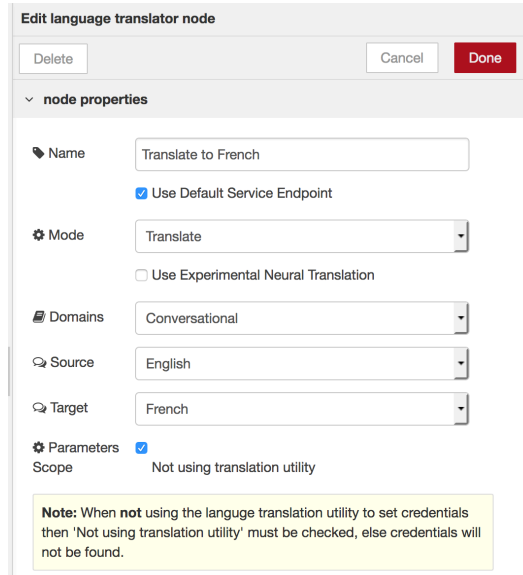


The 'Edit inject node' dialog shows the following configuration:

- Payload:** "Node-RED" & Watson is a strong combinat
- Topic:** (empty)
- Repeat:** none
- Inject once at start?:** ☐
- Name:** Inject some text

Note: "interval between times" and "at a specific time" will use cron. See info box for details.

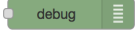

2. Add a  node as shown below. The text in this example is English, so select **English**. Based on your source choose the right domain: News or Conversational.



The 'Edit language translator node' dialog shows the following configuration:

- Name:** Translate to French
- Use Default Service Endpoint:** ☒
- Mode:** Translate
- Use Experimental Neural Translation:** ☐
- Domains:** Conversational
- Source:** English
- Target:** French
- Parameters:** ☒
- Scope:** Not using translation utility

Note: When **not** using the language translation utility to set credentials then 'Not using translation utility' must be checked, else credentials will not be found.

3. Add a  node. The translated text will be returned in `message.payload` and displayed in the Debug tab.
4. Connect the nodes together as shown below and click on the red  in the upper-right of the screen.

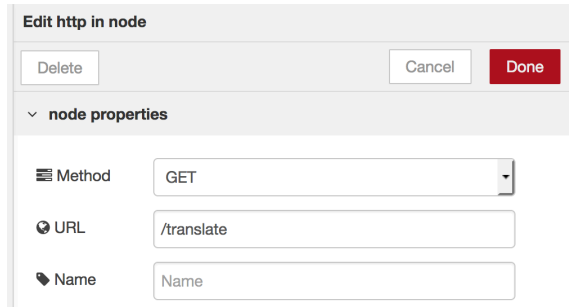


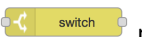
5. Click on the tab on the left side of the inject node to start the flow. The output from the debug node will show up in the debug tab in the right-hand pane.

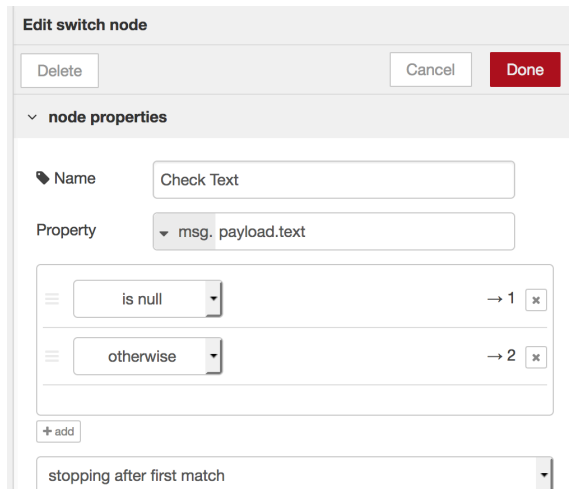
# Creating an Interactive Web UI

In this section, we will create a simple webpage that accepts user input, translates it into French, and then displays it on the webpage.

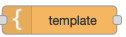
1. Add a  as shown below.

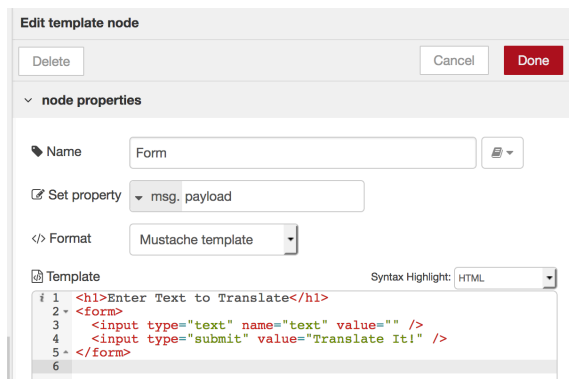


2. Add a  node as shown below. This node checks whether the text parameter is passed in.



If webpage is called with a query parameter named text, the first flow will be processed. Otherwise, the second flow will be processed.

3. For the first flow, add a  node to display a form for the user to input text to translate as shown below.



```
1 <h1>Enter Text to Translate</h1>
2 <form>
3   <input type="text" name="text" value="" />
4   <input type="submit" value="Translate It!" />
5 </form>
6
```




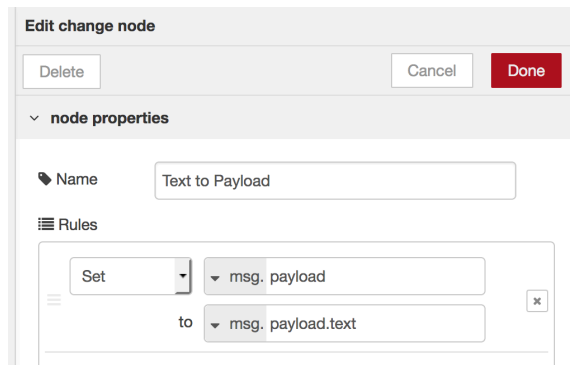
Get the code:  
[ibm.biz/Bd4mTi](https://ibm.biz/Bd4mTi)

4. Add a  node and connect the nodes together as shown below.



This will display the form when the user opens a browser and goes to the application URL, appended by /translate.

5. Next, add a  node to extract the text from the query parameter and put it in the msg.payload for the Language Translation node.



**Edit change node**

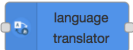
Delete Cancel Done

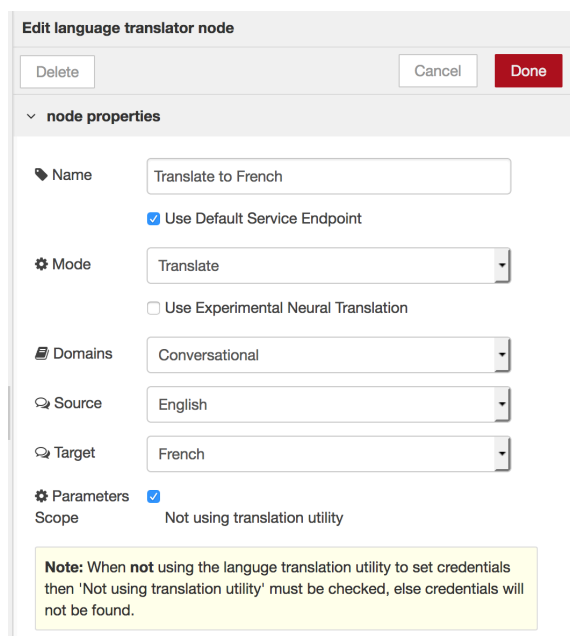
node properties

Name: Text to Payload

Rules

Set to msg.payload to msg.payload.text

6. Add a  node as shown below. For this example, we'll translate English text into French.



**Edit language translator node**

Delete Cancel Done

node properties

Name: Translate to French

☒ Use Default Service Endpoint

Mode: Translate

☐ Use Experimental Neural Translation

Domains: Conversational

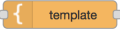
Source: English

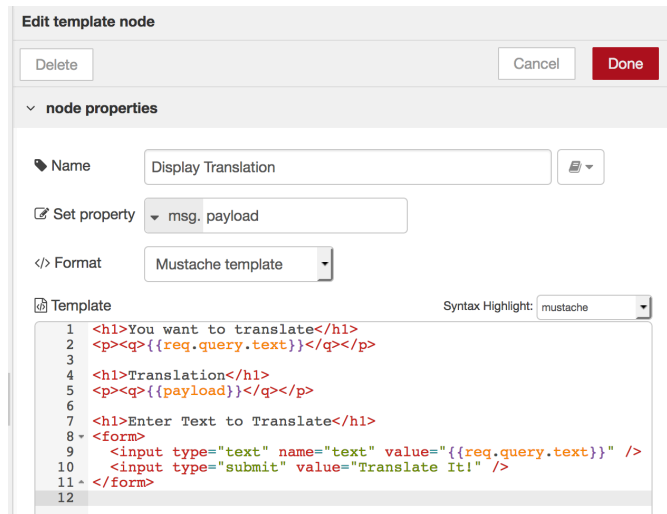
Target: French

Parameters: ☒

Scope: Not using translation utility

**Note:** When **not** using the language translation utility to set credentials then 'Not using translation utility' must be checked, else credentials will not be found.


7. Add a  node to display the result along with the form for the user to translate another piece of text.



```
1 <h1>You want to translate</h1>
2 <p><q>{{req.query.text}}</q></p>
3
4 <h1>Translation</h1>
5 <p><q>{{payload}}</q></p>
6
7 <h1>Enter Text to Translate</h1>
8 <form>
9   <input type="text" name="text" value="{{req.query.text}}" />
10  <input type="submit" value="Translate It!" />
11 </form>
12
```



Get the code:  
[ibm.biz/Bd4mTZ](https://ibm.biz/Bd4mTZ)

8. Connect the nodes together as shown below, and click on  to save the changes.



9. Test the application by visiting the application URL, appended with /translate.

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