Node RED on IBM i Workshop

Andy Youens

FormaServe Systems Ltd

IBM Champion 2021, 2022 & 2023

Presenter - Andy Youens

IBM i Consultant/Instructor at FormaServe for over 33 years

Over 40 years IT experience working with S/34, S/36, AS400, iSeries & IBM i

Specialties RPG Node.js PHP DB2 & Open Source on IBM i

Instructor & article writer

IBM Champion 2021, 2022 & 2023

Certified IBM Developer

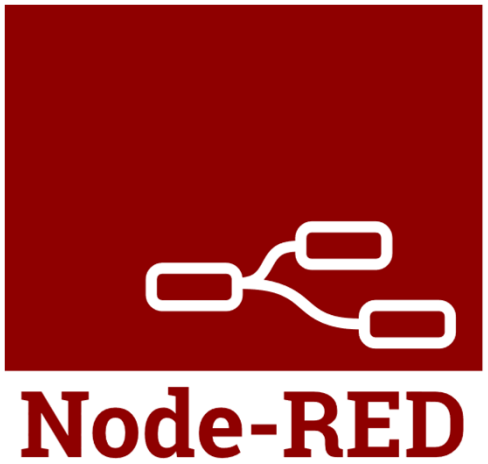
Prior to IT, Andy is proud to have served 10 years in the Royal Navy

- We are proud to be a leading IBM i software vendor with a remarkable track record of over 33 years
- We offer comprehensive IBM i training courses, both remotely & on-site, to equip individuals & teams with the necessary skills & knowledge
- As open-source on IBM i specialists, we possess deep expertise in leveraging open-source technologies to enhance & optimise IBM i environments
- Our team comprises of modernisation experts who can guide you in transforming your IBM i to embrace modern practices & technologies
- Part of IBM ISV Advisory Council
- Provide Open-Source support to i-UG education initiative

FormaServe

- FormaServe & NC Communications have announced a strategic partnership to publish PowerWire
- Do you want to write a blog for us & share your insights & experiences with the IBM i community?
- Do you want to advertise with us & reach thousands of potential customers & partners in the IBM i market?
- Give me a shout!

Agenda



- Node-RED Introduction
- IBM i Installation
- Messages
- Node-RED
- IBM i Integration
- Node-RED Extras

Node-RED Intro

Overview

- Code generator developed for communicating with IOT devices
- Initially developed by IBM in 2013
- Built on Node.js
- Maintained by OpenJS Foundation
- Open Source - join in!
- Generates JavaScript & Node.js code
- A visual programming tool
- Just like a flow-diagram!

Node-RED Platforms

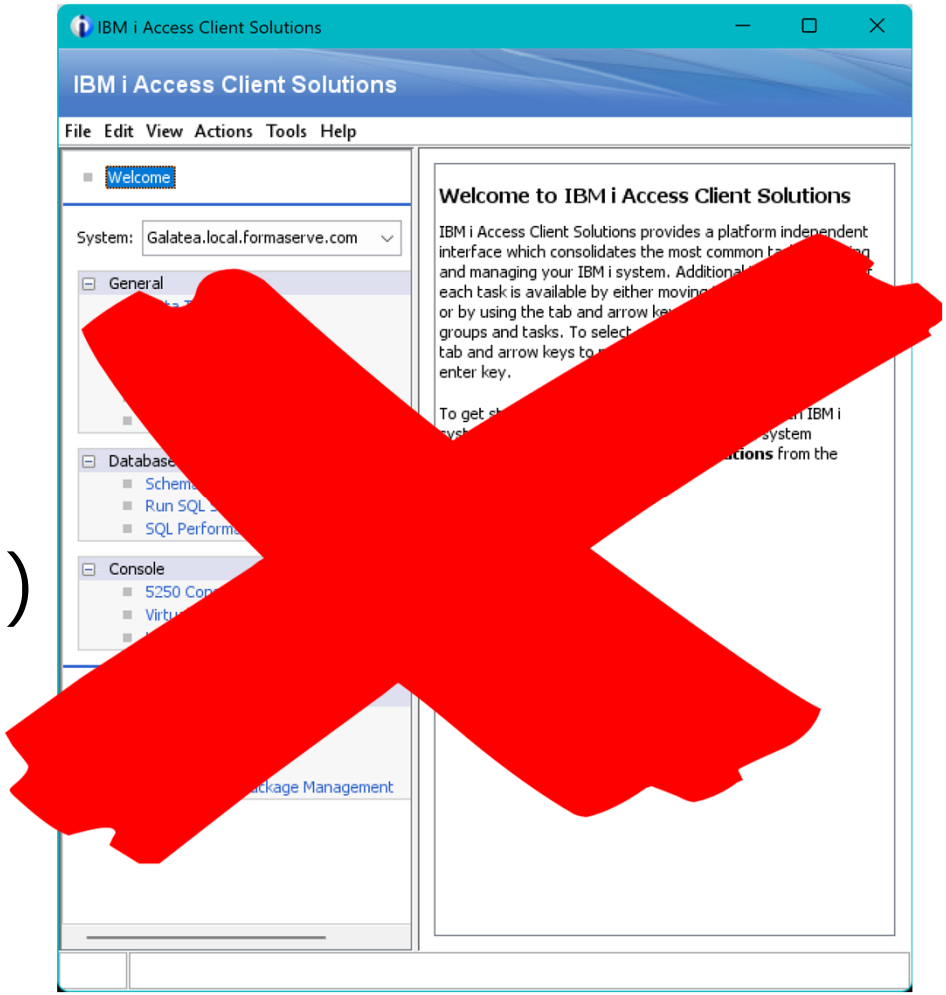
- Red Hat/Fedora/CentOS
- Raspberry Pi
- Docker
- Windows
- Android
- AWS
- Azure
- IBM i

Installing Node-RED

Installing Node-RED on IBM i

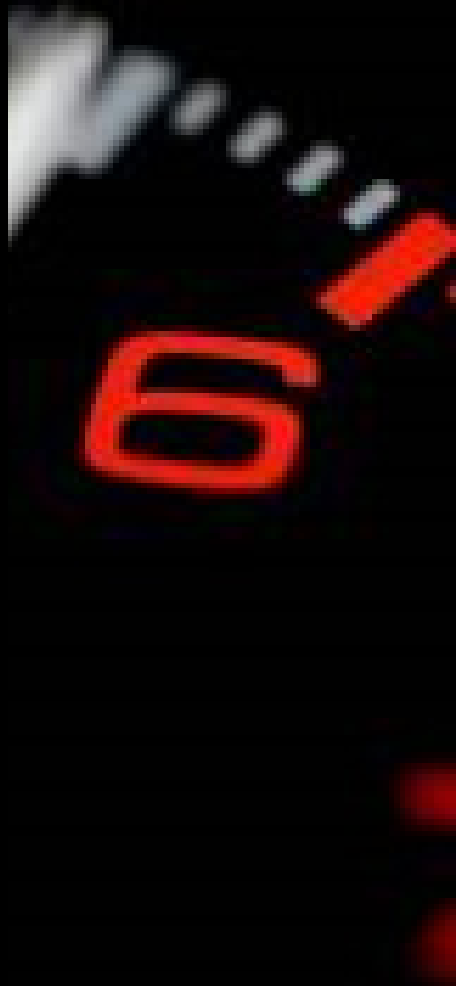
- Pre-Reqs
  - Node 18 or > installed
  - PASE (5770ss1 Option 33)
  - OpenSSH (5733SC1 Option 1)

- Version 3.1.0 current (Nov 23)
- No ACS or Yum install
- NPM is used to install
  - npm install -g node-red



Node-RED - Two Parts

1. Run-time
   - Runs your flows
2. Editor
   - Browser-based editor

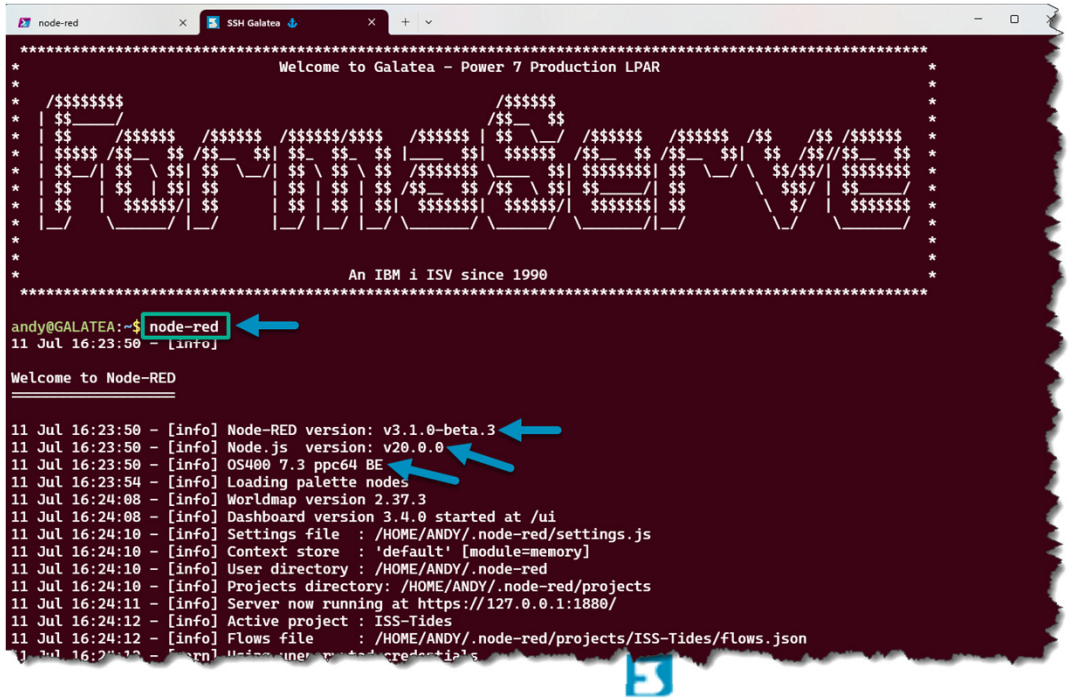


Node-RED Ports

- Both the server & editor runs on TCP/IP port number 1880
- Each user has its own flow & settings!
- For multiple users, change the server port number
  - Use environment variables (EXPORT PORT=1881)
  - Change users BASH profile (~/.profile)
  - Change Node-RED settings file (settings.js)
  - Specify port number when starting server
    - node-red -p 1881
- Use scopenports to check ports available

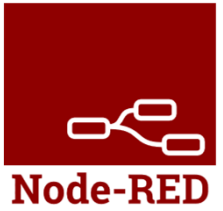
Starting Node-RED

- You have to start the server, in a shell session
  - /QOpenSys/pkgs/lib/nodejs20/bin/node-red
  - Just node-red if in path



Check for errors!

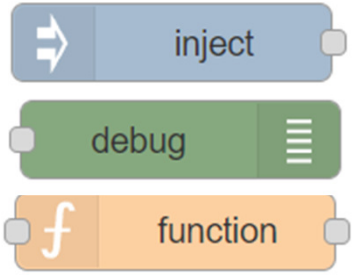
Node-RED Terminology



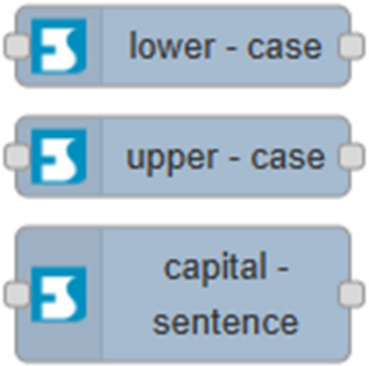
- **Flows**
  - Flows are visual code, or programs
- **Nodes**
  - Nodes are small pieces of code doing a specific job
- **Connectors**
  - Connectors are the small squares attached to the nodes
- **Wires**
  - Wires shows the flow from one node to another
- **Deploy**
  - Creates the program that is run

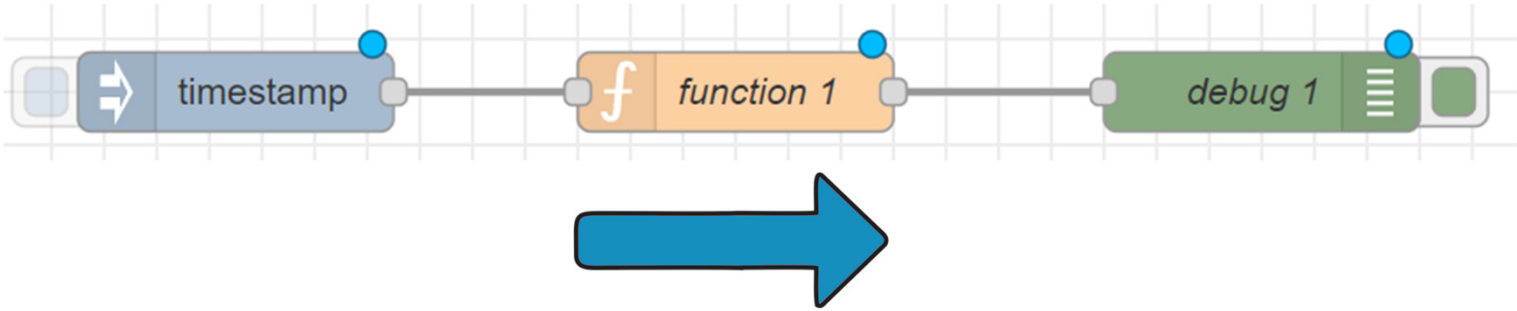
Node-RED Nodes

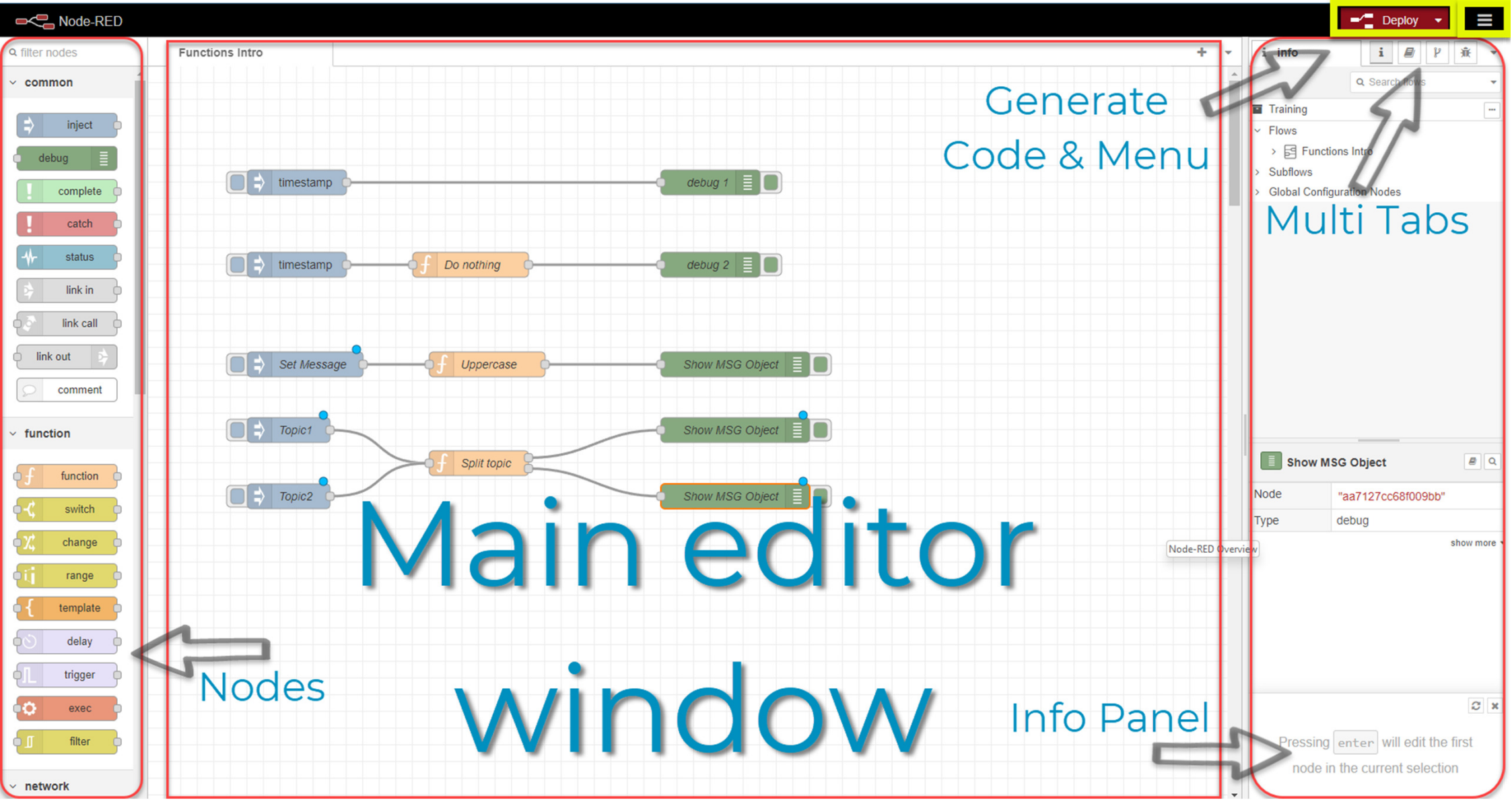
- There are three main types of nodes
  - Input Nodes (e.g., inject)
  - Output Nodes (e.g., debug)
  - Processing Nodes (e.g., function



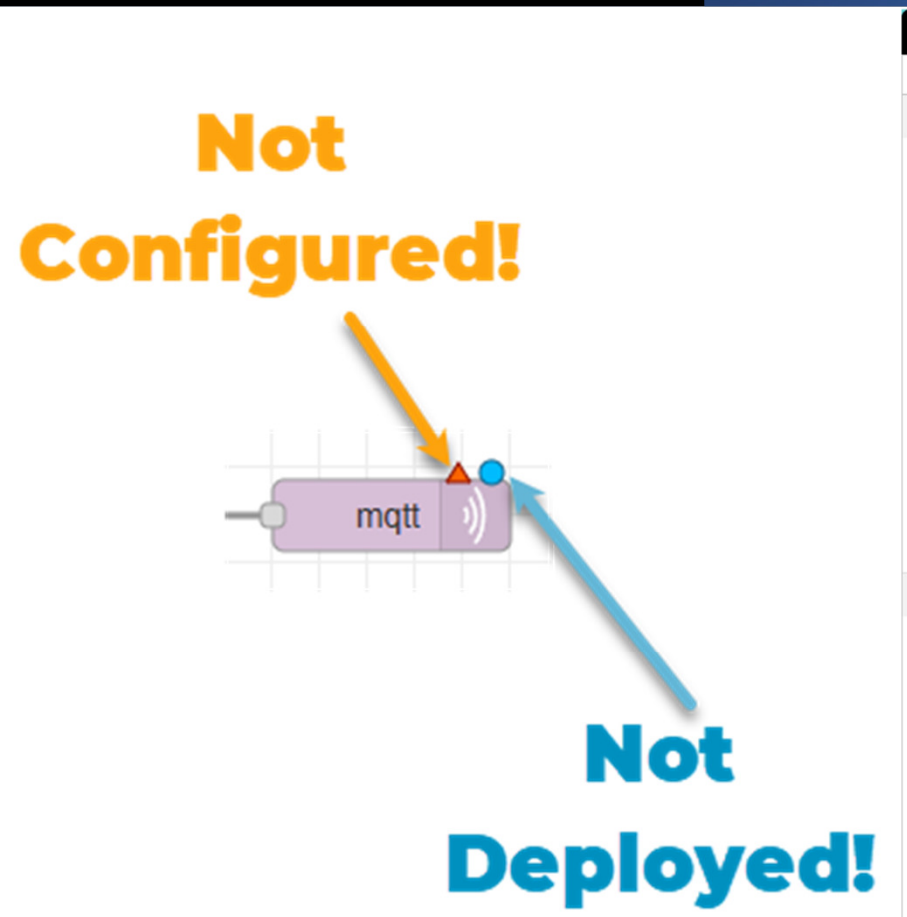
- No need to 're-invent the wheel!'
- You can create your own!

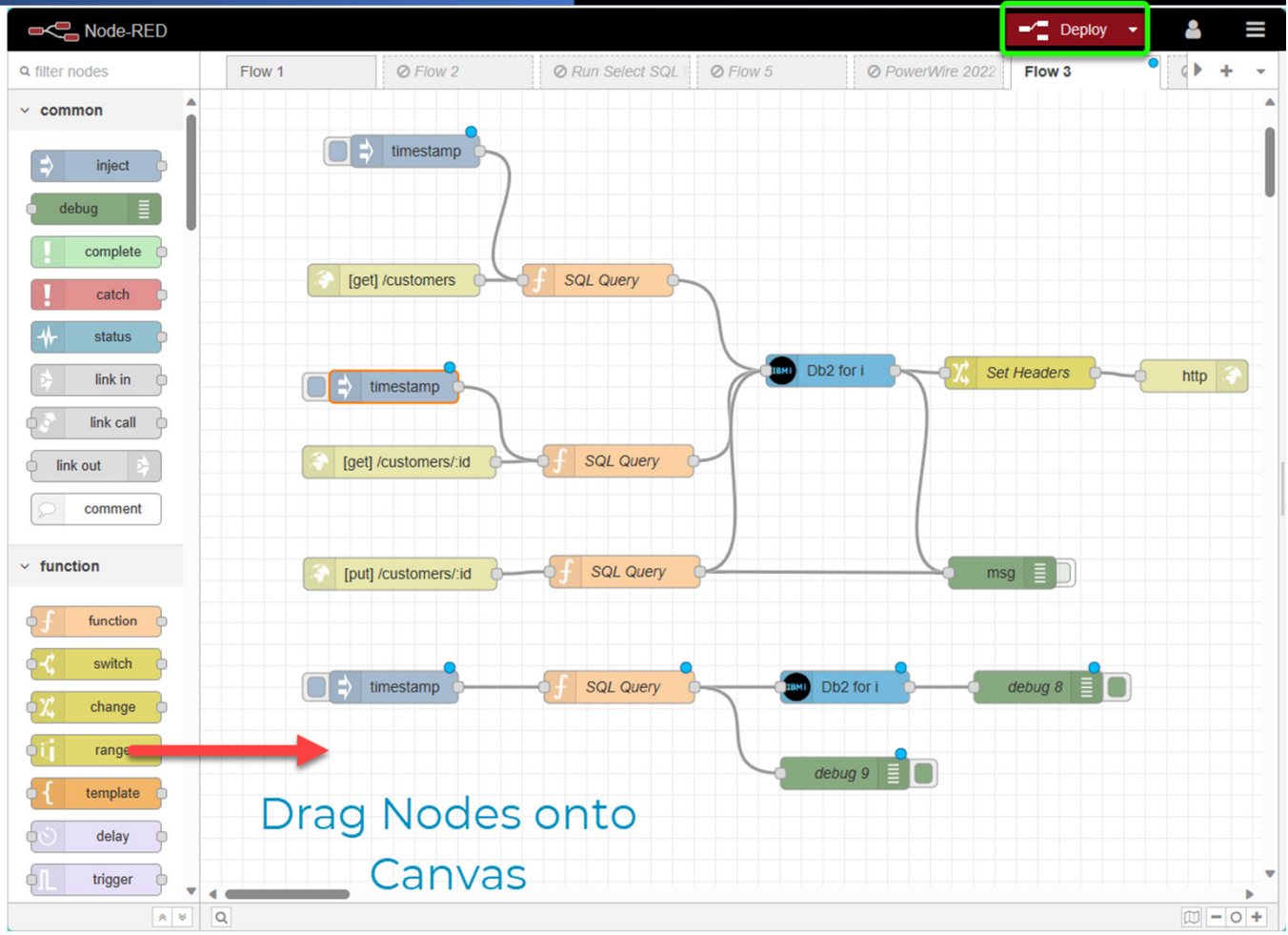






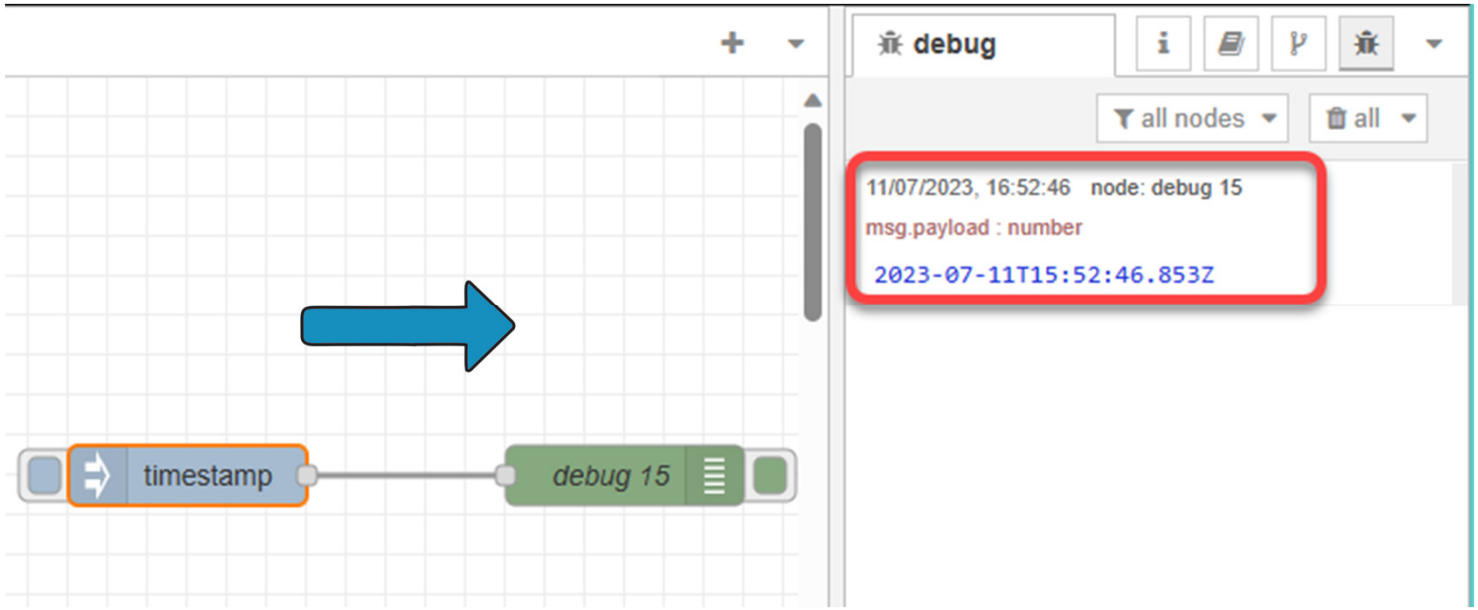
Node-RED Canvas





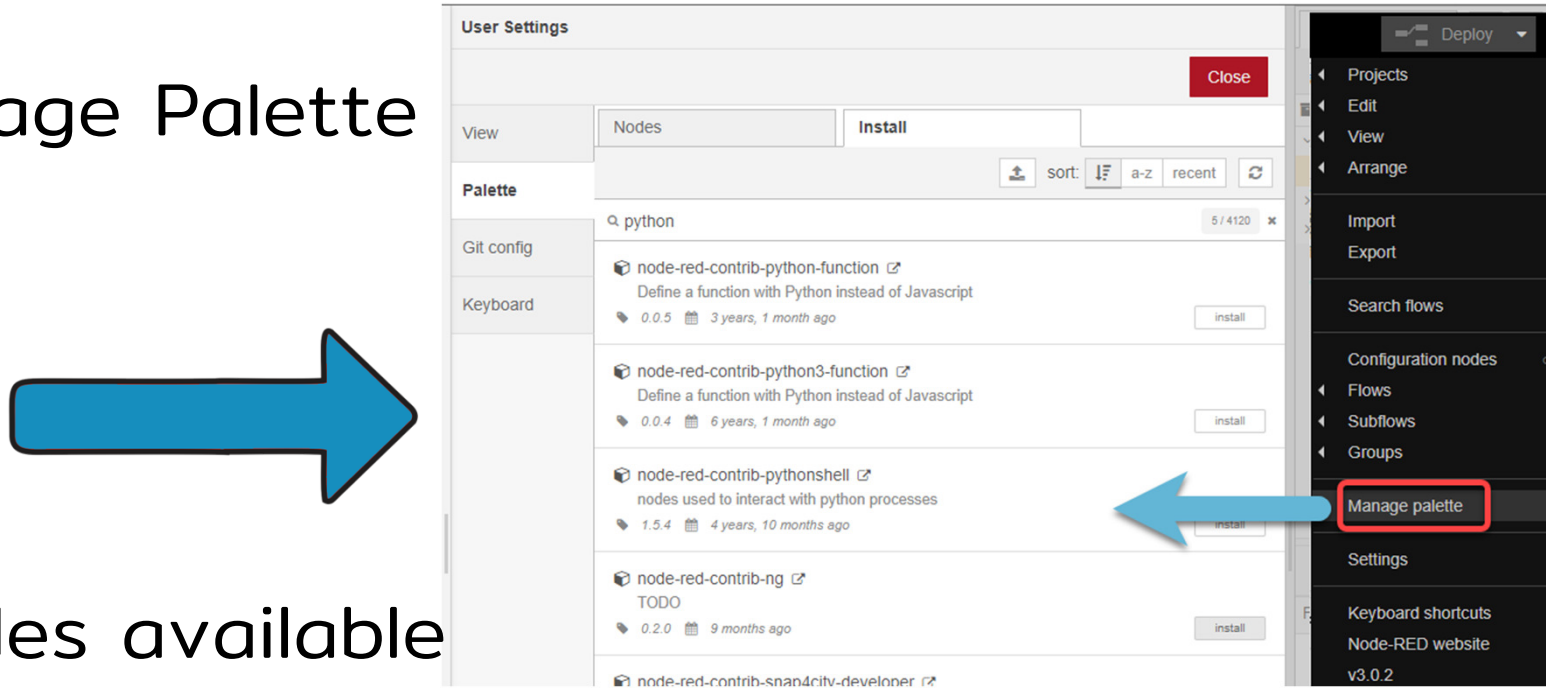
Demo & Exercise

- Explain the Node-RED canvas
- Create a first flow by wiring a timestamp node to a debug node



Node-RED Adding Extra Nodes

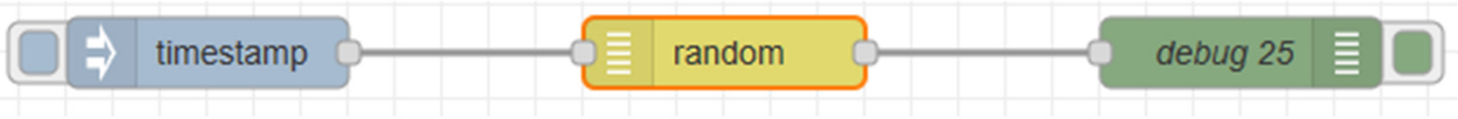
- Extra nodes can be added that extends the core of Node-RED
- From menu
  - Select Manage Palette

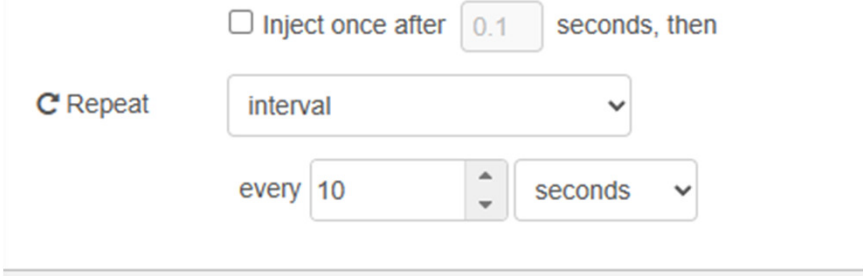


- Over 4.5k nodes available

Demo & Exercise

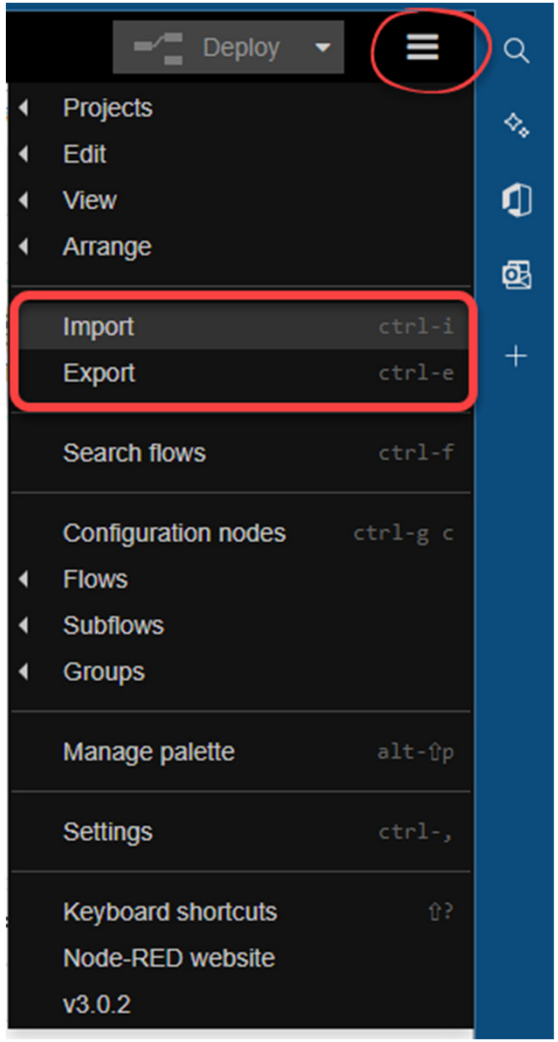
- Add the random node to your palette (node-red-node-random)
- Create a flow using the random node picking a number from 1 to 100
- Make your flow run automatically every 10 seconds





Flows

- Import
  - Import from your file/clipboard
- Export
  - Export from your file/clipboard
- Both great for
  - Taking backups
  - Storing on GitHub
  - Support
  - Experimental, POC etc



Demo & Exercise

- I'll use one of the many flows on the Node-RED website https://flows.nodered.org/
- I'll import it into a new flow in my browser
- I'll deploy it & run it

Node-RED Messages

- The message gets passed between nodes
- It is the only method of communicating between nodes
- It's a JavaScript object
- It has properties
  - msg.carManufacturer = Ford
  - msg.carModel = Capri
- Messages usually have payload & topic property

msg.payload may get overwritten!

Node-RED Message Object

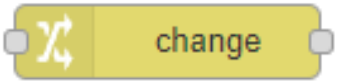
- Our car → message (abbrev. to msg throughout)
- It goes from A to B → from one node to another node
- It has properties
  - Manufacturer, make & model
  - It has a driver
  - It can have passengers & luggage
  - It has 4 wheels (array)
  - The wheels can have different tyres manufacturers

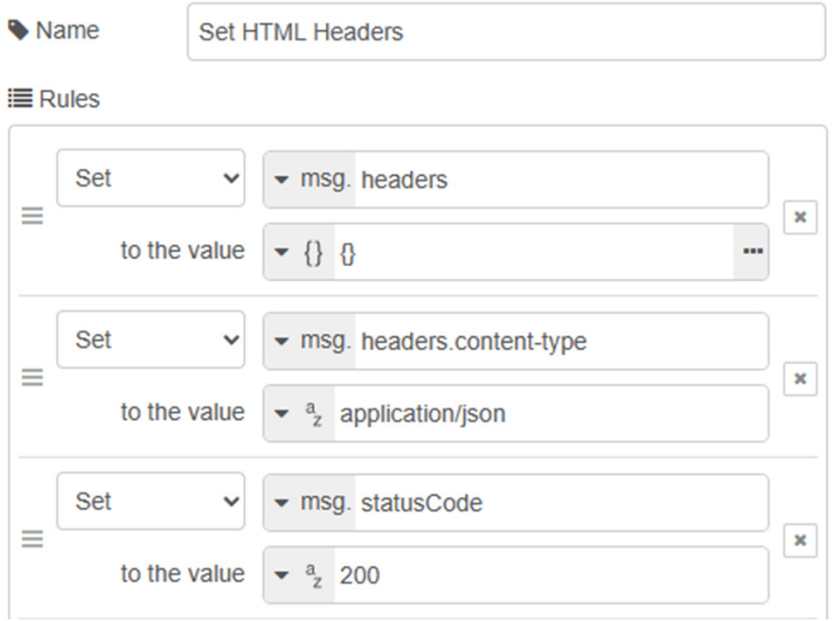
Node-RED Message Object

- There are 2 ways you can change the properties of the message object
  1. **Change node** - No programming required
  2. **Function node** - Manipulate the object using JavaScript

Node-RED Change Node

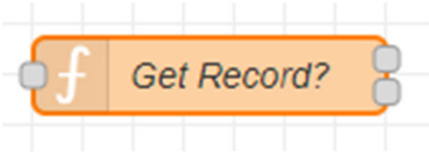
- It provides four basic operations:
  - Set a property to a value
  - Change a string property by search & replace
  - Delete a property
  - Move a property
- No programming skills!





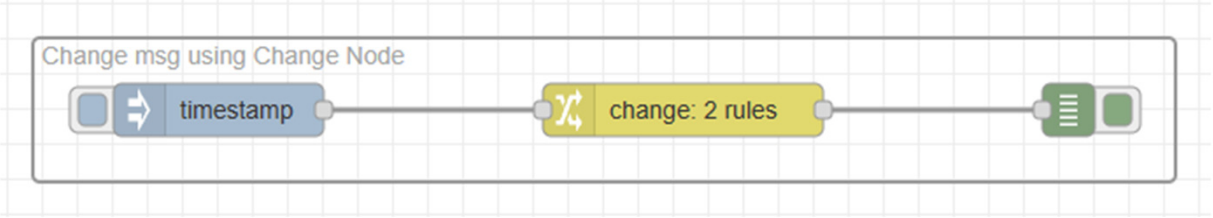
Node-RED Function Node

- Is used to run JavaScript code against the message object
- It accepts a message object as input & returns message objects as output
- The message object can have a payload property msg.payload & usually has other properties depending on the proceeding nodes
- Can have more than one output (not seen that often)



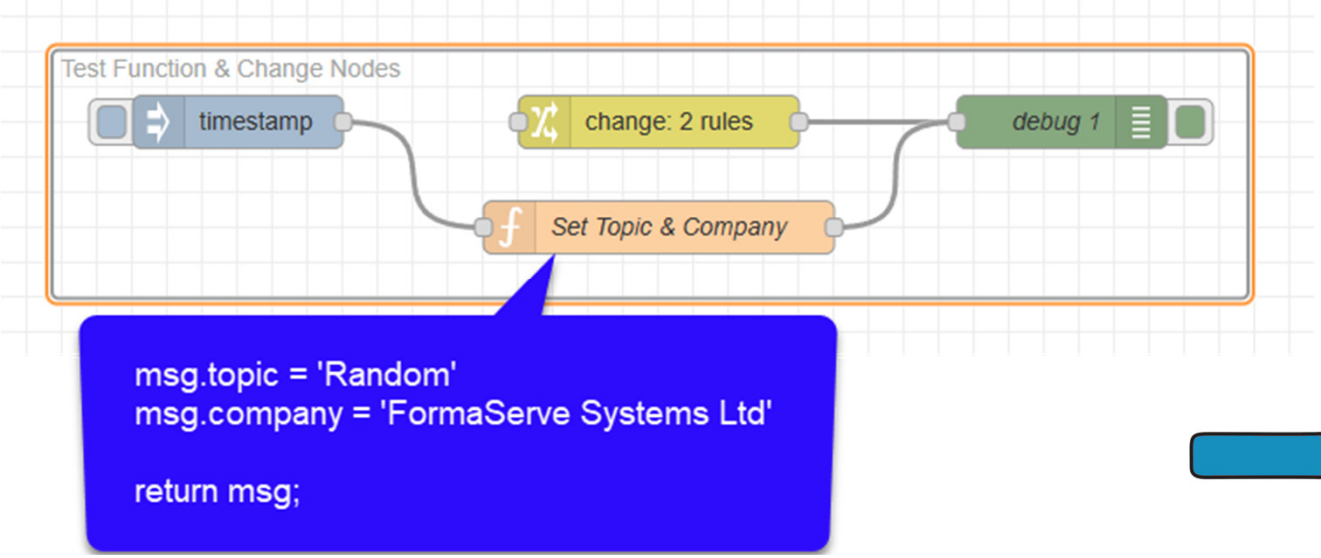
Demo & Exercise

- Use a change node to change the msg.topic to have a value of Random
- Use the change node to give the msg.company variable some name
- Deploy & run your flow



Demo & Exercise

- Use the function node instead of the change node



What is JSON ?

JSON

- JSON stands for JavaScript Object Notation
- JSON is a lightweight format for storing & transporting data
- JSON is self-describing & easy to understand
- Very powerful
- Supported by many languages
- A replacement for XML? - Yes, in many cases!

JSON

{ JSON }

- JSON format is text only
- Data is in name value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays
- Objects can contain arrays
- Arrays can contain objects

No comments allowed!

JSON Examples

- A name:value pair consists of a field name with a value
- In double quotes (no single quotes!)
- Name, followed by a colon, then the value

```
{ "name":"andy" }
```

JSON Value Types

- A string must be in double quotes
- A number
- Boolean (true or false)
- Null
- An object { }
- An array [ ]

```
{
    "name": "f-weather",
    "version": 1,
    "private": true,
    "developer": null
}
```

JSON Examples

Basic

```
{
    "name": "Andrew",
    "Born": "London",
    "age": 21,
    "car": "Capri"
}
```

Object

```
{
    "employee": {
        "name": "Andy",
        "salary": 56000,
        "married": true
    }
}
```

Array

```
{
    "cars": [
        "Ford",
        "BMW",
        "Fiat"
    ]
}
```

Object & Array

```
{
    "id": 7,
    "name": "Andrew Youens",
    "age": 21,
    "hobbies": {
        "indoor": [
            "Home Automation"
        ],
        "outdoor": [
            "Football",
            "Cricket"
        ]
    }
}
```

Node-RED & IBM i

Node-RED DB2 for i Adapter

- Latest version 0.2.3 dated 15/12/2022
- DB2 for i adapter - 2 Nodes
  - One node for configuration
  - One node for SQL
- Input
  - msg.payload → SQL statement
- Output
  - msg.payload → SQL Result set
    - Row mode - 1 payload per record (Default)
    - Array mode - 1 message per result-set in an array
- Error handling can be poor or non-existent!

Install DB2 for i Adapter

- Due to problems with IBMs DB2 for i adapter, sometimes it must be installed manually!
- From a shell session, in the ~/.node-red directory
  - Install the pre-req's
    - npm i idb-connector -g
  - DB2 node
    - npm i node-red-contrib-db2-for-i -g
  - To confirm
    - npm list -g

Install DB2 for i Adapter

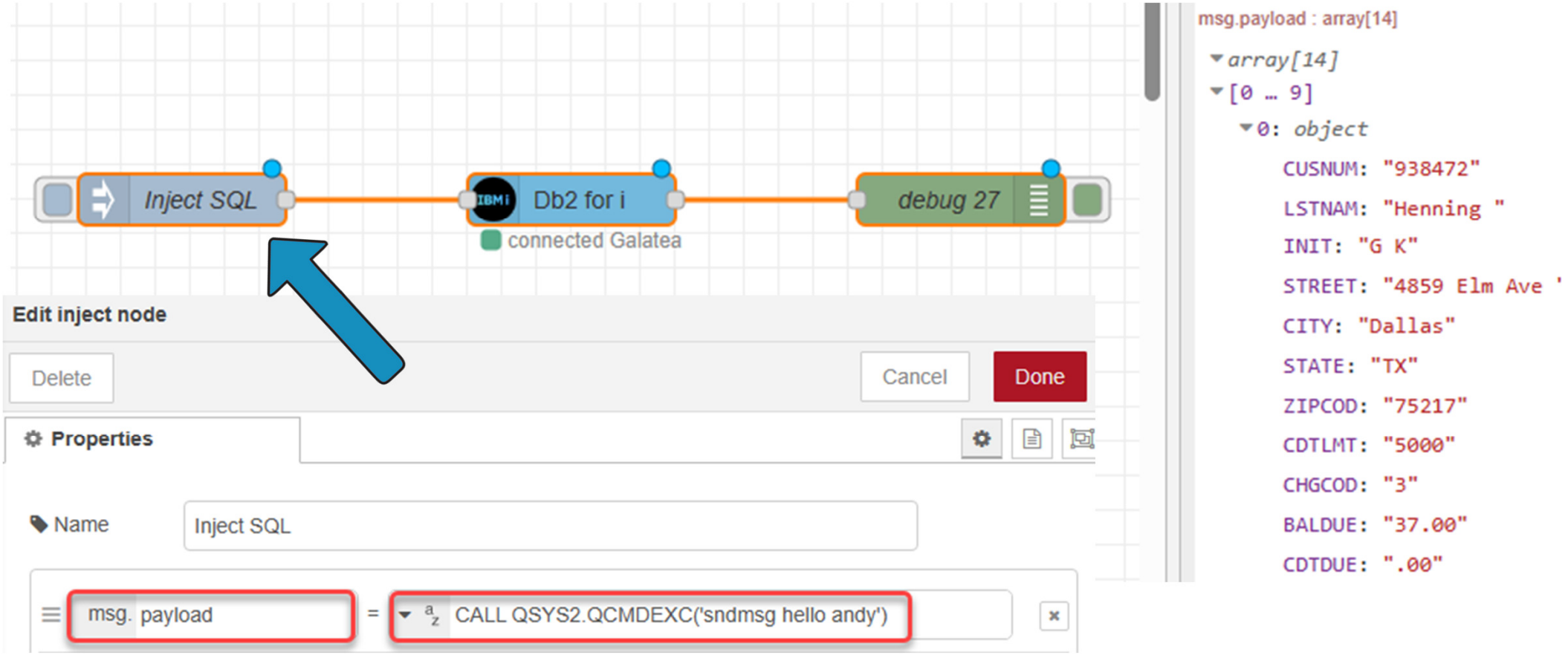
SSH Galatea | SSH Galatea

```
andy@GALATEA:~$ cd .node-red
andy@GALATEA:.node-red$ npm i -g idb-connector

added 60 packages in 8s

4 packages are looking for funding
  run `npm fund` for details
andy@GALATEA:.node-red$ npm i -g node-red-contrib-db2-for-i

added 61 packages in 6s

4 packages are looking for funding
  run `npm fund` for details
andy@GALATEA:.node-red$ npm list -g
/QOpenSys/pkgs/lib/nodejs20/lib
+-- corepack@0.17.2
+-- idb-connector@1.2.18
+-- node-red-contrib-db2-for-i@0.2.3
`-- npm@9.7.1

andy@GALATEA:.node-red$
```

Demo & Exercise

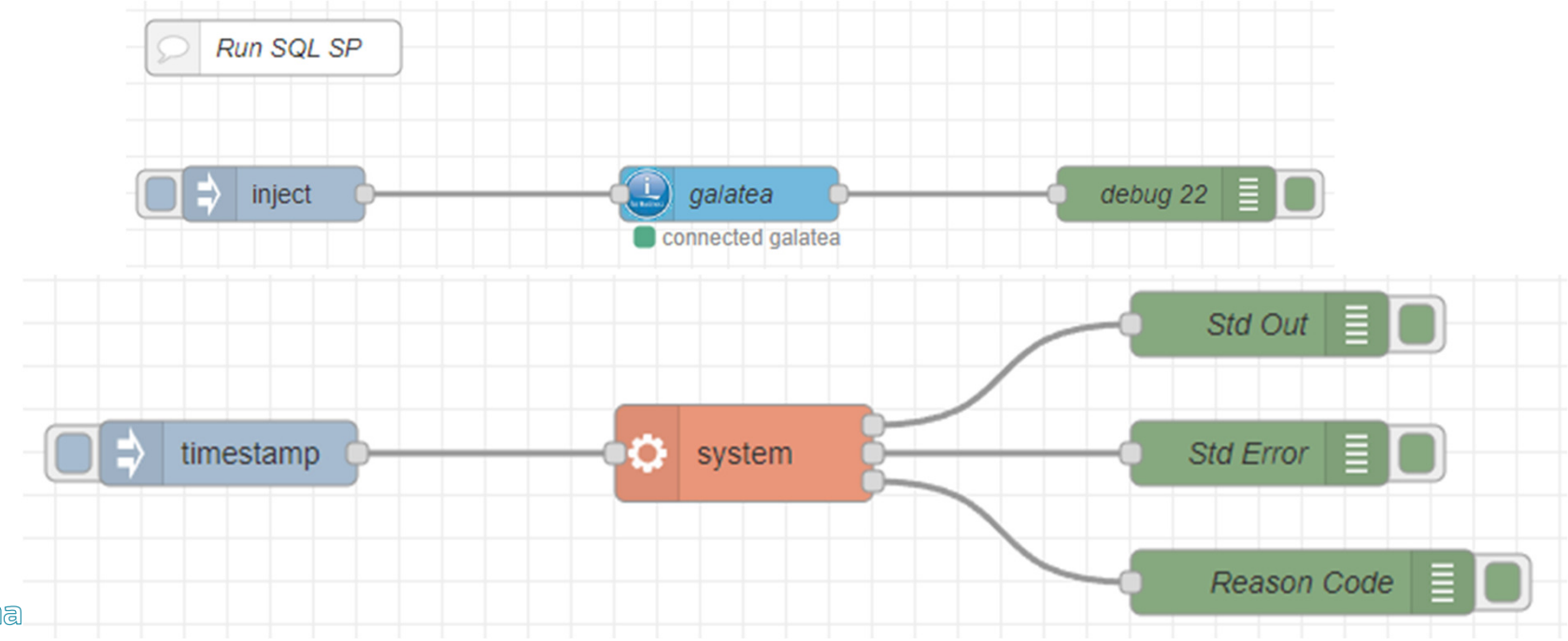
- Install the DB2 for i package, if you have not done so already
- Create a flow, that will output to the debug window, all records in the QCUSTCDT table in schema QIWS
- The DB2 for i node should return results in a single array

Demo & Exercise



Node-RED IBM i Bumph

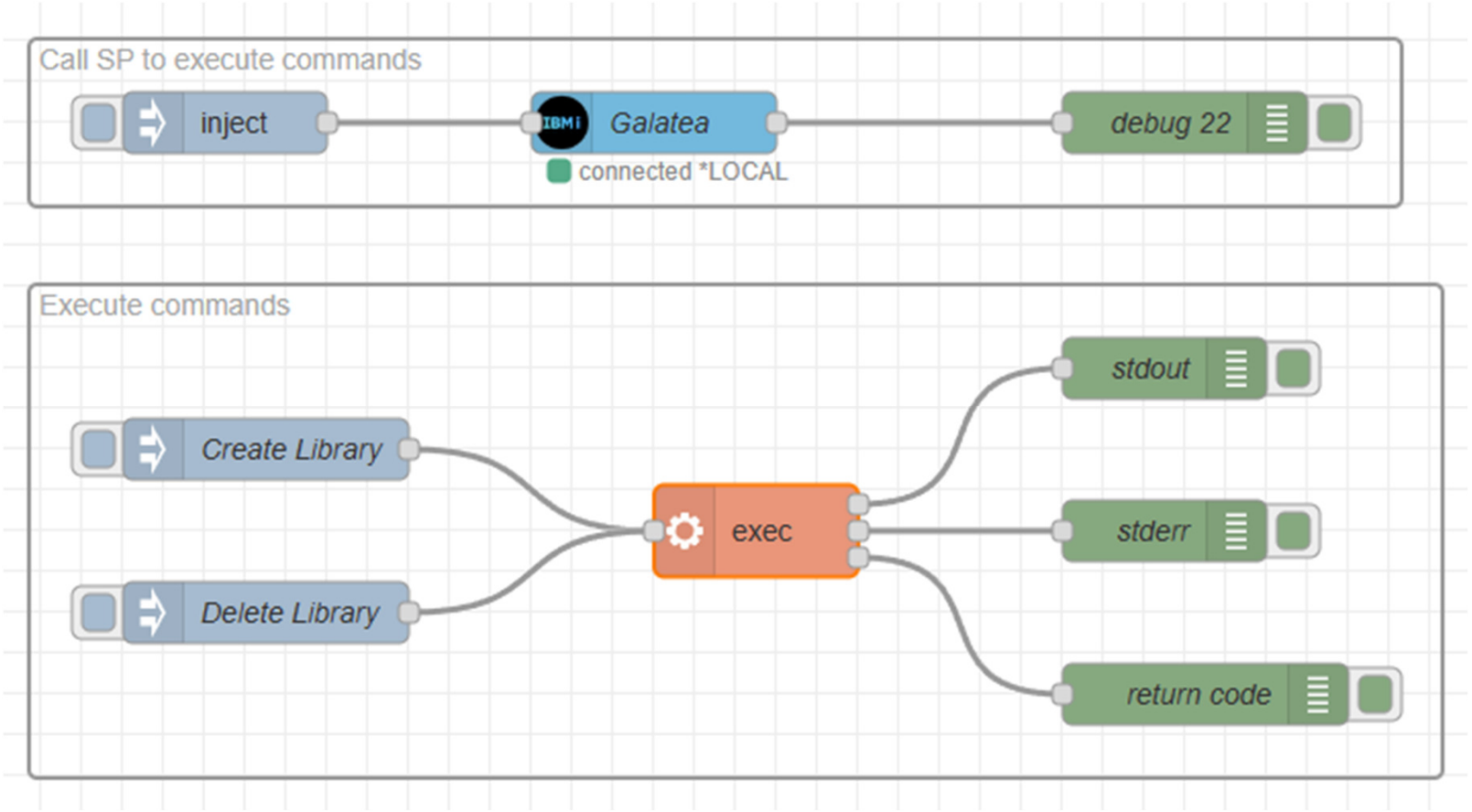
- Calling a program or running command
  - Use SQL Stored Procedures
  - Use Exec node



Demo & Exercise

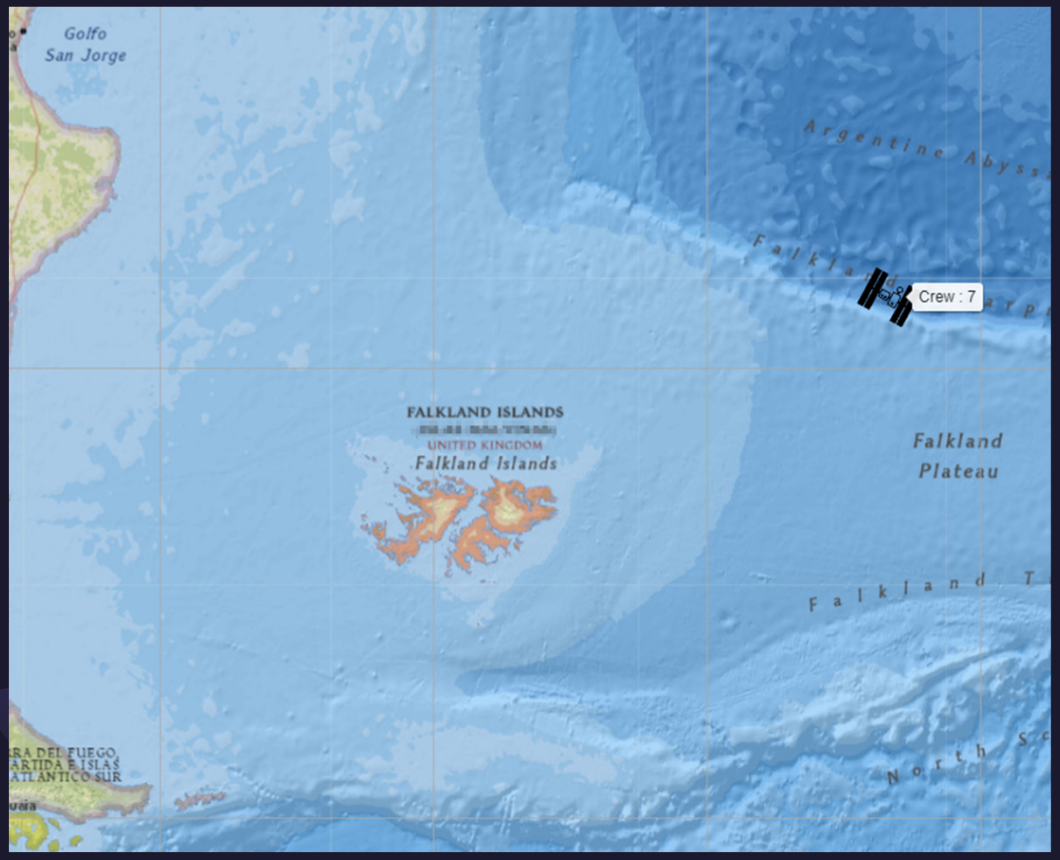
- Create a flow that will
  - Create a library on your server
  - Delete the same library on your server
  - Send a message to QSYSOPR

Demo & Exercise

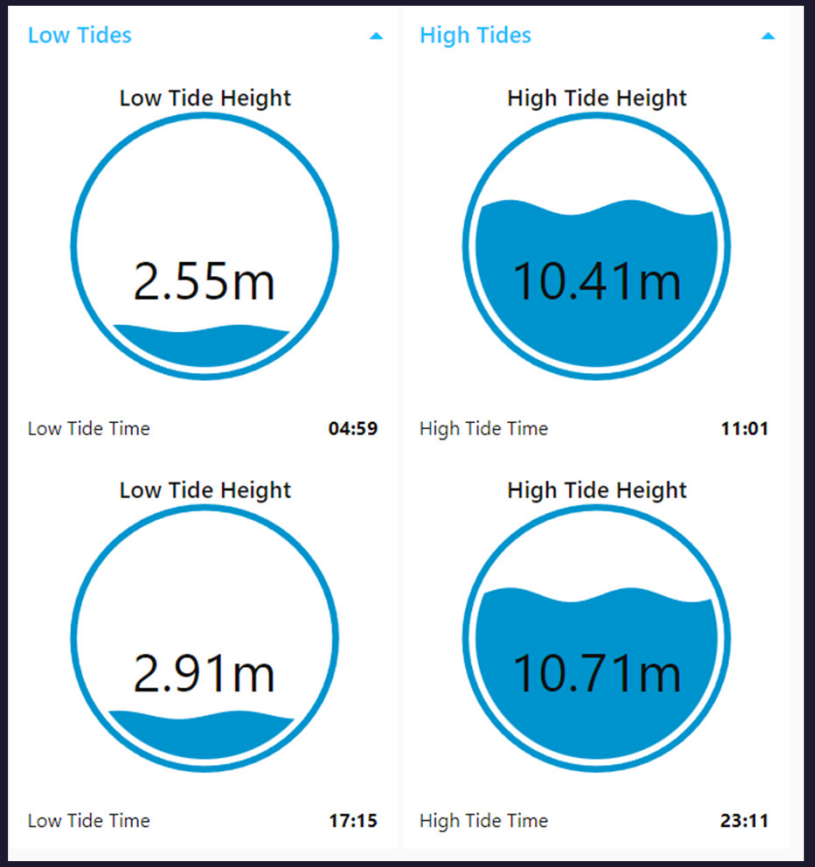


A Couple of Extras

- International Space Station



- High & Low Tide Times



Thank You!

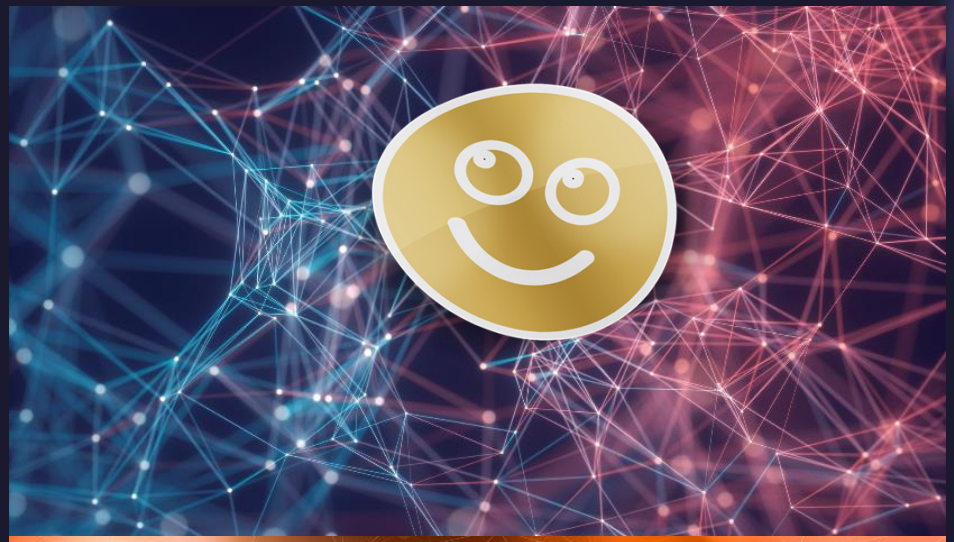
Andy Youens

Andy@FormaServe.co.uk

www.formaserve.co.uk

learning.formserve.co.uk

01908 109978



Legal Bumph!

- © Copyright FormaServe Systems 1990 - 2023
- All rights reserved
- All trademarks, trade names, service marks & logos referenced herein belong to their respective companies
- No unauthorised use, copying or distribution permitted

- This presentation is for informational purposes only
- FormaServe assumes no responsibility for the accuracy or completeness of the information within