# IoT Domain Analyst

## Lab Record — Lab 7

- 15 April 2021

| | | | | | |
|---|---|---|---|---|---|
| Programme | : | B.Tech(CSE) | Semester | : | Winter 2020–21 |
| Course Title | : | IoT Domain Analyst – Lab | Code | : | ECE3502 |
| | | | Slot | : | L5+L6 |
| Name | : | **Aadhitya Swarnesh** | Registration. No | : | [redacted] |
| Faculty (s) | : | [redacted] | Expt. No | : | **7** |

## Experiment 1 :

**Exchange data between your Local Node red and to another Node red instance hosted in a server.**

## Aim :

To establish a network connection between our local node-red instance and another node-red instance hosted in a remote cloud server using an FRED instance, and use this connection to transfer data and messages between these node-red instances.

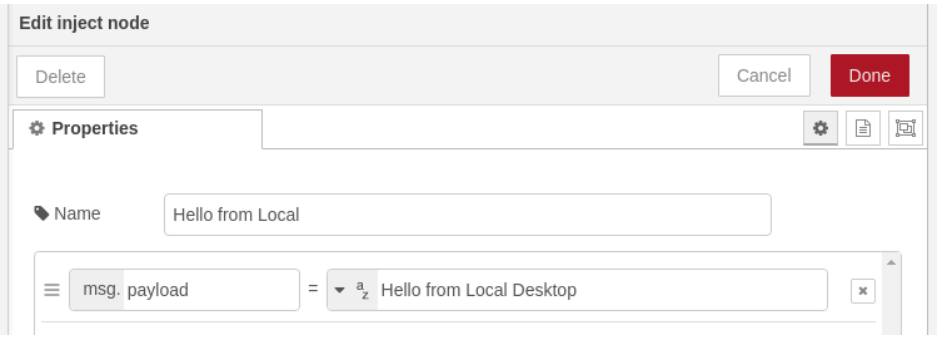
## Description :

- Start up a FRED instance in a remote server by signing up with them.
- Use the FRED nodes in the node-red , to establish a working network connection between our local flow and the one hosted in FRED.
- Send a message from our local flow and see it displayed in the FRED's debug window, and to send a message from the FRED flow and see it in the local Node-red's debug window.

# Flow Diagrams :

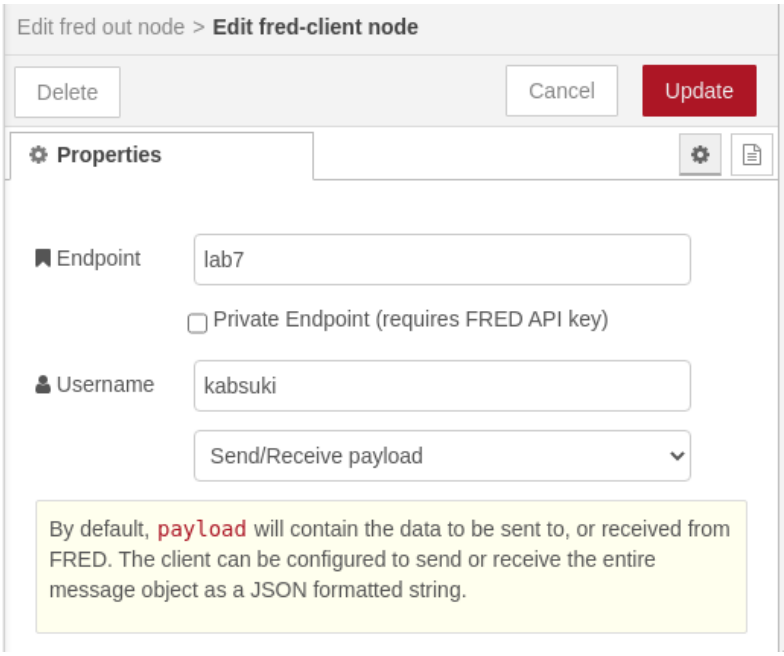
### Stage 1 : Inject node in local Node Red Flow

We insert an “inject node” into our local node red flow. This is used to inject a message into the flow which will later be passed to the FRED flow. It contains a sample message in its payload, and its configurations are as follows :




### Stage 2 : FRED Client in local Node Red Flow

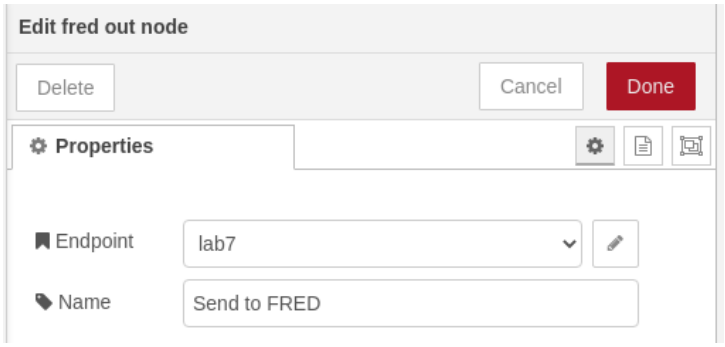
We establish a link to the FRED flow by specifying the FRED username, and by specifying an end point to connect. Look at the diagram for a sample :




This same endpoint is to be specified in the FRED flow as well, and this has to be configured for both ways.

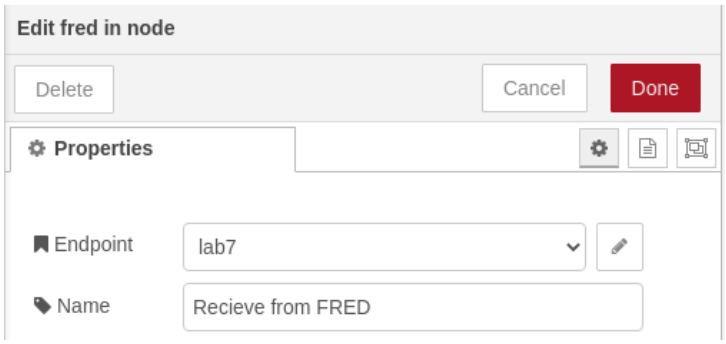
### Stage 3 : FRED Out node in local Node Red Flow

We insert an FRED OUT node into the local flow which is used to pass the data from the local flow to the FRED flow. Its configuration is as follows :



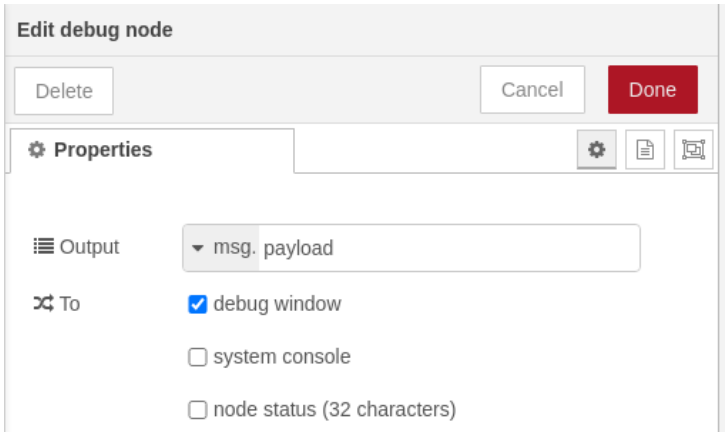

### Stage 4 : FRED In node in local Node Red Flow

We insert an FRED IN node into the local flow which is used to receive the data from the FRED flow and passes it to the Debug window for display. Its configuration is as follows :



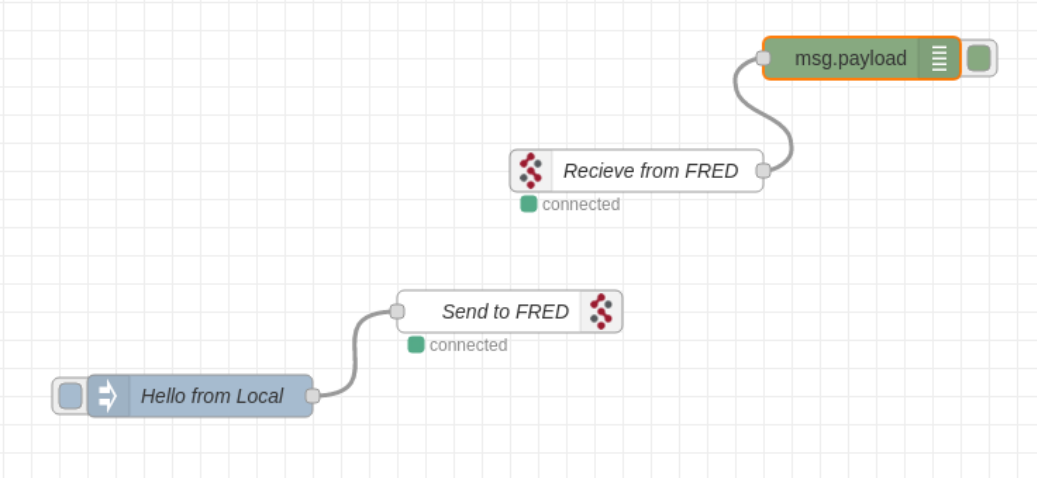

### Stage 5 : Debug node in local Node Red Flow

We insert a Debug node in the local flow so as to display the data from the FRED flow in the debug window of the local node-red flow. Its Configurations are standard and are as follows :



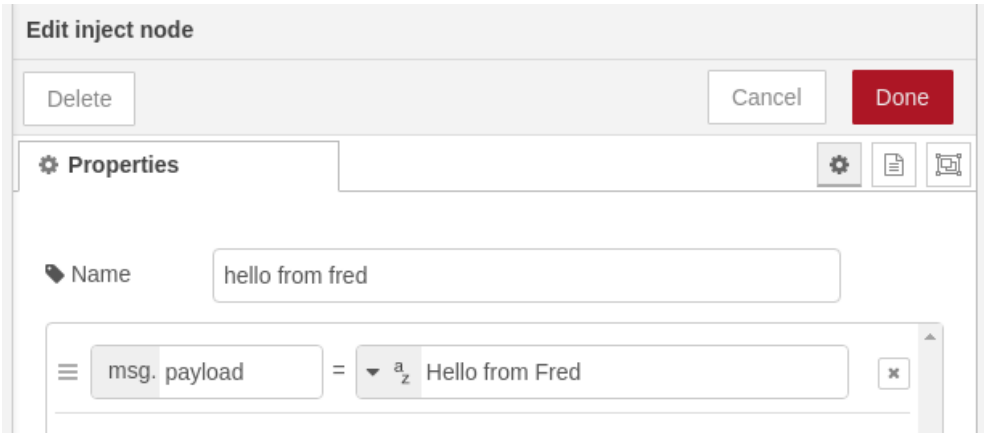
### Stage 6 : Complete local Node Red Flow

The overall flow in the local node-red runtime looks as follows, it has two parts, one is for sending data to a FRED runtime, and the other is for receiving data from the FRED runtime.



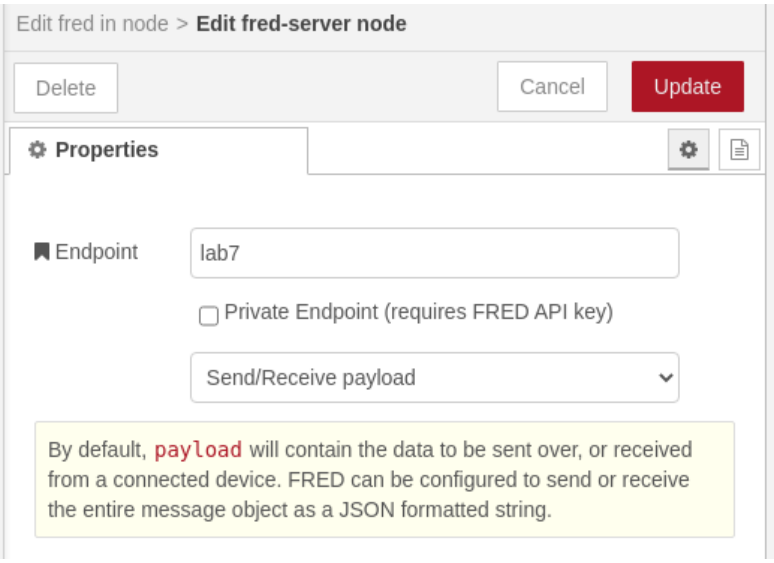

### Stage 7 : Inject node in FRED Flow

We insert an “inject node” into our FRED flow. This is used to inject a message into the flow which will later be passed to the local node-red flow. It contains a sample message in its payload, and its configurations are as follows :



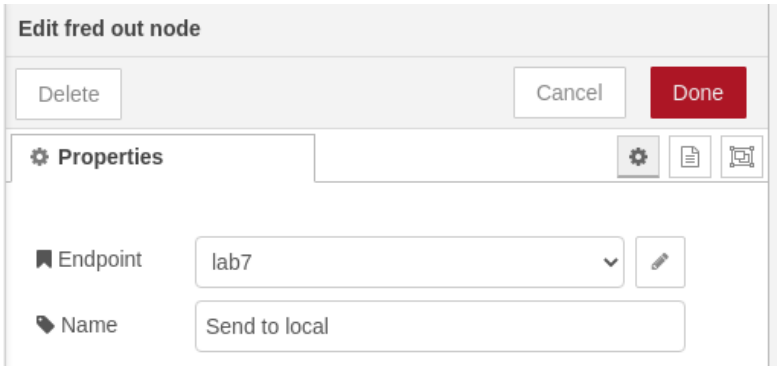
### Stage 8 : FRED server in FRED Flow

We establish a link to the local node-red flow by specifying the end point to connect. This endpoint should be the same as the one specified in the FRED client in the local node-red flow. Look at the diagram for a sample :



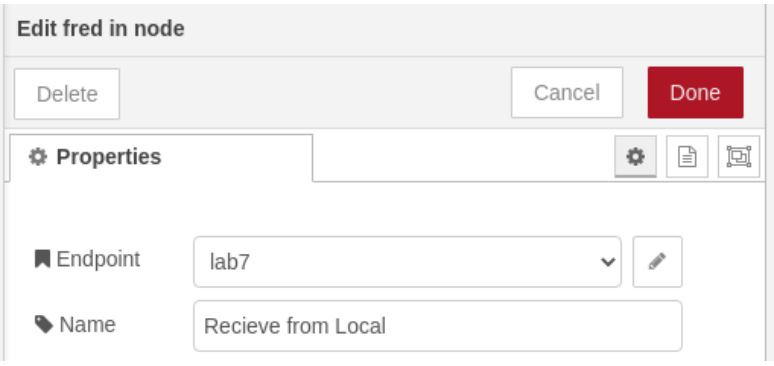

### Stage 9 : FRED OUT in FRED Flow

We insert an FRED OUT node into the FRED flow which is used to pass the data from the FRED flow to the local node-red flow. Its configuration is as follows



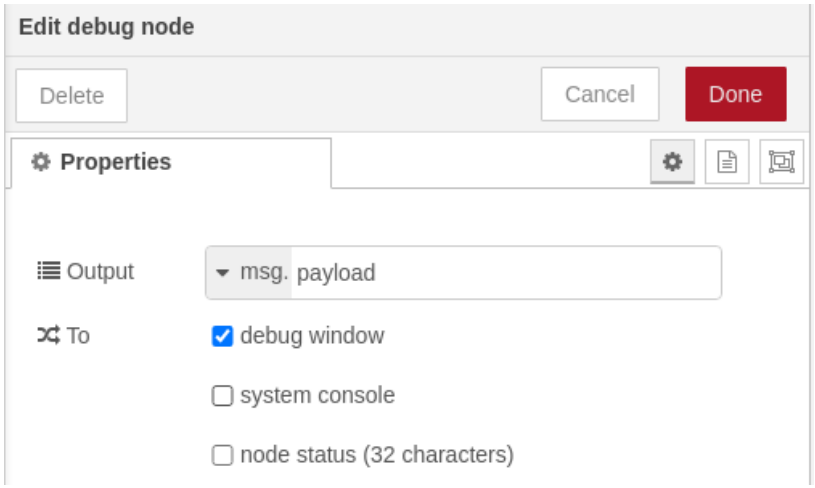

### Stage 10 : FRED IN in FRED Flow

We insert an FRED IN node into the FRED flow which is used to receive the data from the local node-red flow and passes it to the Debug window for display. Its configuration is as follows :



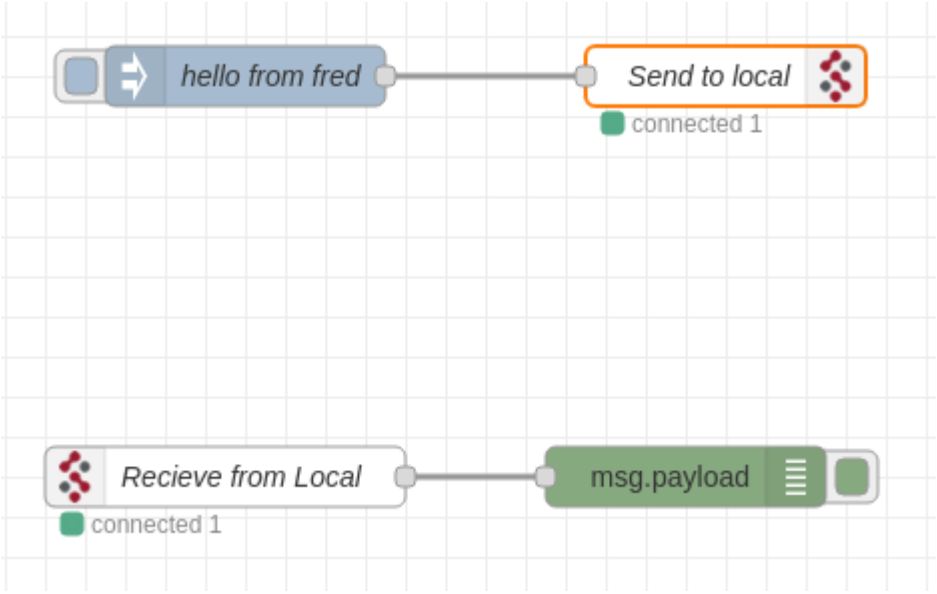
### Stage 11 : Debug in FRED Flow

We insert a Debug node in the FRED flow so as to display the data from the local node-red flow in the debug window of the FRED flow. Its Configurations are standard and are as follows :




### Stage 12 : Complete FRED Flow

The overall flow in the FRED runtime looks as follows, it has two parts, one is for sending data to a local node-red runtime, and the other is for receiving data from the node-red runtime and to display its data in its debug window.

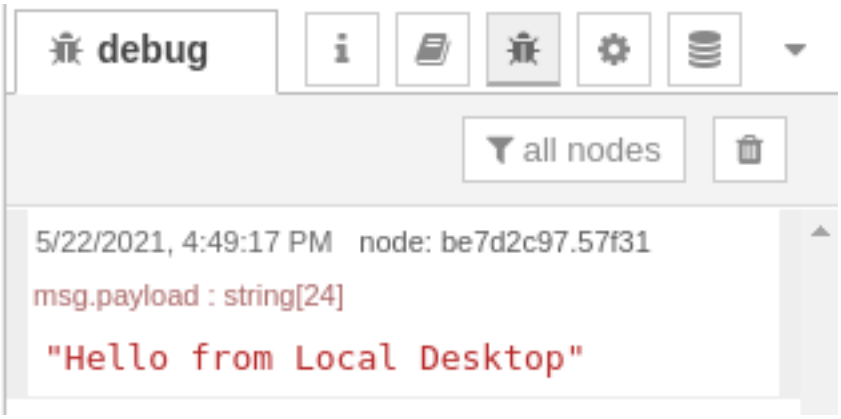



It is absolutely necessary that the endpoint configured in both the FRED client and server should be the same for this exchange of data to take place. Otherwise these nodes will not show that “Connected” green dot below, and the message transfer would fail.

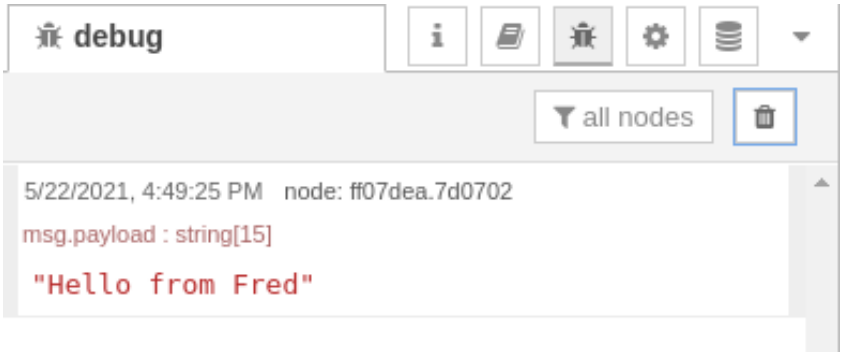
## Result :

As mentioned in the stages above, we have performed the experiment and have produced successful results.

We have sent a message form the Local Node red flow which has been received by the FRED flow and is displayed in its debug window. This diagram below demonstrates this event.




On the other hand, we have also deployed a reverse of the above event, where we have sent a message from the FRED flow which has been received by the local node-red flow and is displayed in its debug window. The output for the same looks as follows :




With both these parts working, we have successfully implemented a Message Passing system between a client which here is our local Node red flow, and the server which in this case is a FRED instance deployed in a remote server. This has endless possibilities of usage, and has been implemented successfully using **Node-Red**, **FRED** and the concepts of **IoT**.