HITACHI
Inspire the Next

Node-RED Hands-on with Open Technologies

October 22, 2024

Kazuhito Yokoi
Hitachi Academy

Self-introduction

HITACHI
Inspire the Next

Kazuhito Yokoi
(横井 一仁)

- Software Engineer, Hitachi Academy
- No.3 contributor in Node-RED project
- Organizer of Node-RED User Group

OpenJS Foundation

Node-RED Unleashed: Transforming Industrial IoT Development and Industry Collaboration with Hitachi

Posted On: March 27, 2024

Hear from Hitachi's Kazuhito Yokoi to find out more about Node-RED and why it is becoming so popular in Industrial IoT applications.

My interview on
OpenJS Foundation blog

https://openjsf.org/blog/node-red-unleashed

HITACHI
Inspire the Next

Contents

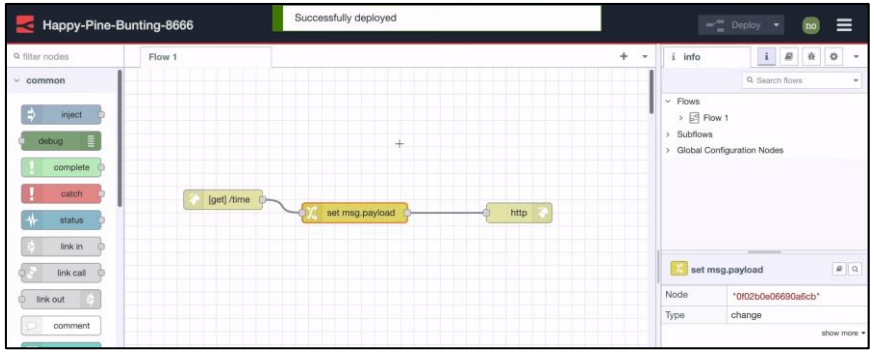
What is Node-RED?

What is Node-RED?

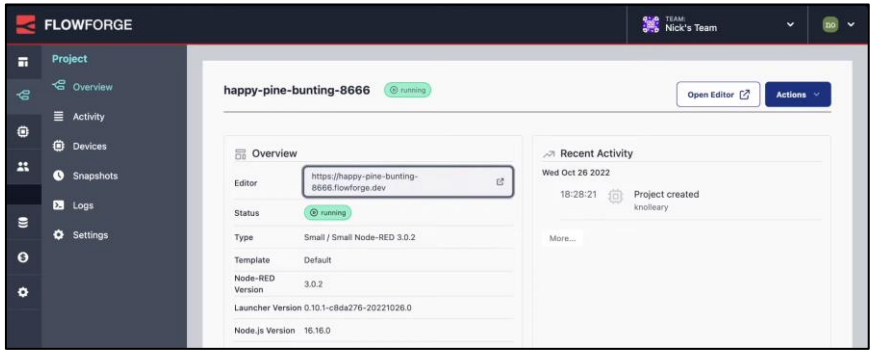
HITACHI
Inspire the Next

Visual programming tool for IoT and web applications

- Developed by IBM in 2013
- OSS hosted in OpenJS Foundation
- Browser based development environment
- Management OSS available



Node-RED flow editor



Management software

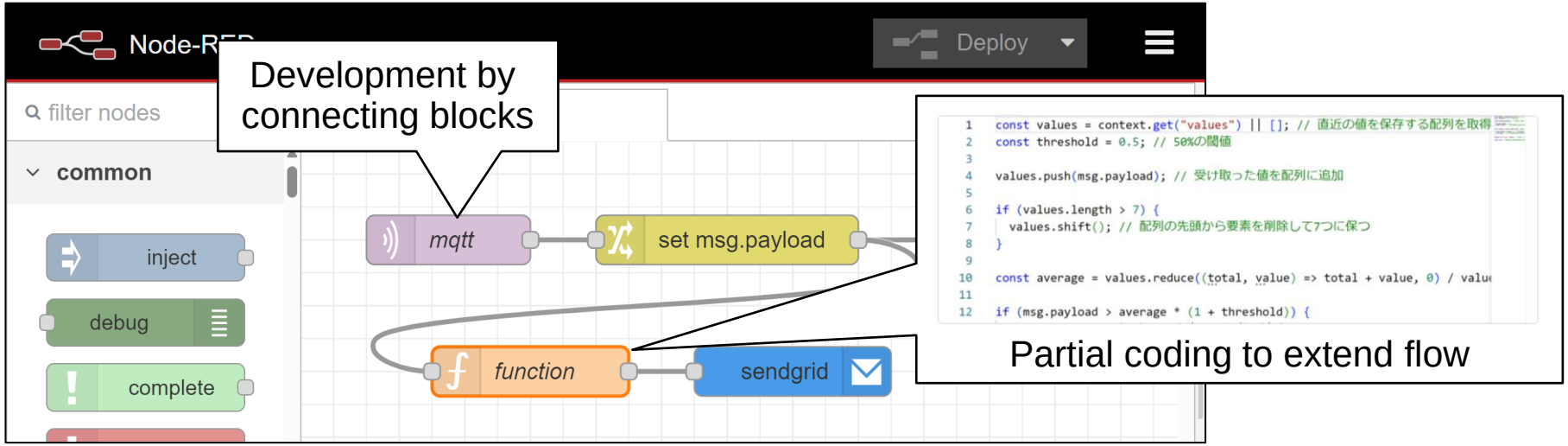
https://flowfuse.com/

Node-RED Flow Editor

HITACHI
Inspire the Next

Visual programming environment by connecting functional blocks in the processing order

- No code development environment for beginners
- Extensibility through partial code writing



Node-RED Nodes

HITACHI
Inspire the Next

Nodes are functional blocks that connect to services and process data

(1) Data collection

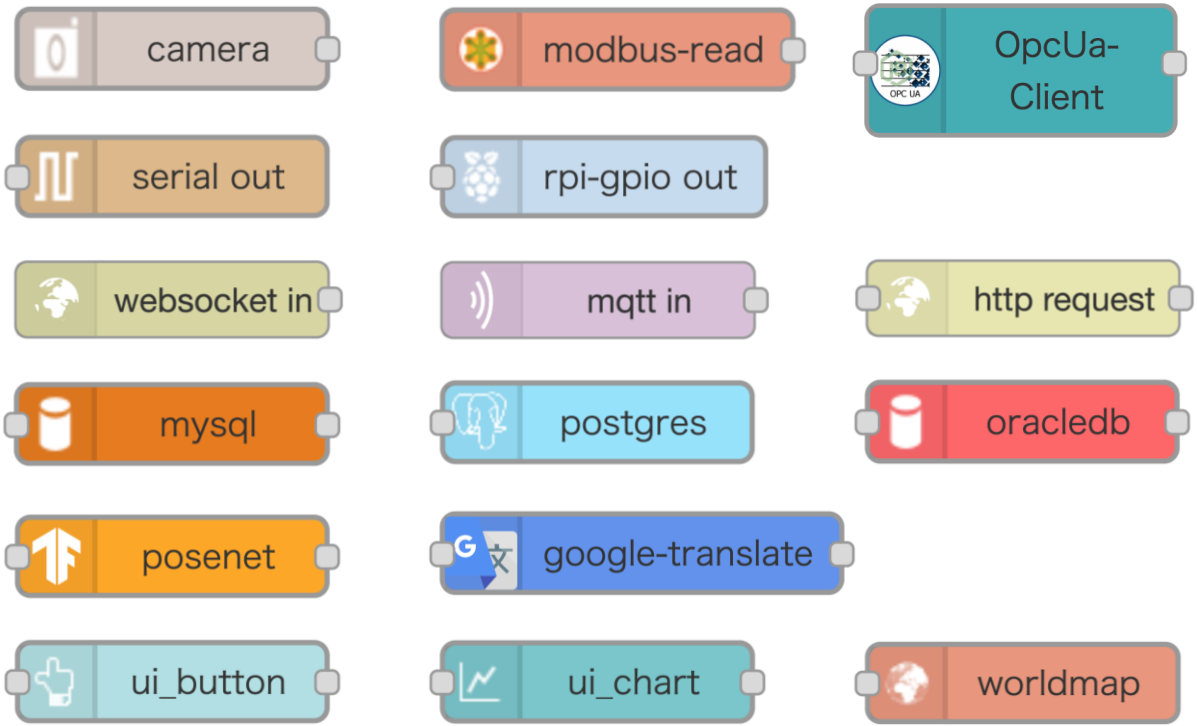
(2) Device control

(3) Connection with external systems

(4) Databases

(5) Data analysis

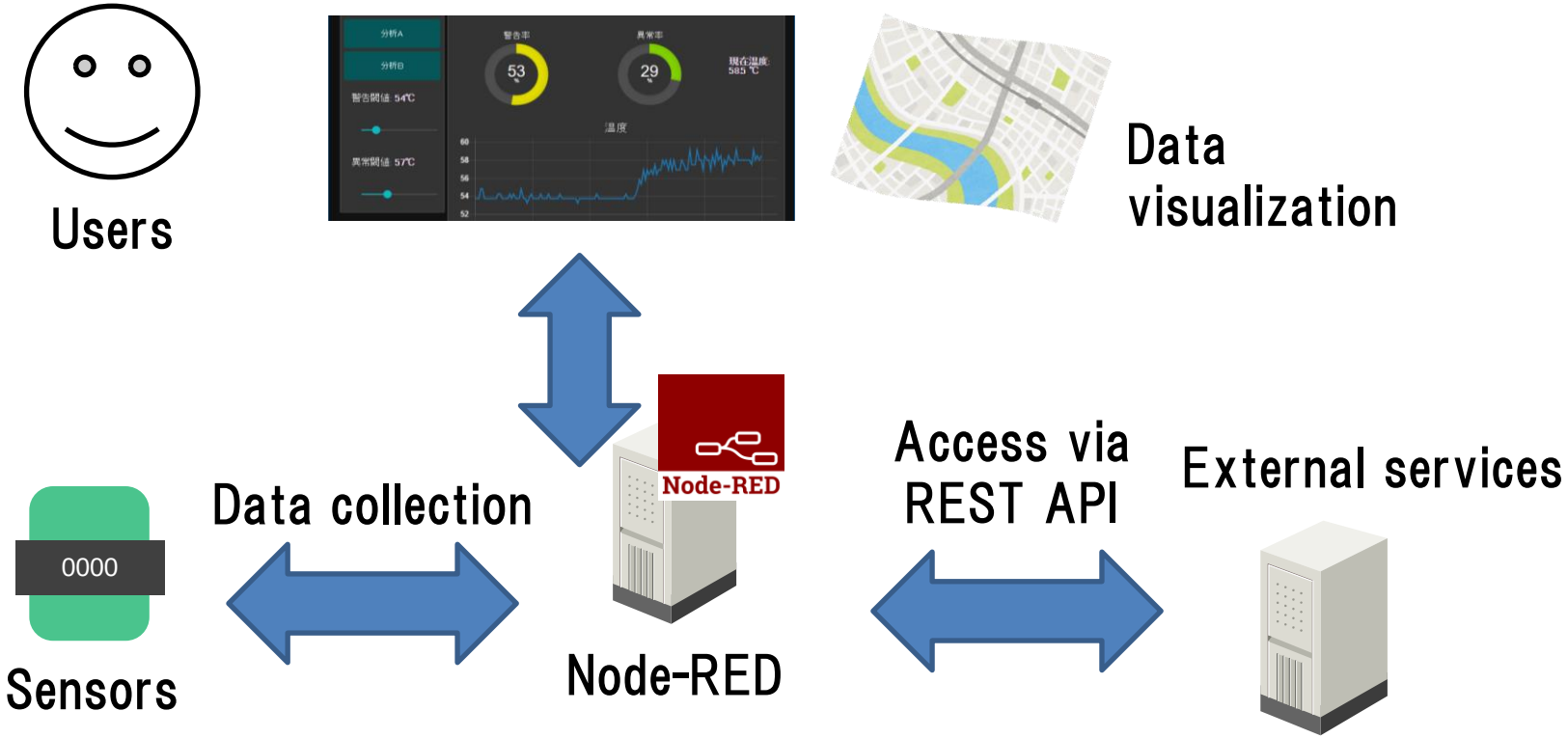
(6) Data visualization



Node-RED as the Hub Tool to Integrate Services

HITACHI
Inspire the Next

Node-RED is suitable for building applications by integrating many components.



Preparing Node-RED Environment

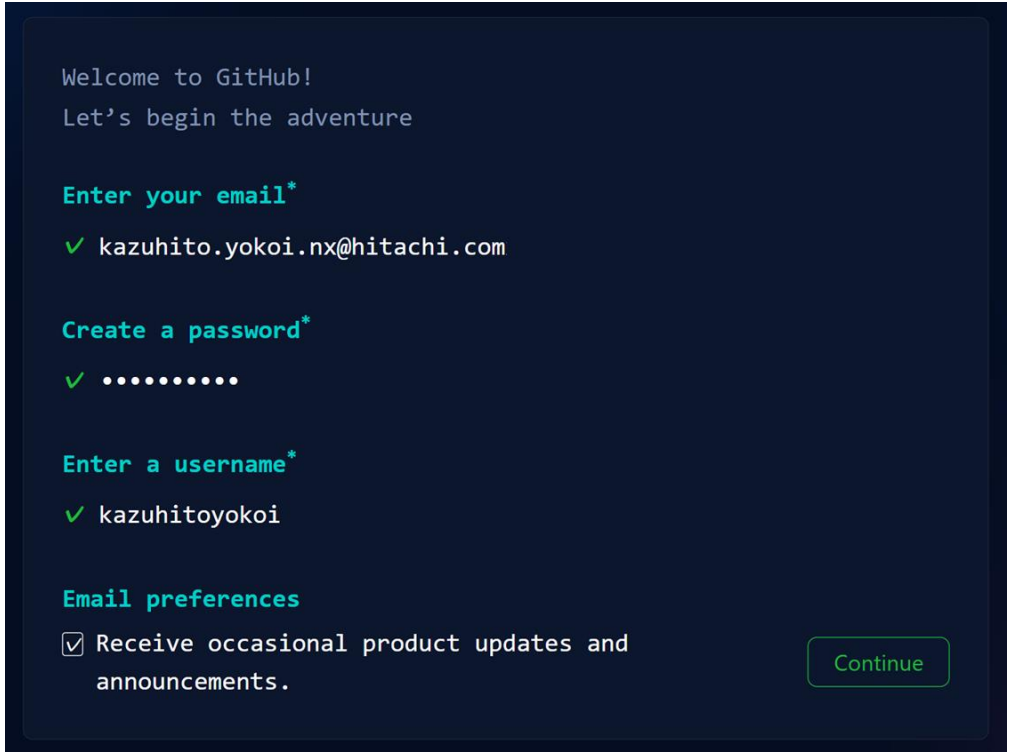
[Hands-on] Singing up for a GitHub Account

If you have no GitHub account, create your GitHub account from the following URL.

https://github.com/signup

To register for the account, you need to input your information.

- e-mail address
- Password
- Username



Registration for creating GitHub account

[Hands-on] Longer Timeout of GitHub Codespaces

HITACHI
Inspire the Next

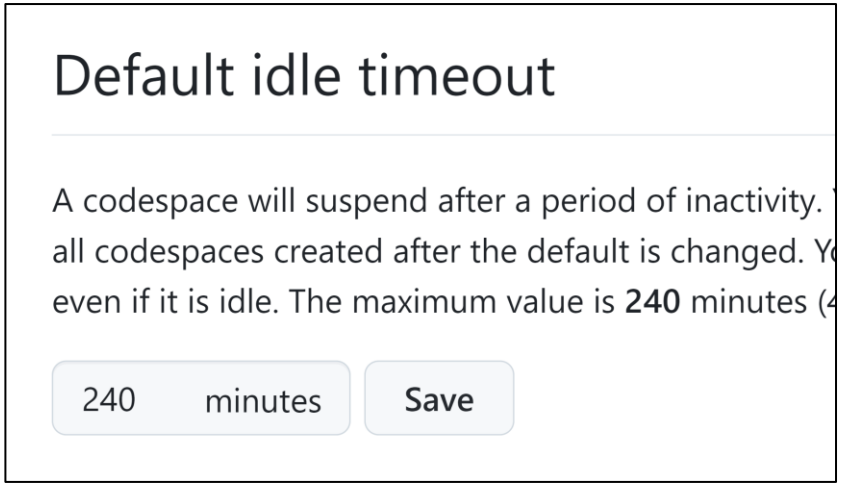
Because the default timeout of 30 minutes is too short,
increase the timeout to the maximum on the settings page.

(1) Access the Codespaces settings from the following URL.

https://github.com/settings/codespaces

(2) Set 240 minutes (4 hours)

(3) Click the "Save" button



[Hands-on] Starting GitHub Codespaces Template

HITACHI
Inspire the Next

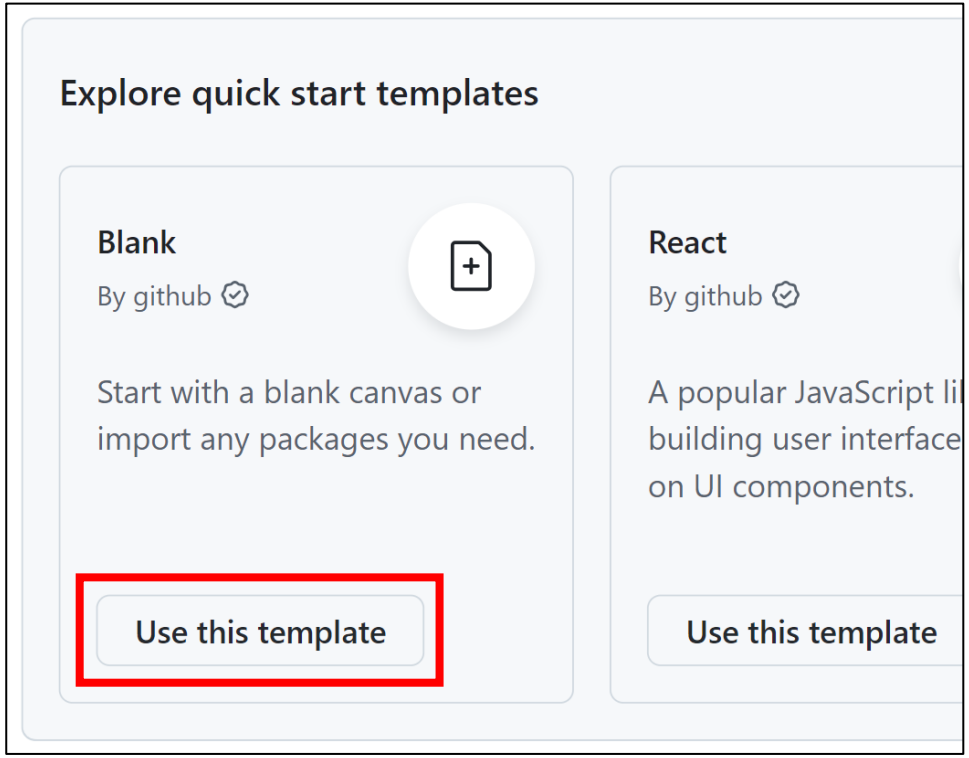
Creating the empty environment on the GitHub Codespaces

(1) Go to the Codespaces page

https://github.com/codespaces

(2) Click the "Use this template" button in the Blank item

-> Visual Studio Code-like editor will be opened.



[Hands-on] Node-RED Installation

HITACHI
Inspire the Next

After opening the code editor, you can use Terminal in the bottom right corner.

(1) Install Node-RED with the npm command

```
npm install -g node-red
```

```
PROBLEMS    OUTPUT    TERMINAL    ···                bash
@kazuhitoyokoi ➜ /workspaces/codespaces-blank $ npm install -g node-red
```

(2) Start Node-RED

```
node-red
```

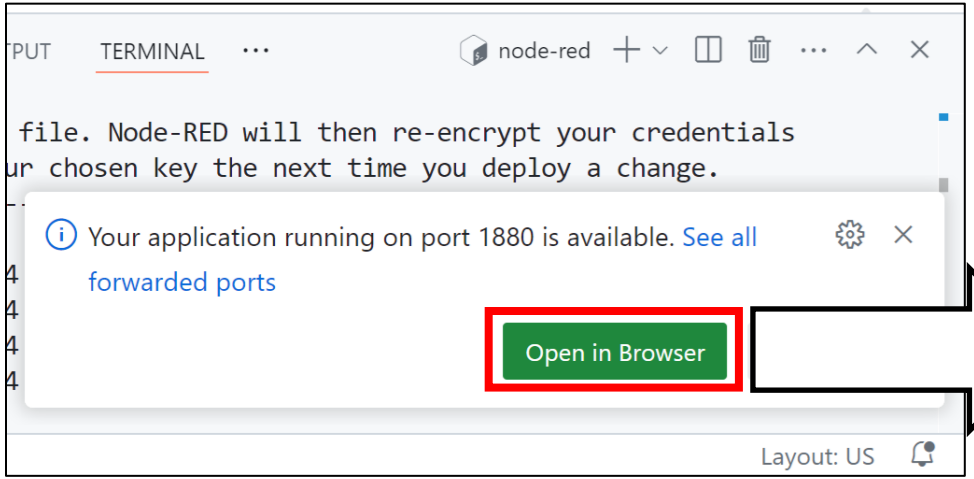
```
npm notice To update run: npm install -g npm@10.9.0
npm notice
@kazuhitoyokoi ➜ /workspaces/codespaces-blank $ node-red
0                                                    Layout: US
```

[Hands-on] Opening Node-RED Flow Editor

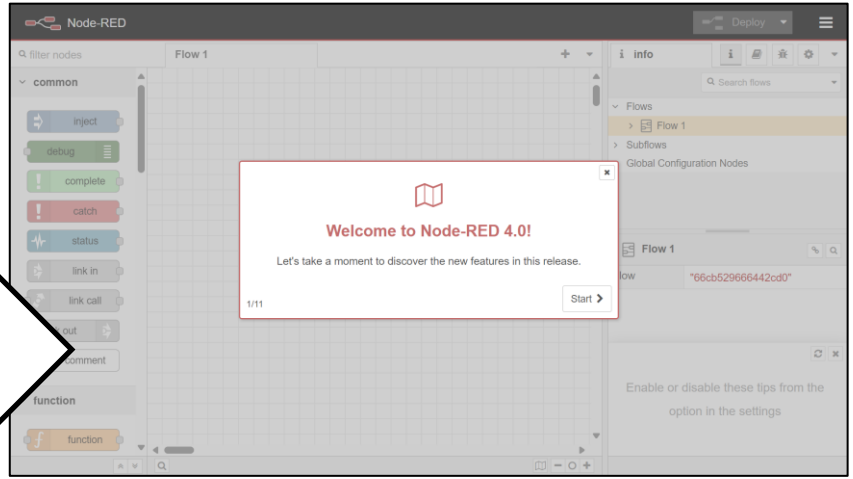
HITACHI
Inspire the Next

- After starting Node-RED, the pop-up dialog will appear to open flow editor.
- Click the green "Open in Browser" button

  -> You can see the Node-RED flow editor in new tab.



Notification dialog

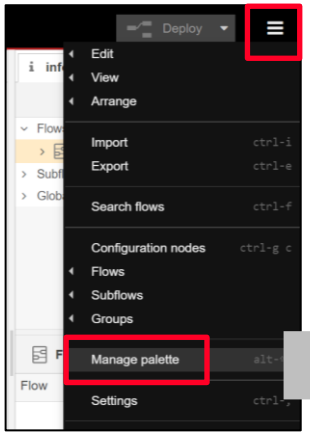


Node-RED flow editor

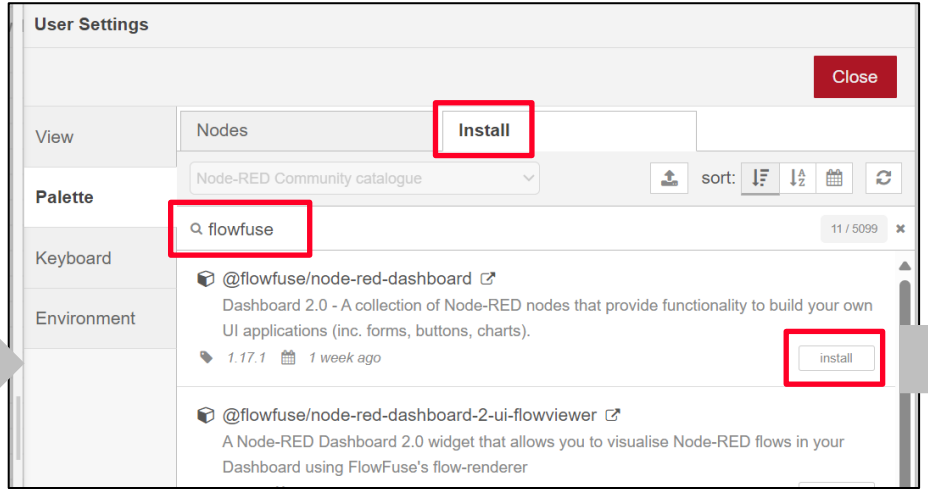
[Hands-on] Required Nodes Installation

HITACHI
Inspire the Next

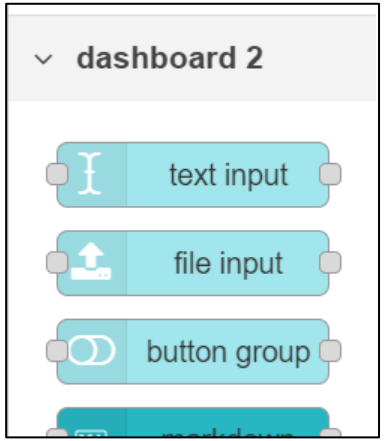
(1) Click the Hemberger icon on the top right to open the menu
(2) Select "Manage palette" in the menu to open the user settings
(3) Select the "Install" tab in the Palette tab
(4) Type "flowfuse" in the search box
(5) Click the "install" button in the "@flowfuse/node-red-dashboard" to install nodes



Select "Manage palette"



Search and add node



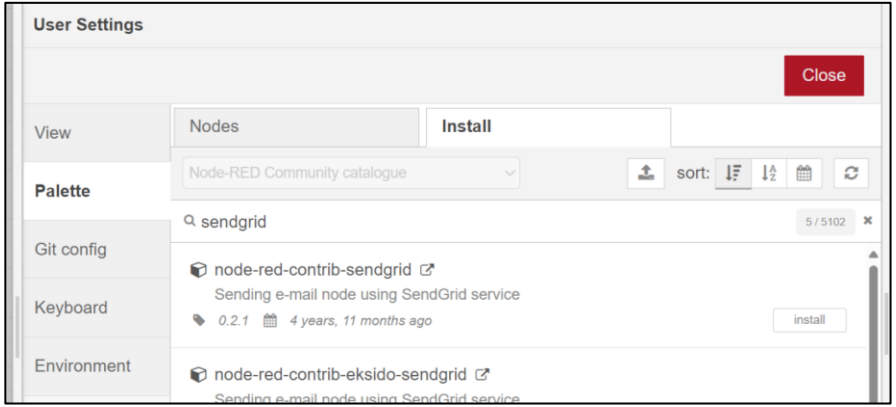
Palette after installation

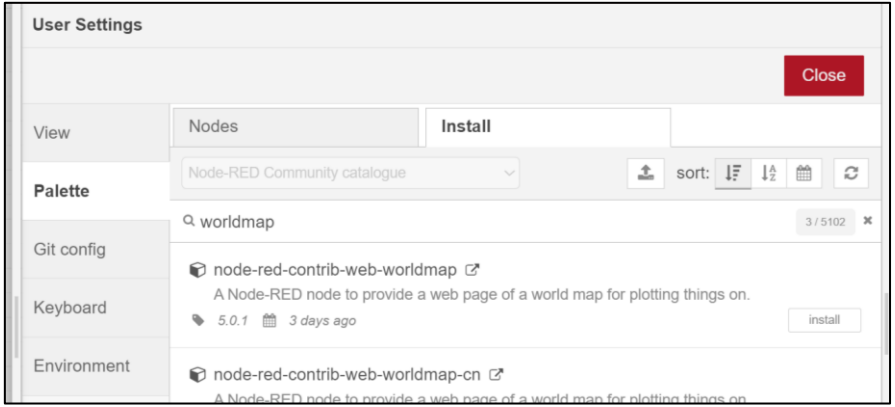
[Hands-on] Other Required Nodes Installation

HITACHI
Inspire the Next

Install two more red nodes in the following table

#	Search keyword	Module name
1	flowfuse	@flowfuse/node-red-dashboard
2	sendgrid	node-red-contrib-sendgrid
3	worldmap	node-red-contrib-web-worldmap



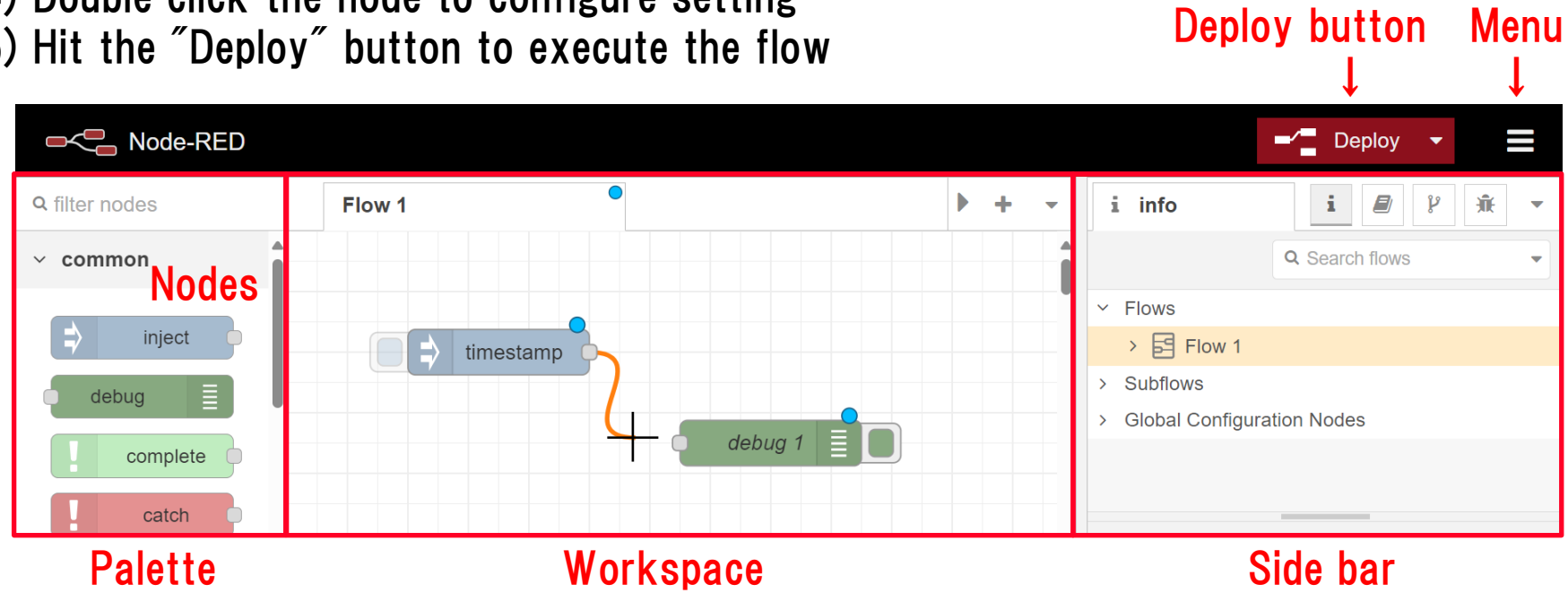


Basic Operations

Typical Development on Node-RED Flow Editor

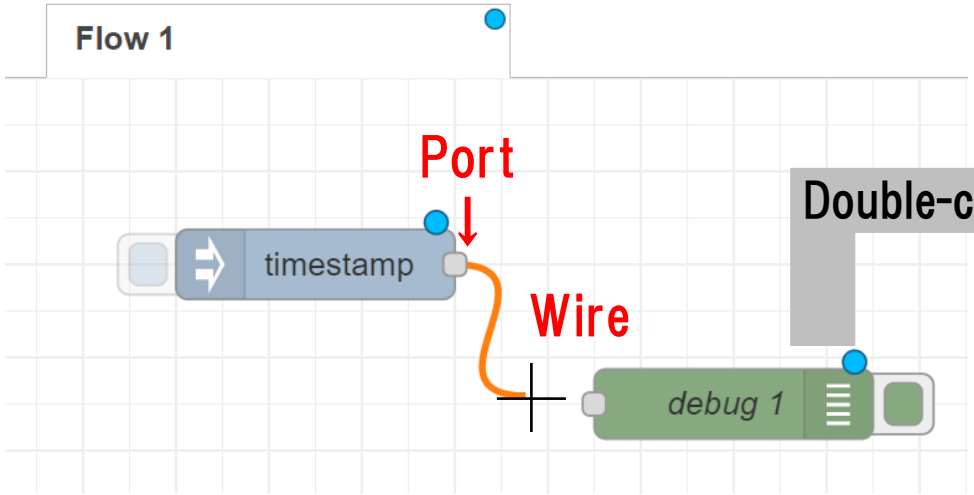
HITACHI
Inspire the Next

(1) Select the node which you want to use from the left palette
(2) Drag and drop the node to the workspace
(3) Connect nodes with wire in the processing order
(4) Double-click the node to configure setting
(5) Hit the "Deploy" button to execute the flow

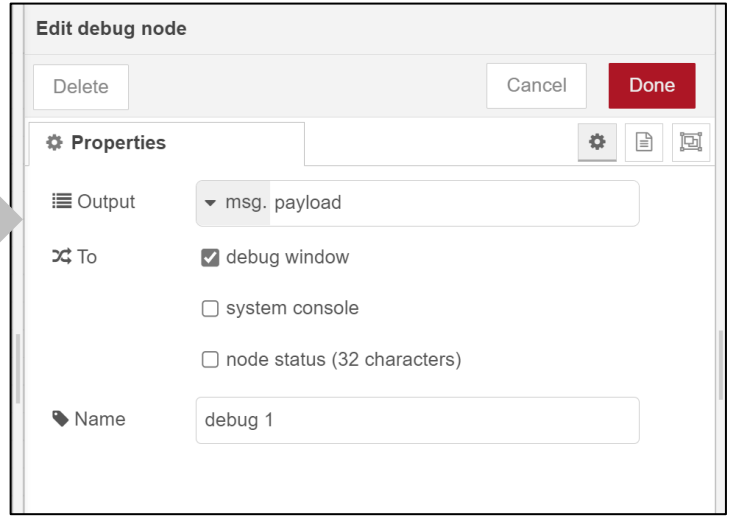


Operations in the Workspace

- To connect nodes, wire the output port to the input port of another node.
- To configure node properties, double-click the node to open the settings UI (after entering the properties, click the "Done" button to apply them).



Connecting nodes by wire



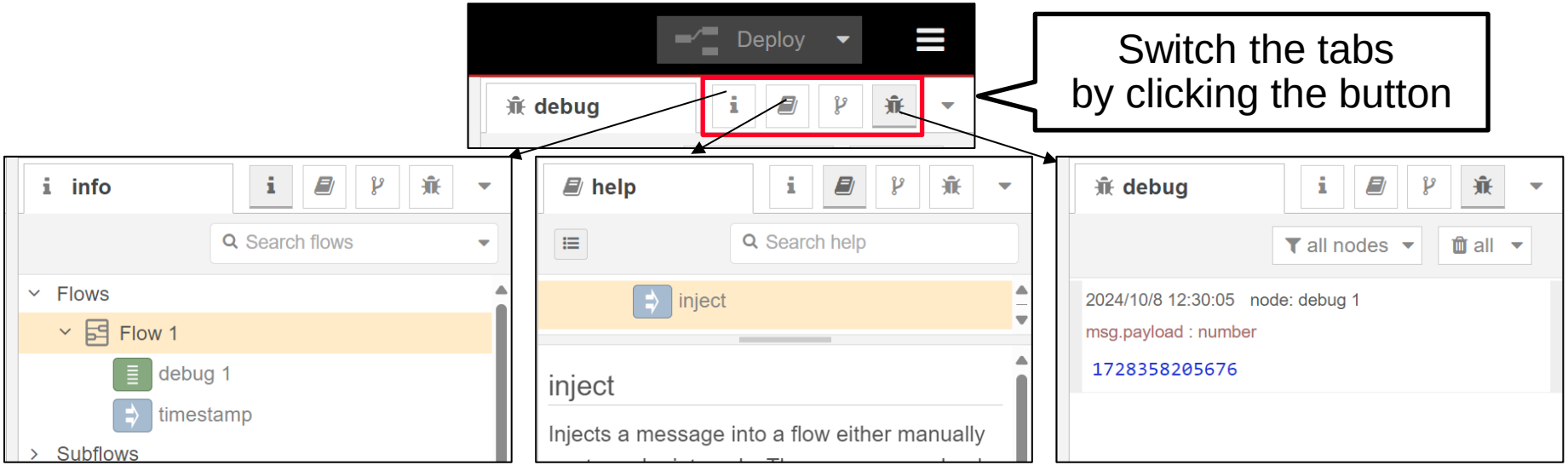
Node property settings UI

Sidebar (1/2)

HITACHI
Inspire the Next

The area that displays the information about the flow, nodes, and messages

- Information tab: Tab that provides an overview of nodes in the workspace
- Help tab: Tab for node details about how to use the selected node
- Debug tab: Tab for viewing the data output from the debug nodes



Information tab

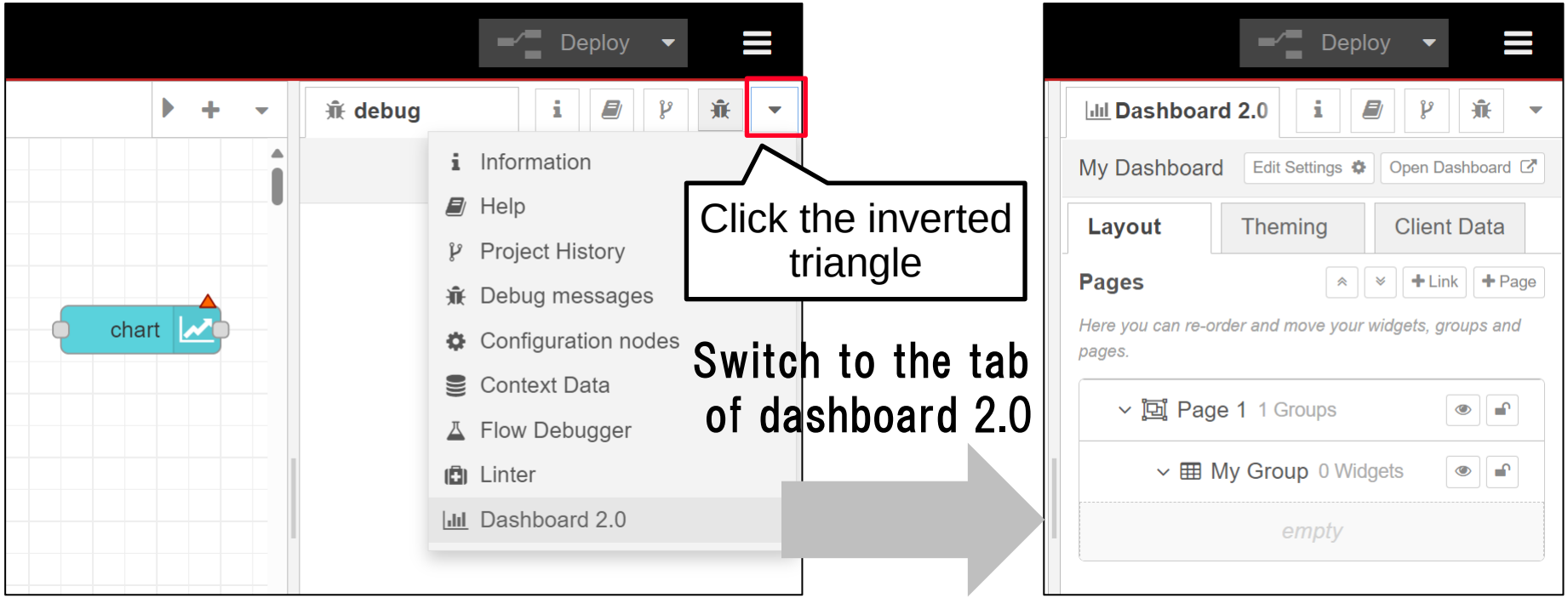
Help tab

Debug tab

Sidebar (2/2)

HITACHI
Inspire the Next

When there is no button on the sidebar, you can display the menu by clicking the inverted triangle button in the right corner.

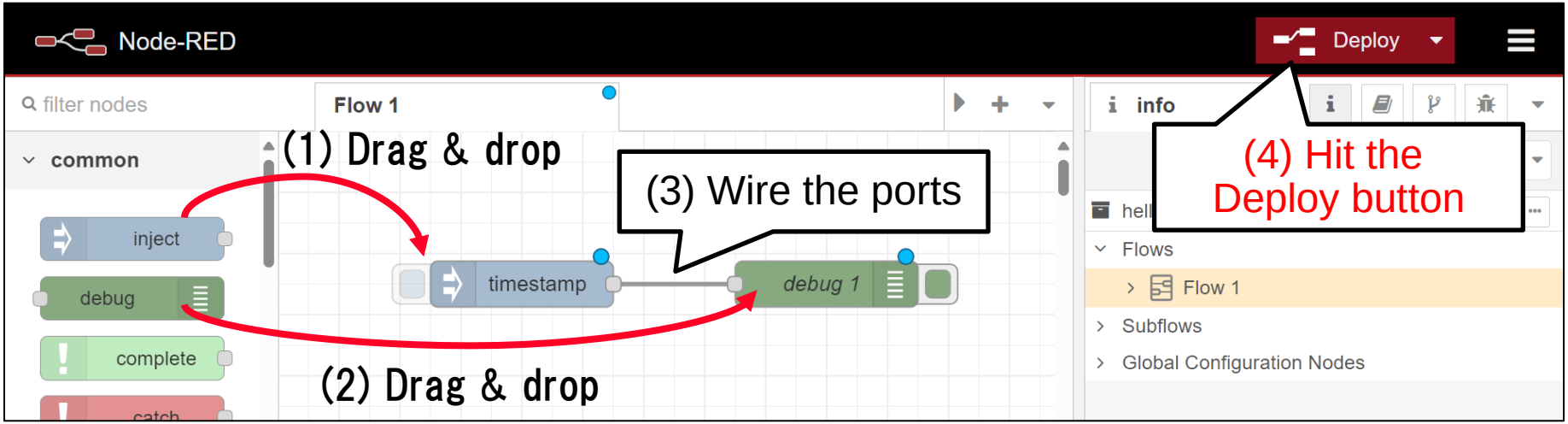


[Hands-on] Creating the First Flow (1/3)

HITACHI
Inspire the Next

Create the flow to output the current timestamp by the following steps

(1) Drag and drop the inject node in the common category to the workspace
(2) Place the debug node to the right side of the inject node
(3) Connect the inject node and the debug node by wiring between the ports
(4) Hit the "Deploy" button to execute the flow



[Hands-on] Creating the First Flow (2/3)

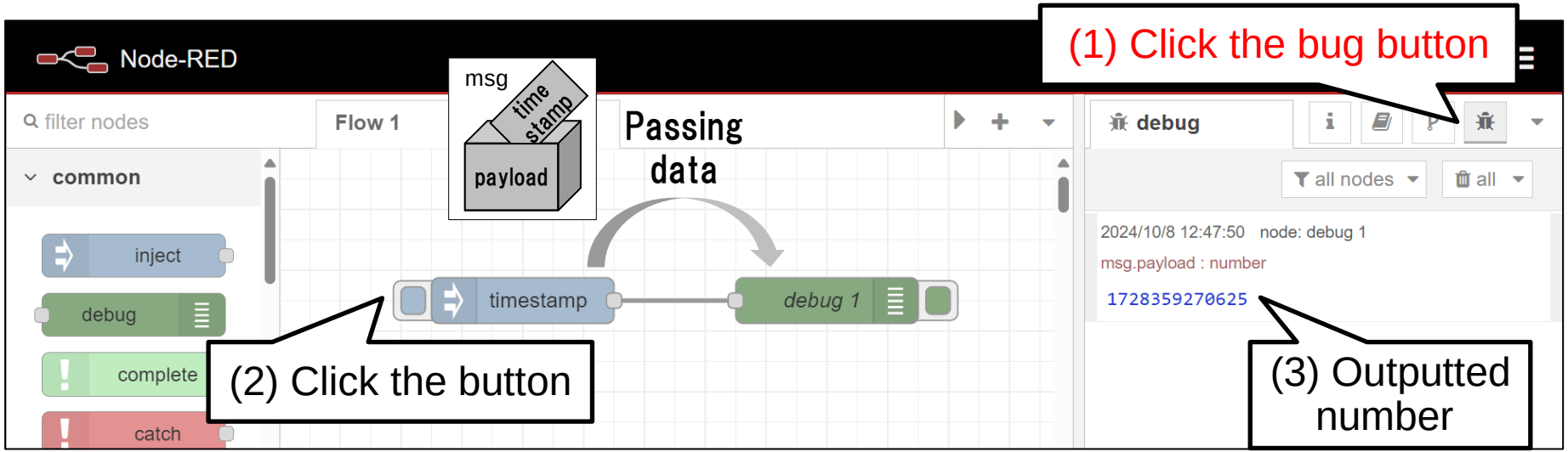
HITACHI
Inspire the Next

Execute the developed flow by the following steps

(1) Click the bug button in the sidebar to open the debug tab

(2) Click the button of the inject node to execute the flow

(3) The number representing the current time will be output to the debug tab

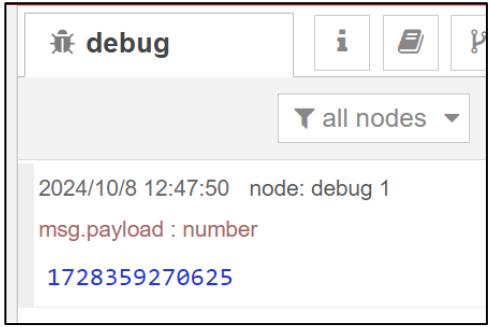


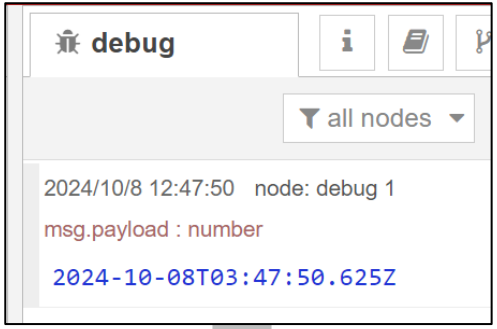
[Hands-on] Creating the First Flow (3/3)

HITACHI
Inspire the Next

The outputted data in the debug tab can be switch to another format by clicking

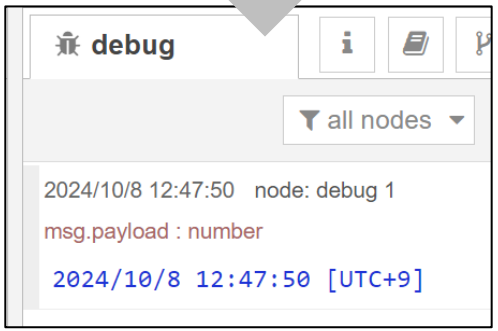
Number

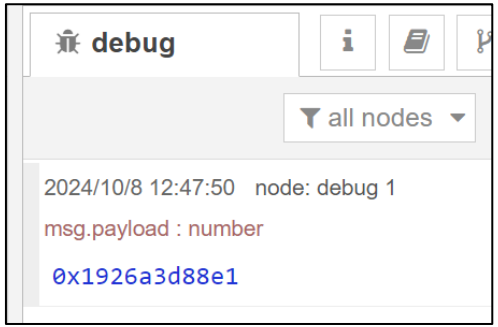




Time in UTC

Local time





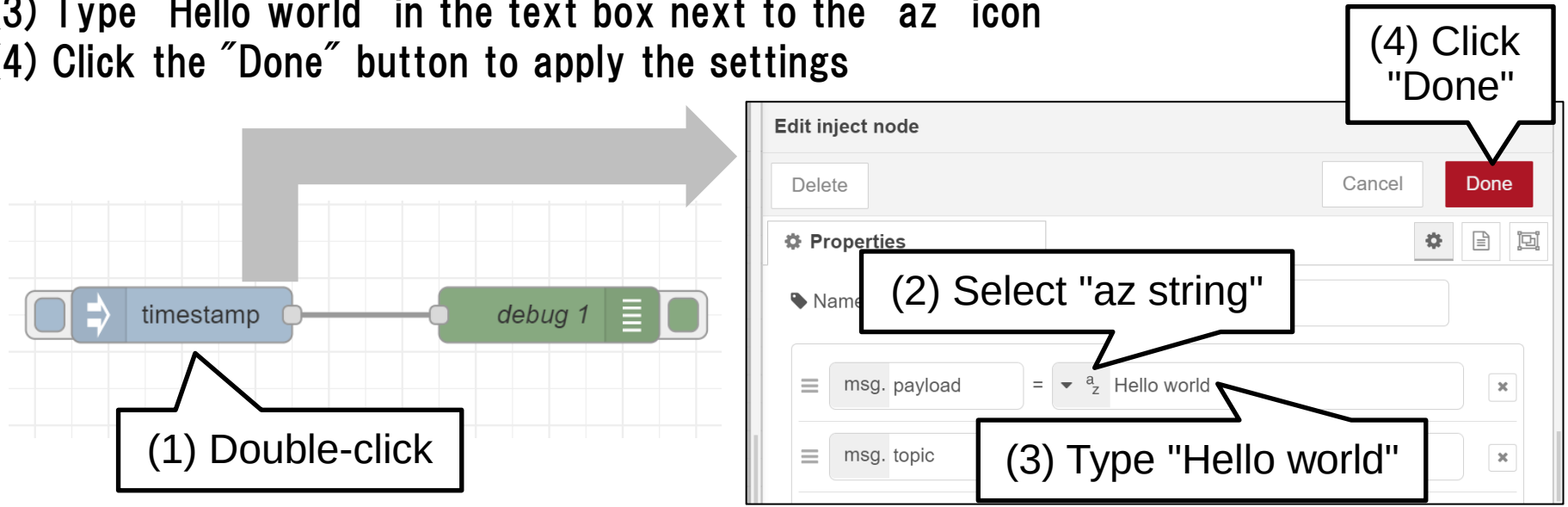
Hexadecimal number

[Hands-on] Creating Flow to Output Hello World (1/2)

HITACHI
Inspire the Next

To develop the flow to output "Hello world", change the output from the current time to the string, "Hello world", in the node property setting of the inject node.

(1) Double-click the inject node to open the property settings UI
(2) Change the data type from "milliseconds since epoch" to "string"
(3) Type "Hello world" in the text box next to the "az" icon
(4) Click the "Done" button to apply the settings



[Hands-on] Creating Flow to Output Hello World (2/2)

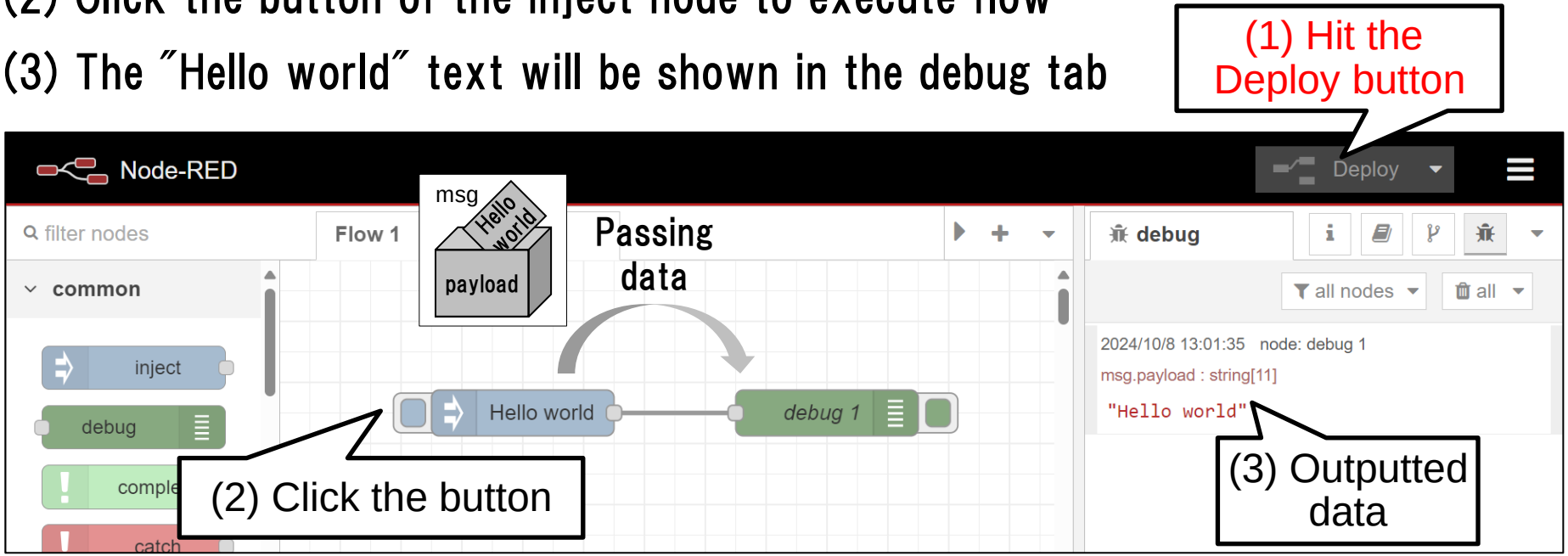
HITACHI
Inspire the Next

Executing the developed flow and checking the outputted "Hello world" text

(1) Hit the Deploy button to apply the flow

(2) Click the button of the inject node to execute flow

(3) The "Hello world" text will be shown in the debug tab

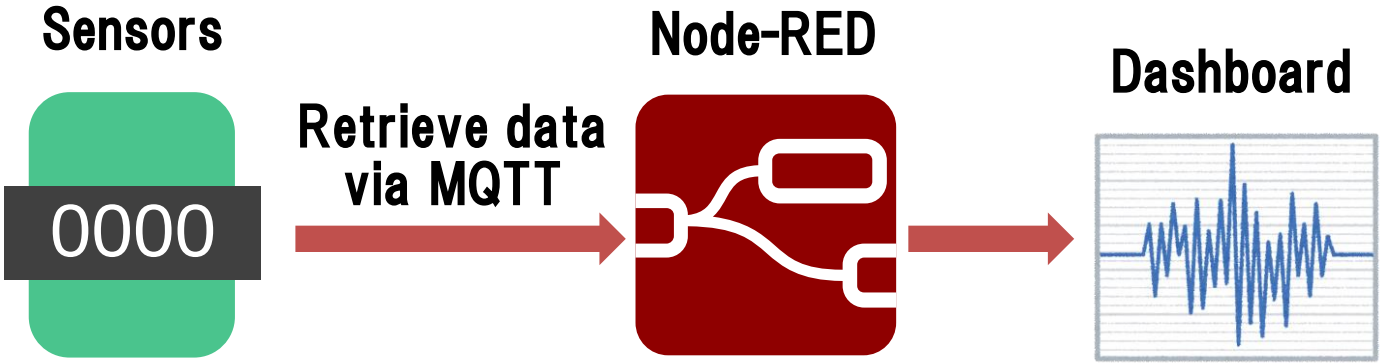


Data Visualization

Data Visualization on the Dashboard

HITACHI
Inspire the Next

Creating a flow to retrieve sensor data via MQTT protocol and then visualize the data in the time series graph on the dashboard



Node-RED has the advantage of natively supporting MQTT, the lightweight data transfer protocol invented by IBM.

[Hands-on] Placing the mqtt-in Node to Retrieve Sensor Data

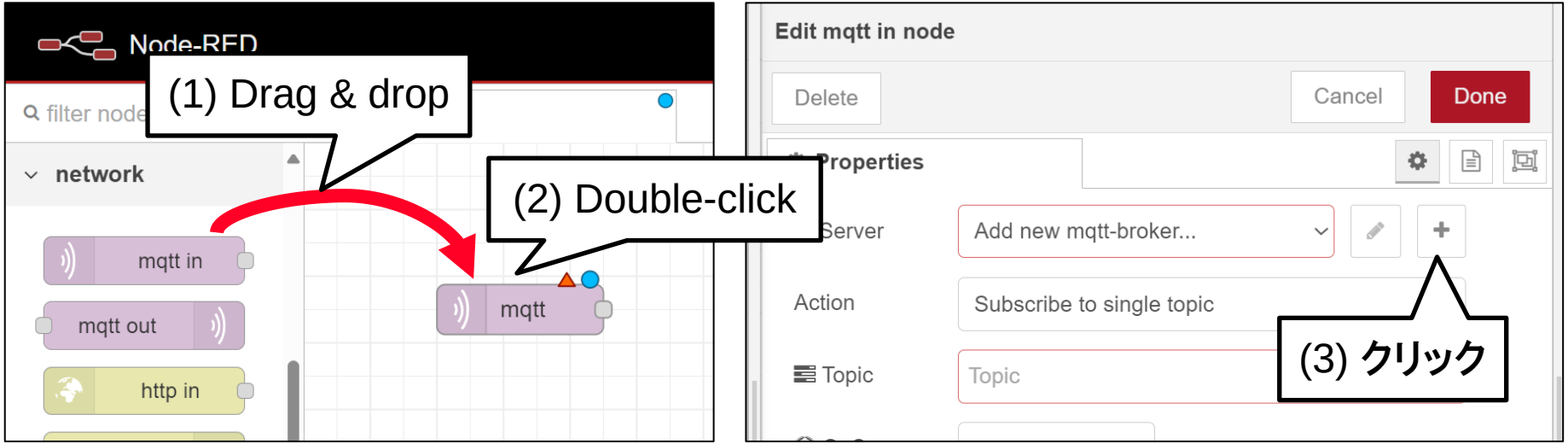
HITACHI
Inspire the Next

Placing the mqtt-in node to retrieve sensor data from the MQTT broker

(1) Drag and drop the mqtt-in node from the network category of palette to workspace

(2) Double-click the mqtt-in to open the node property UI

(3) Click the "+" button next to "Add new mqtt-broker" pull-down menu

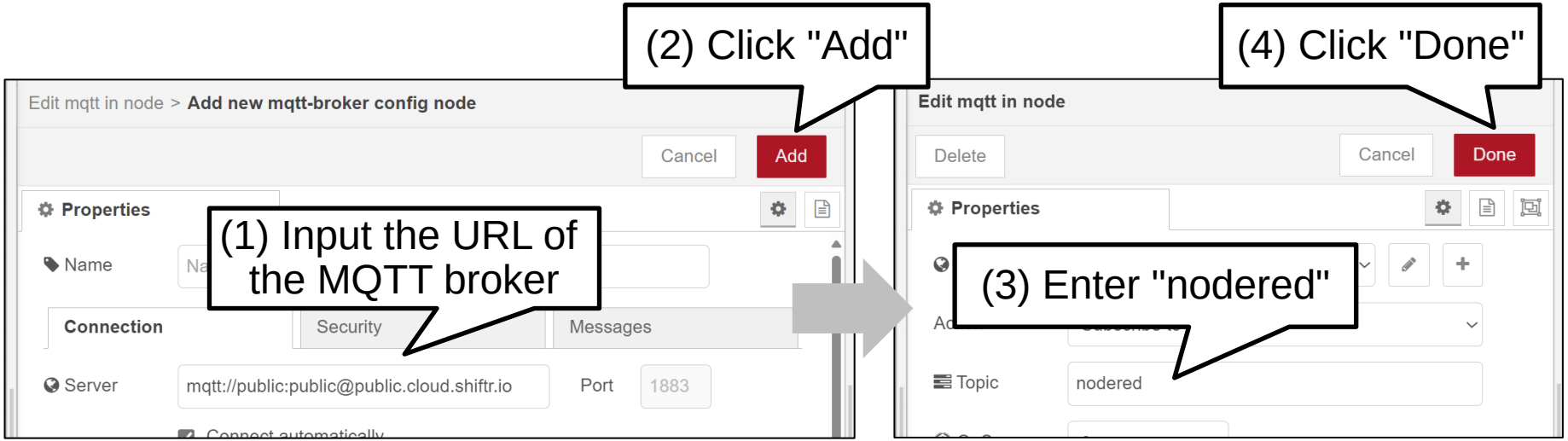


[Hands-on] Configuring mqtt-in Node Property

HITACHI
Inspire the Next

(1) Input the following URL of the public MQTT broker
"mqtt://public:public@public.cloud.shiftr.io"

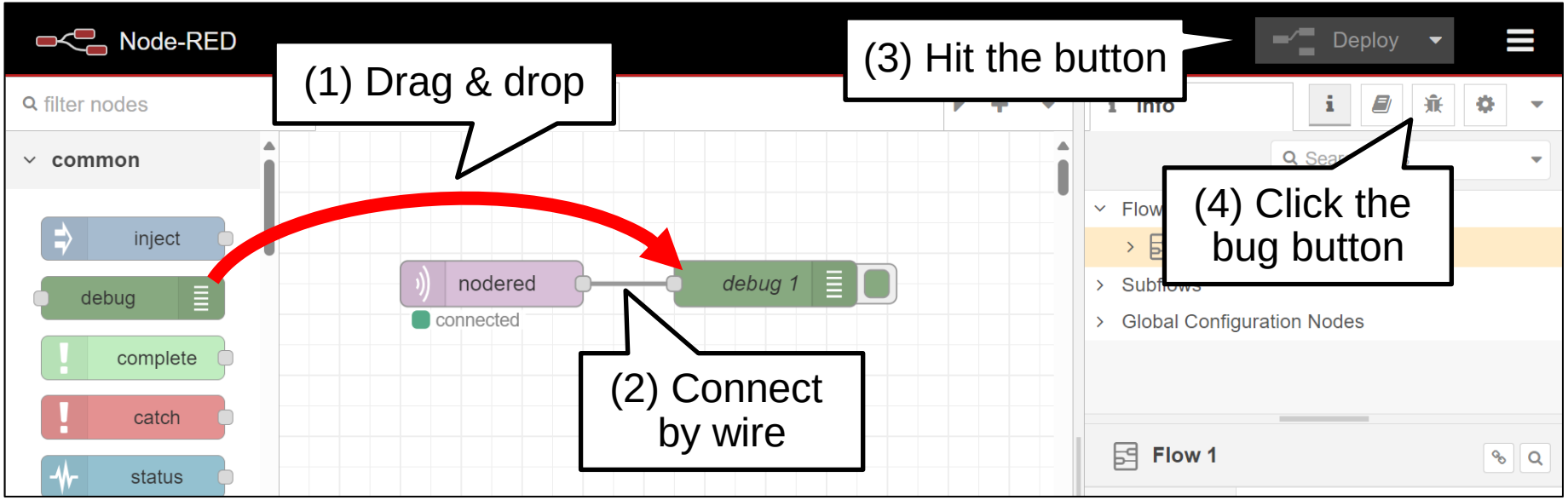
(2) Click the "Add" button to add the configuration
(3) Type "nodered" in topic field to specify the data to retrieve from the broker
(4) Click the "Done" button to apply for the settings



[Hands-on] Placing debug Node to Check the Retrieved Data

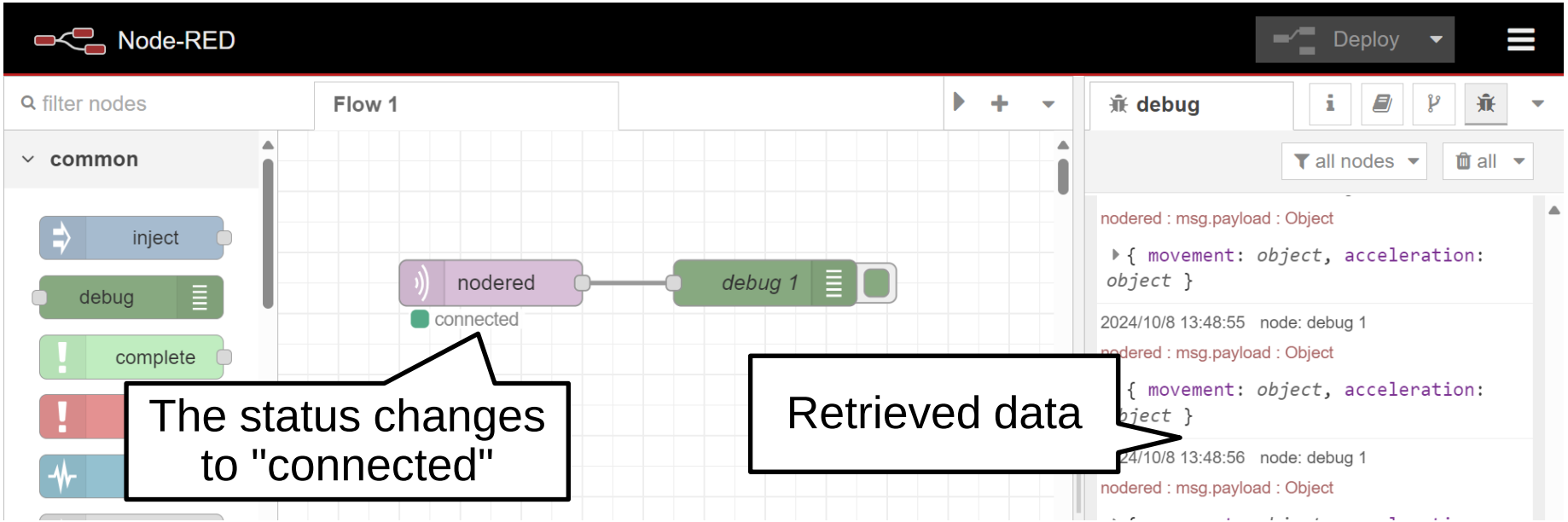
HITACHI
Inspire the Next

(1) Add the debug node next to mqtt-in node
(2) Connect the mqtt-in node and debug node by wire
(3) Hit the "Deploy" button to apply and execute the flow
(4) Click the bug icon to show the debug tab where the sensor data is observed



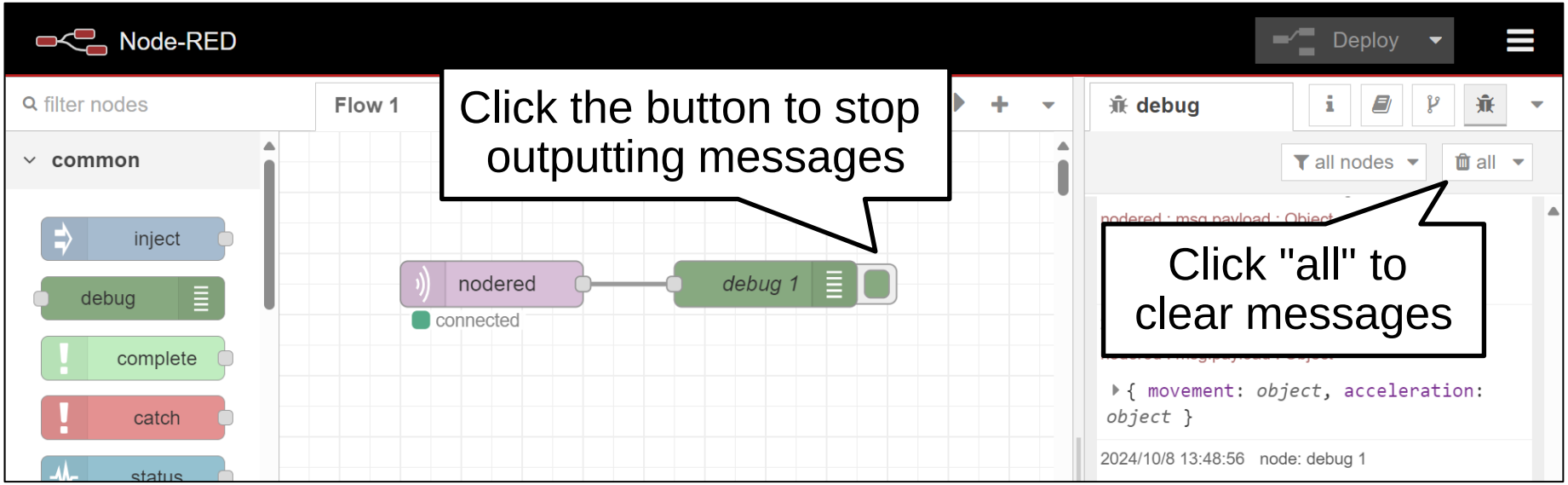
[Hands-on] Checking the Retrieved Sensor Data

- After successfully connecting to the MQTT broker, the node status under the node will have "connected" message along with green rectangle.
- The retrieved data will be outputted on the debug tab every second.



[Hands-on] How to Stop and Clear Output of debug Node

- If you want to stop outputting messages from the debug node, click the button of the debug node.
- To delete all of messages from the debug tab, click the "all" button with the trash can icon.

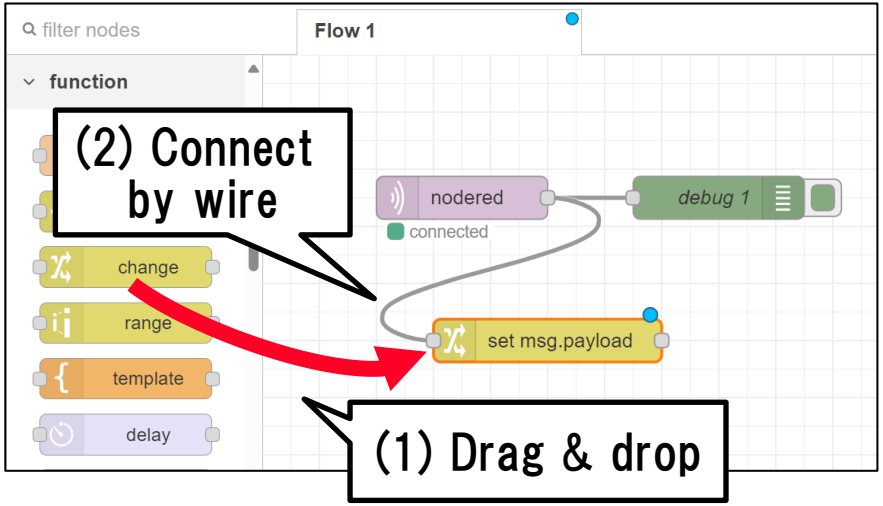


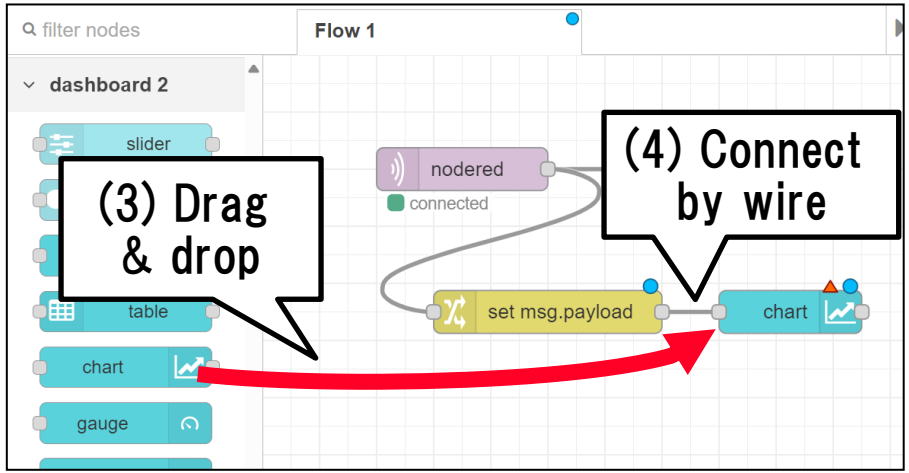
[Hands-on] Creating Flow to Output Sensor Data on Dashboard

HITACHI
Inspire the Next

To show sensor data on a time series graph, use the change and chart nodes

(1) Drag and drop the change node of the function category under the mqtt-in node
(the name will be change to "set msg.payload")
(2) Connect the mqtt-in node to the input port of change node by wire
(3) Place the chart node of the "dashboard 2" category after the change node
(4) Wire the output port of the change node to the chart node by wire



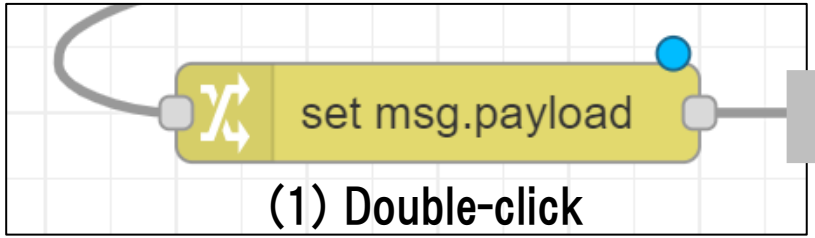


[Hands-on] Configuring change Node to Extract Specific Value

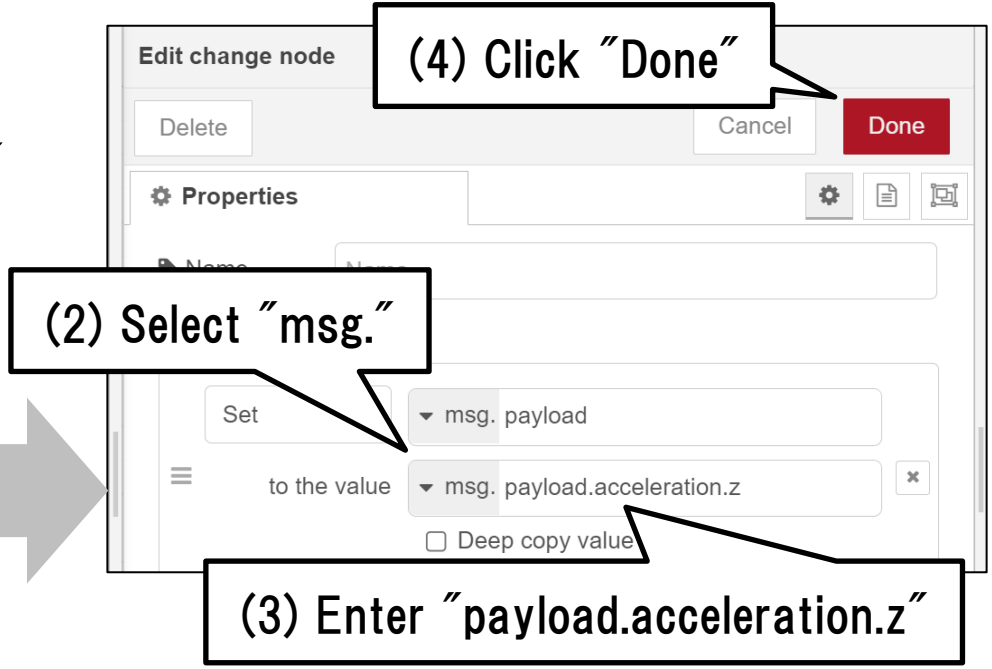
HITACHI
Inspire the Next

Edit the change node settings to extract specific value from the retrieved data

(1) Double-click the change node to open the node property settings UI
(2) Select "msg." from the data type menu of the "to the value" field
(3) In the text box, enter the "payload.acceleration.z" to extract the value
(4) Click the "Done" button



(1) Double-click



(4) Click "Done"

(2) Select "msg."

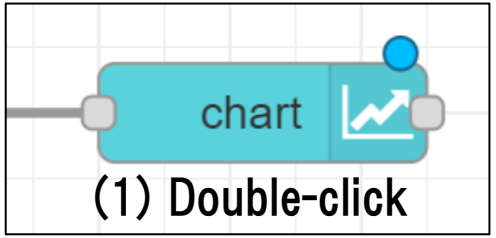
(3) Enter "payload.acceleration.z"

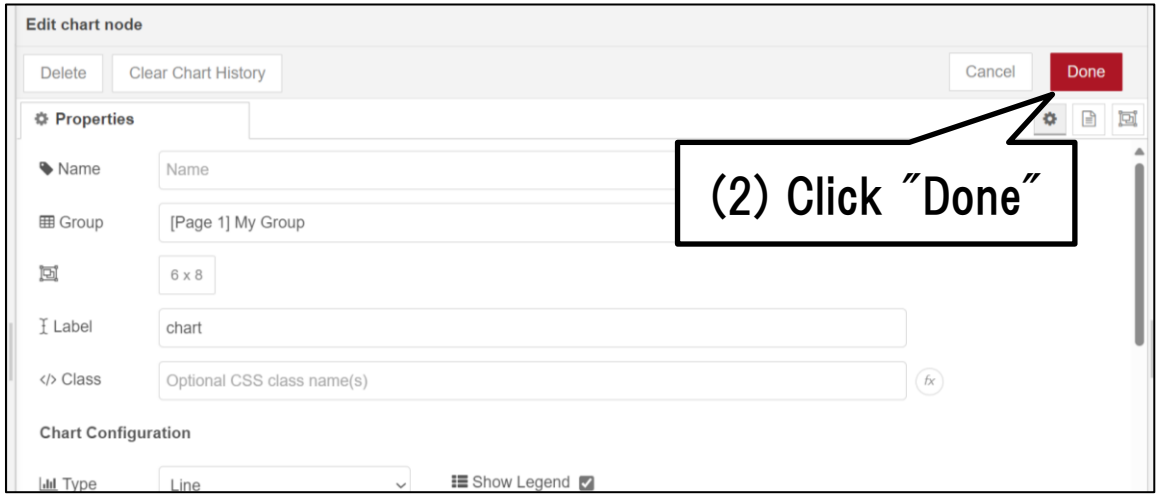
[Hands-on] Configuring chart Node

To apply the default dashboard setting, just open and close the node property UI

(1) Double-click the chart node to open the node property UI

(2) Click the "Done" button to close the node property UI
(The settings will be configured automatically without any changes)





[Hands-on] Opening Dashboard UI (1/2)

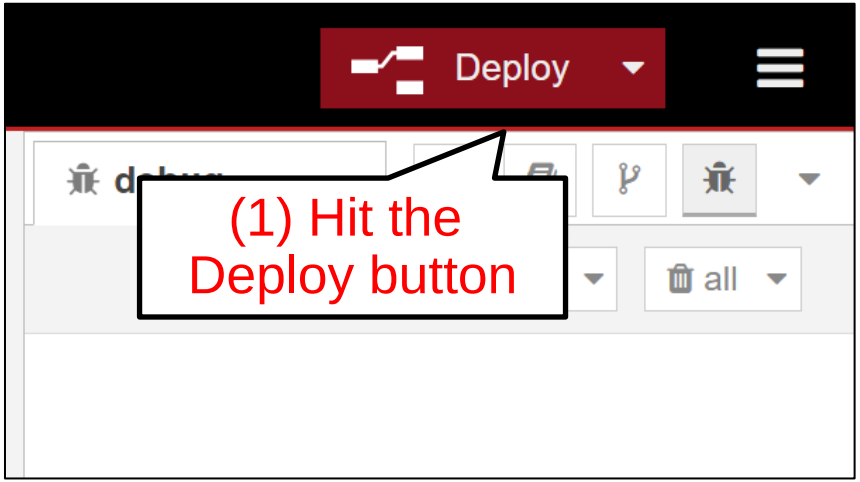
HITACHI
Inspire the Next

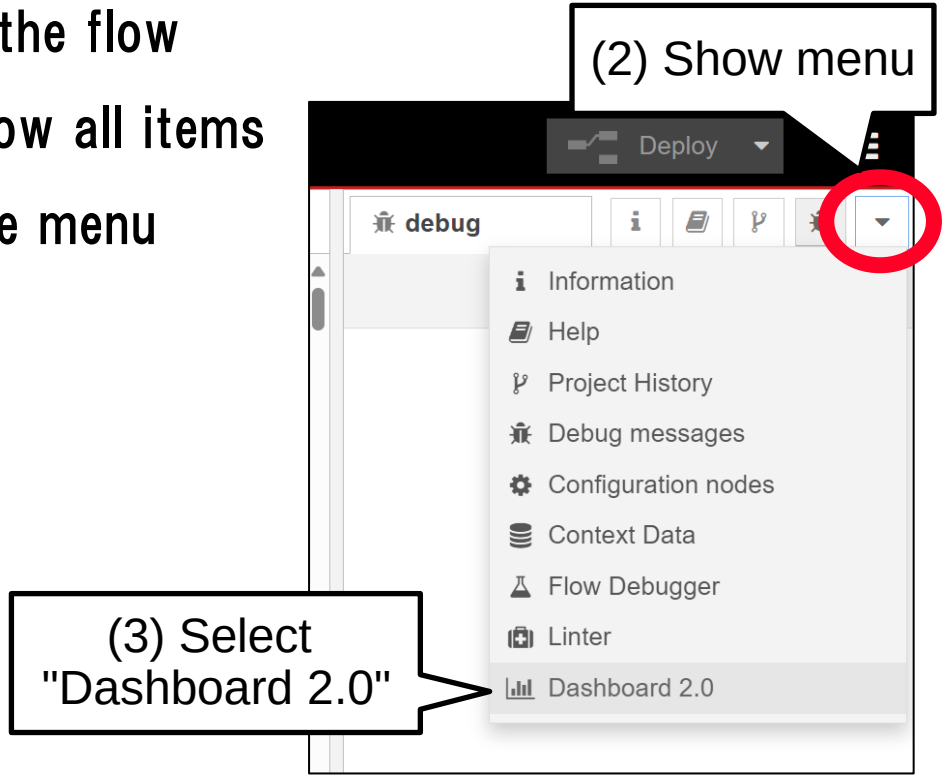
After deploying the flow, move to the dashboard UI from the flow editor

(1) Hit the "Deploy" button to execute the flow

(2) Open the menu in the sidebar to show all items

(3) Select the "Dashboard 2.0" from the menu



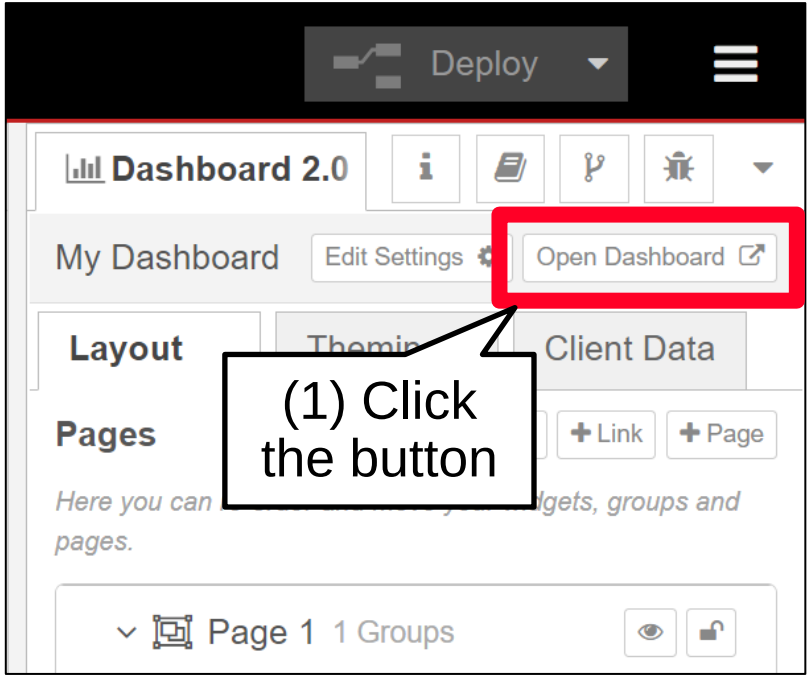


[Hands-on] Opening Dashboard UI (2/2)

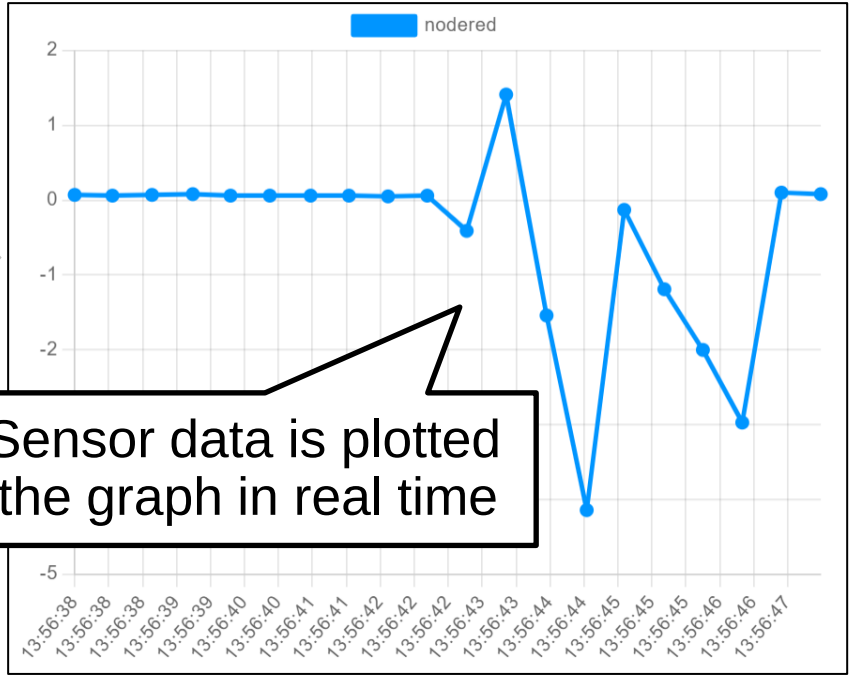
HITACHI
Inspire the Next

(1) Click the "Open Dashboard" button in the Dashboard 2.0 tab

(2) Dashboard UI in another browser tab shows the time series chart



(1) Click the button

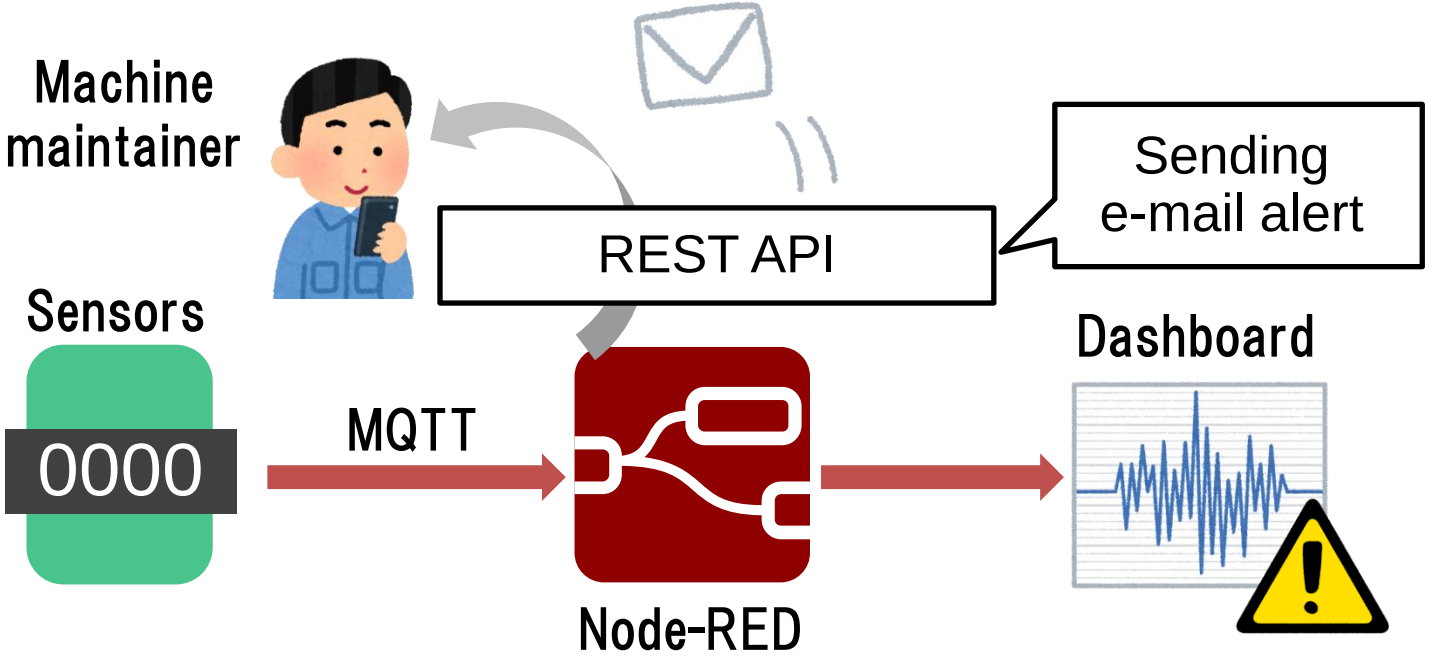


(2) Sensor data is plotted on the graph in real time

E-mail Notification

E-mail Notification

- Developing flow to send alert e-mail when the anomaly value is detected
- Using the SendGrid, e-mail delivery service, the flow sends e-mail to recipient

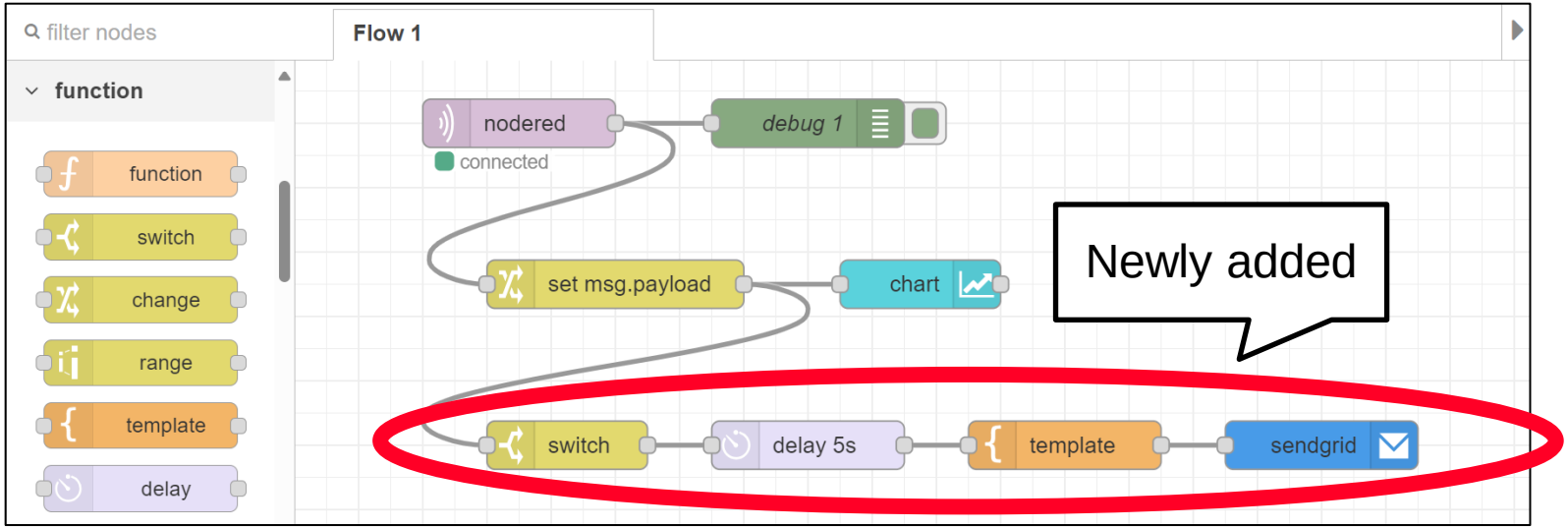


[Hands-on] Flow to Send E-mail in Case of Anomaly Value

HITACHI
Inspire the Next

Add and connect the following nodes after the change node

- switch node: node to pass if the received values match the defined condition
- delay node: node to limit the number of messages in the specified period
- template node: node to write the template text for the e-mail message
- sendgrid node: node to send e-mail using the SendGrid service

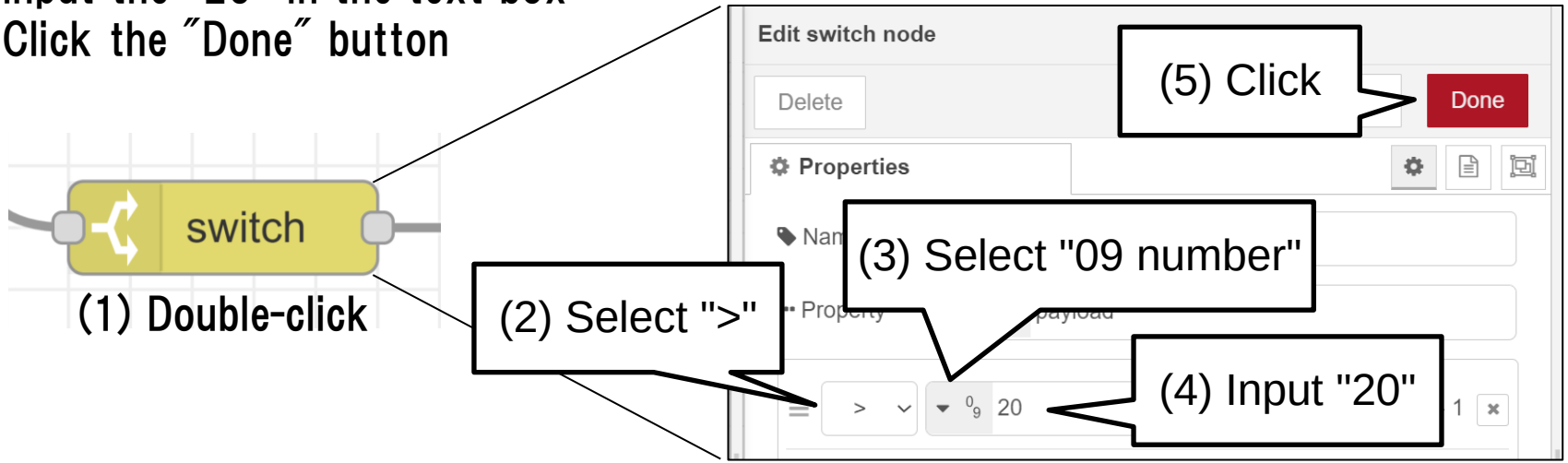


[Hands-on] Configuring switch Node for Conditional Filtering

HITACHI
Inspire the Next

Setting the threshold value to classify as an anomaly value
(In this hands-on, the received number greater than 20 is an anomaly value)

(1) Double-click the switch node to open the node property UI
(2) Change the comparison operator from "==" to ">", which means greater than
(3) Change the data type from "az string" to "09 number" from the menu
(4) Input the "20" in the text box
(5) Click the "Done" button

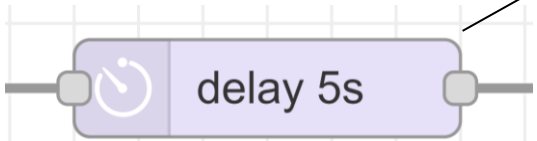


[Hands-on] Configuring delay Node to Limit Messages

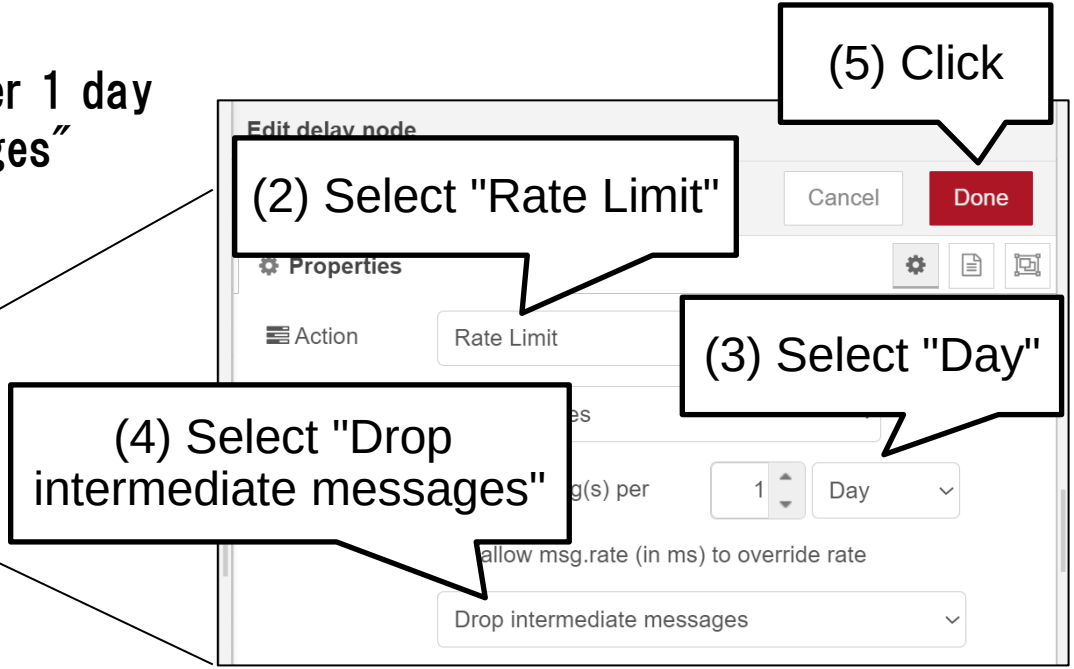
HITACHI
Inspire the Next

To prevent sending many e-mail messages in a short period of time, limit the maximum number of messages in a day using the delay node

(1) Double-click the delay node
(2) Select "Rate Limit" as action
(3) Select "Day" to set 1 message per 1 day
(4) Select "Drop intermediate messages" to remove overflow messages
(5) Click the "Done" button



(1) Double-click

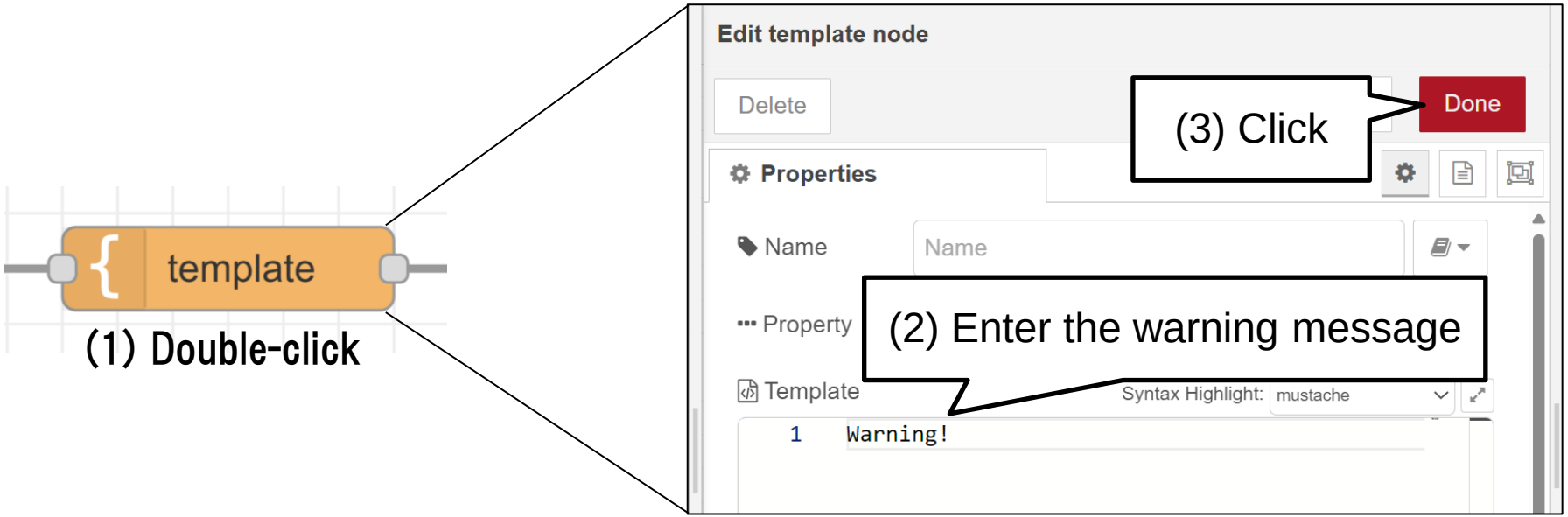


[Hands-on] Configuring template Node to Write Fixed Text

HITACHI
Inspire the Next

Defining the template message for sending e-mail

(1) Double-click the template node to open the node property UI
(2) In the template editor area, type the "Warning!" as the e-mail body
(3) Click the "Done" button



[Hands-on] Configuring SendGrid Node

HITACHI
Inspire the Next

(1) Double-click the sendgrid node to open the node property UI
(2) Copy and paste the following API key into the "API key" field

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

(3) Input the sender's e-mail address, "hiac.it-kensyu74.hw@hitachi.com" into the "From" field
(4) Input your e-mail address in the "To" field
(5) Click the "Done" button

(1) Double-click

(2) Paste an API key

(3) Input the sender's e-mail

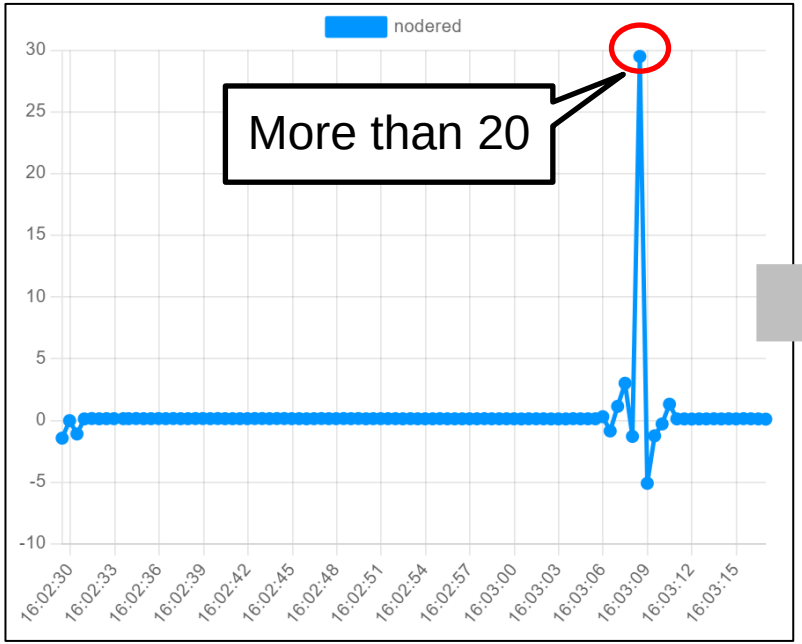
(4) Input the recipient e-mail

(5) Click

[Hands-on] Checking Behaviors of Sending E-mail

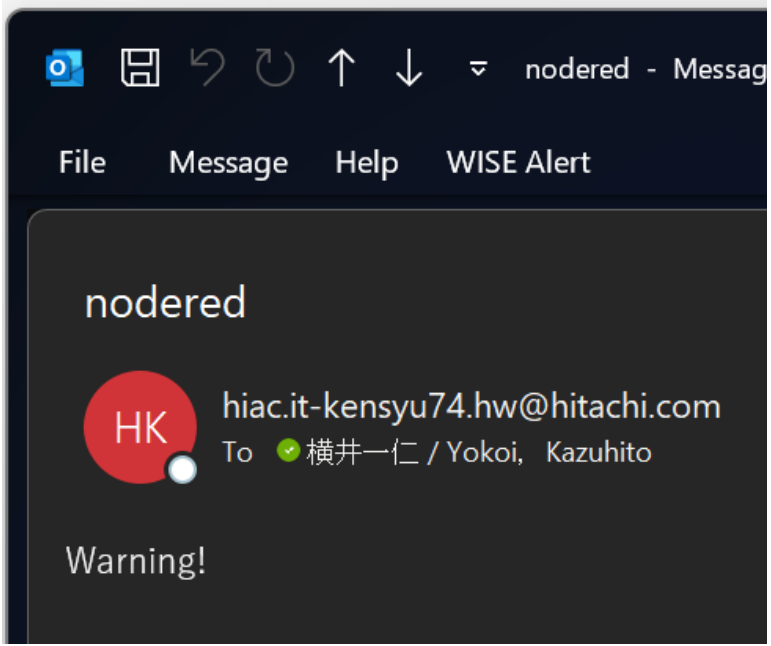
HITACHI
Inspire the Next

After the flow is deployed, Node-RED will send the alert e-mail to the recipient when the value is greater than 20.



Time-series graph on application

Send e-mail

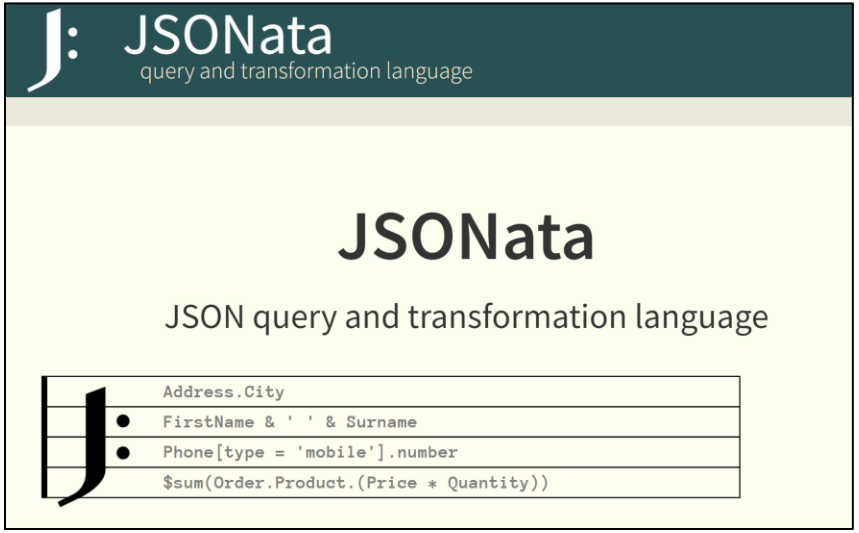


Received e-mail

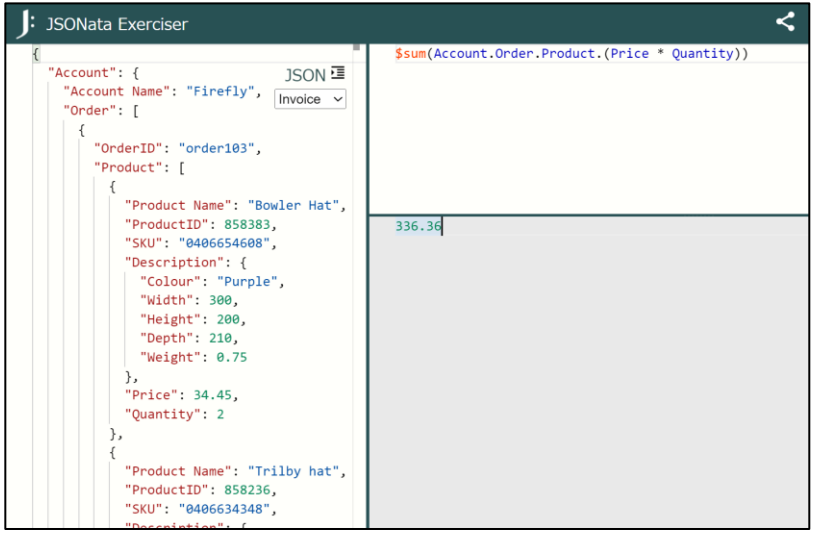
JSONata

A lightweight query and transformation language for JSON data

- Node-RED natively supports JSONata expressions
- It allows users to implement simple data handling without complex code



JSONata website



JSONata Exerciser

https://jsonata.org/

Example of JSONata

Using only the path to the key, "books.title", JSONata can extract the values as arrays from arrays containing JSON structures.

```
{
  "books": [
    {
      "title": "Practical Node-RED Programming",
      "author": "Taiji Hagino"
    },
    {
      "title": "First time with Node-RED",
      "author": "Node-RED User Group Japan"
    },
    {
      "title": "Building your first app with Node-RED",
      "author": "Hitachi Node-RED Evangelists"
    }
  ]
}
```

Source JSON data

Transform JSON data using JSONata

```
books.title
```

```
[
  "Practical Node-RED Programming",
  "First time with Node-RED",
  "Building your first app with Node-RED"
]
```

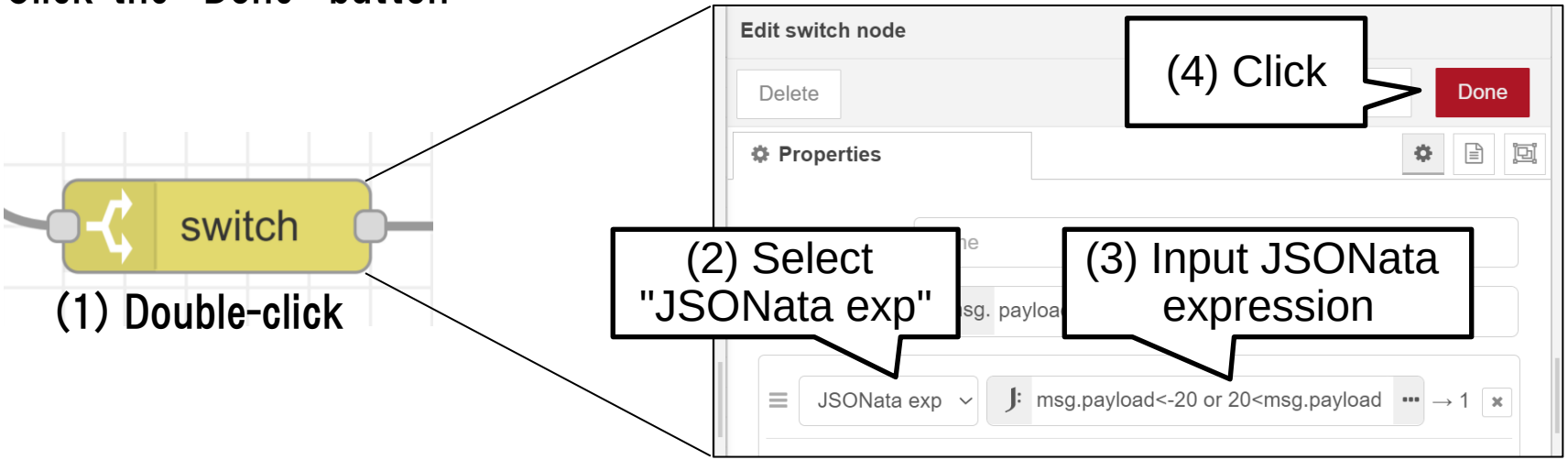
Output JSON data

[Hands-on] Configuring switch Node for Conditional Filtering

HITACHI
Inspire the Next

Setting the threshold value to classify as an anomaly value
(In this slide, the received value between -20 and 20 is an anomaly value)

(1) Double-click the switch node to open the node property UI
(2) Change the comparison operator from ″>″ to ″JSONata exp″
(3) Input the ″msg.payload<-20 or 20<msg.payload″ in the text box
(4) Click the ″Done″ button

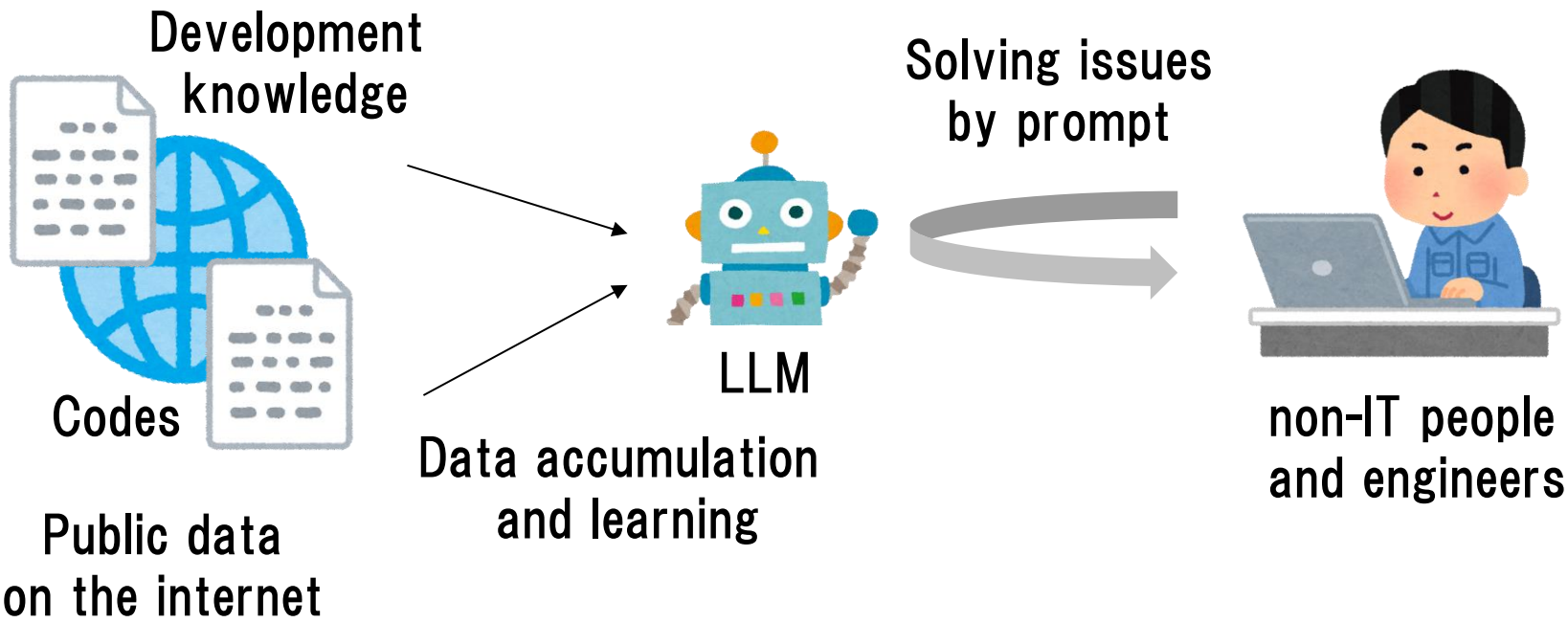


ChatGPT with Node-RED

Large Language Model

HITACHI
Inspire the Next

Both non-IT people and engineers have used LLM like ChatGPT to solve the programming and system problems by themselves.



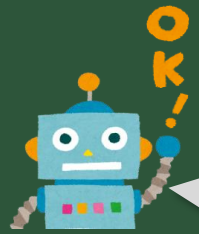
Code Generation using ChatGPT

HITACHI
Inspire the Next

Generating code to detect sudden changes in the sensor data

Code to output warning when received value is 50% greater than the average of the last 7 values

non-IT engineer



ChatGPT

Here is source code for Node-RED.

```
const values = context.get("values")
|| [];
const threshold = mean * 1.5; // 50%の閾値
values.push(msg.payload);
if (values.length > 7) {
  values.shift();
}
```

Code to detect anomaly value

[Algorithm of anomaly detection]

(1) Add value

Queue to store 7 values

(3) Calculate average

(4) Warn when latest value is 50% greater than average

(2) Remove the oldest value

ChatGPT can generate the complex code in a few seconds.

[Hands-on] Code Generation using ChatGPT

HITACHI
Inspire the Next

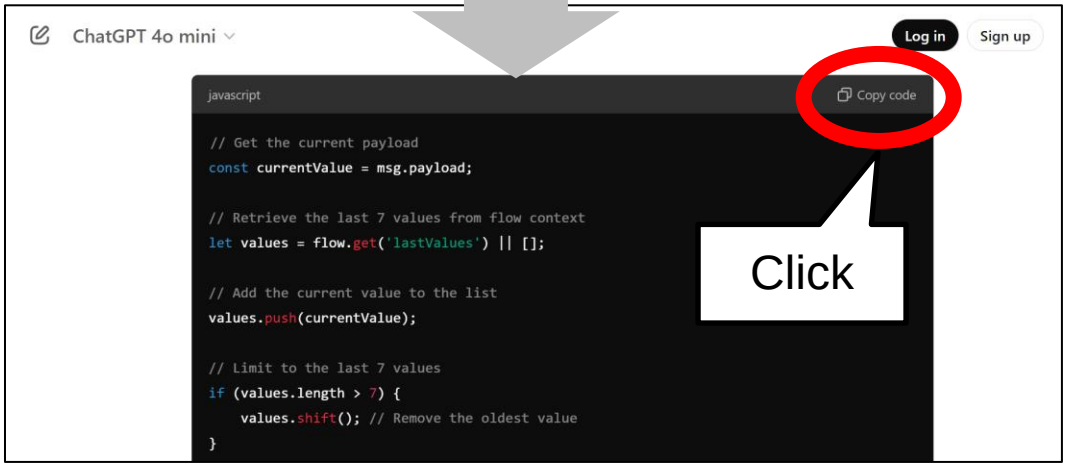
(1) Access the following URL from your browser
https://chatgpt.com/

(2) Input the following prompt in the text box

> Use node.warn when msg.payload is 50% greater than the average of the last 7 values stored in the flow context.

(3) Click the "copy code" button

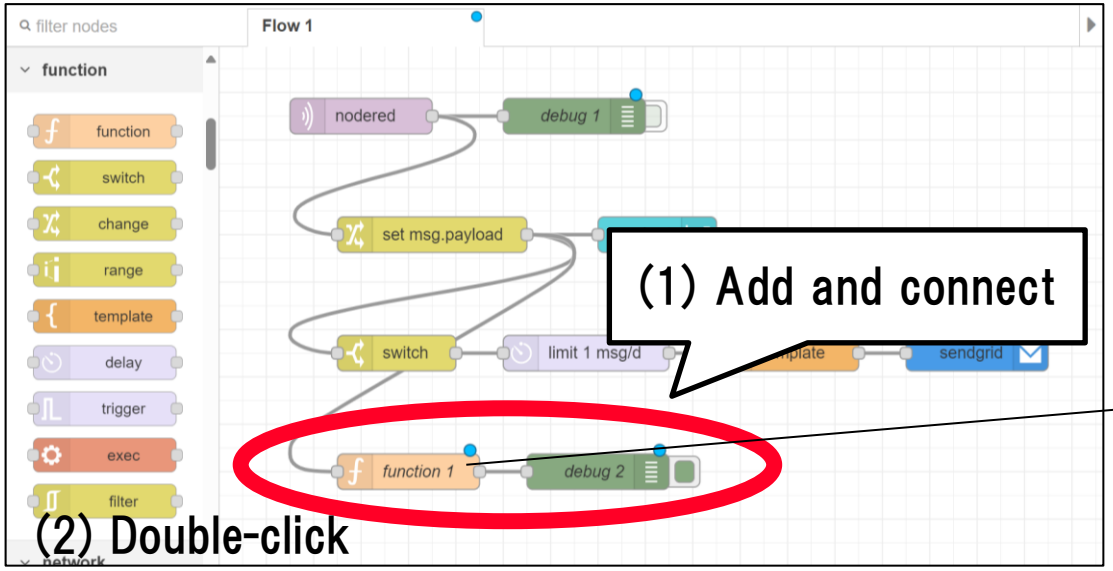
(If there is no code in the result, try typing the same prompt again)



[Hands-on] Adding Code to the function Node

HITACHI
Inspire the Next

(1) Add function node and debug node after the change node and wire them
(2) Double-click the function node to open the node property UI
(3) Paste the code from the clipboard into the editor of the "On Message" tab
(4) Click the "Done" button to apply the code

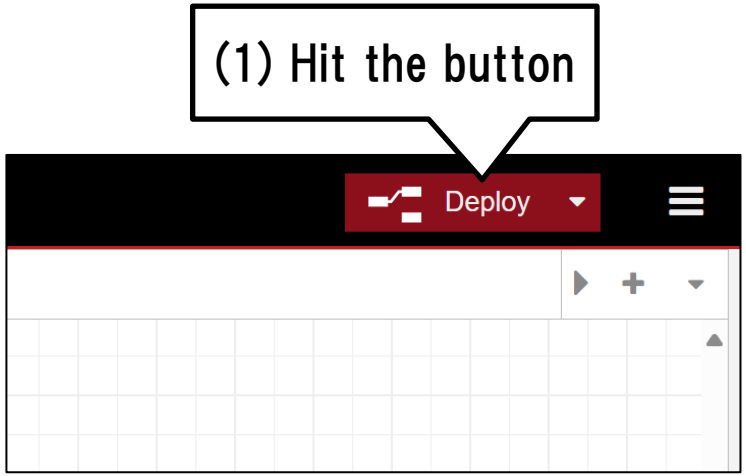


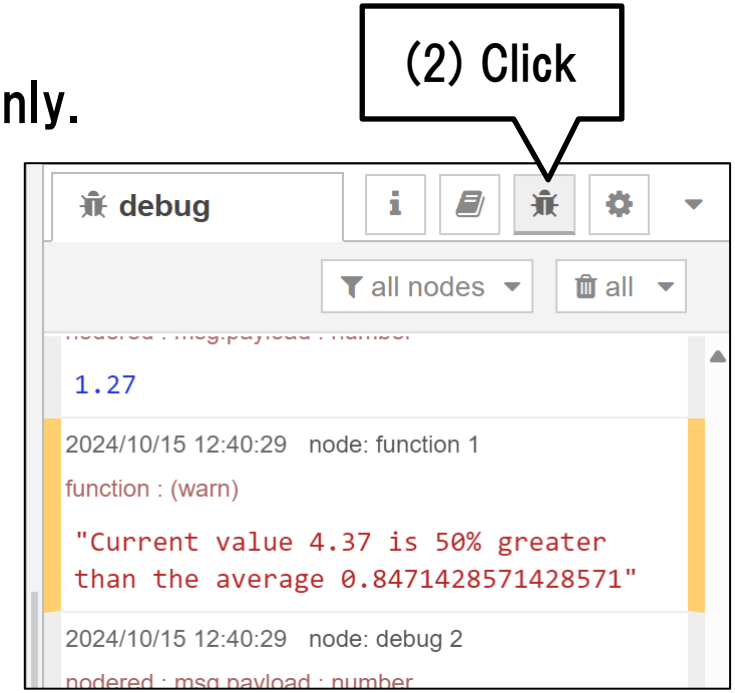
[Hands-on] Executing the Flow

(1) Hit the "Deploy" button to execute the flow

(2) Open the debug tab and check the warning message

-> Warning message will be displayed
when the received value is changed suddenly.





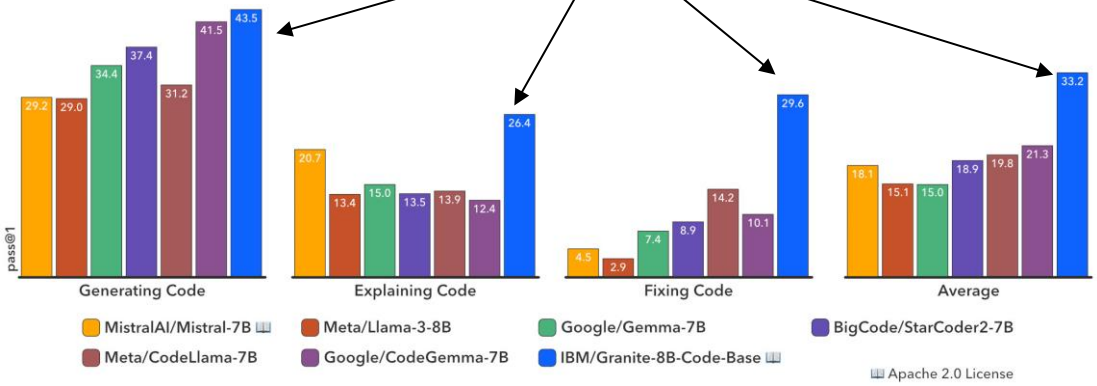
Granite Code with Node-RED

Granite Code Model

Local LLM for programming tasks released by IBM Research in May 2024

- High quality compared to other models
- Apache-2.0 license
- 116 programming languages supported
- Open data source
- Model size:
  - 3B parameters: 2.0 GB
  - 8B parameters: 4.6 GB
  - 20B parameters: 12 GB
  - 34B parameters: 19 GB

Granite Code



Quality Comparison with Local LLMs

https://github.com/ibm-granite/granite-code-models

Calculating Distance between Two Locations

HITACHI
Inspire the Next

Generating code to calculate the distance between two locations consist of latitude and longitude values

[Haversine Formula]

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

It takes a time to write the code :(

[Hands-on] Code Generation using Granite Code Model

HITACHI
Inspire the Next

(1) Access the following URL from your browser

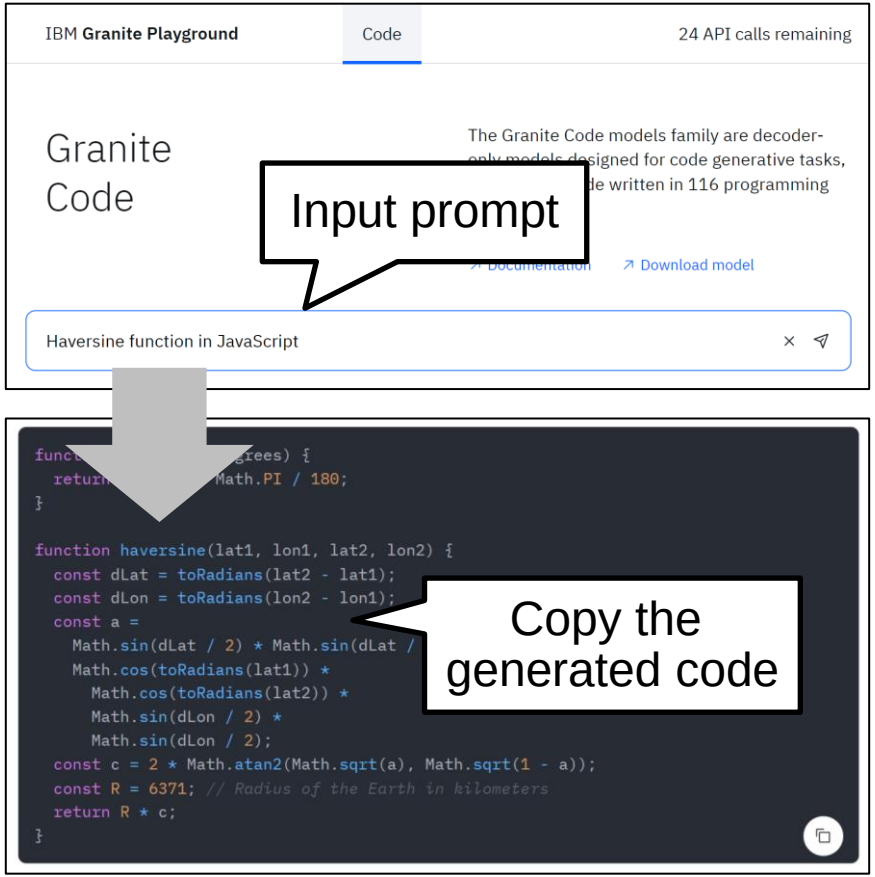
https://www.ibm.com/granite/playground/code/

(2) Input the following prompt in the text box

Haversine function in JavaScript

(3) Copy the generated code

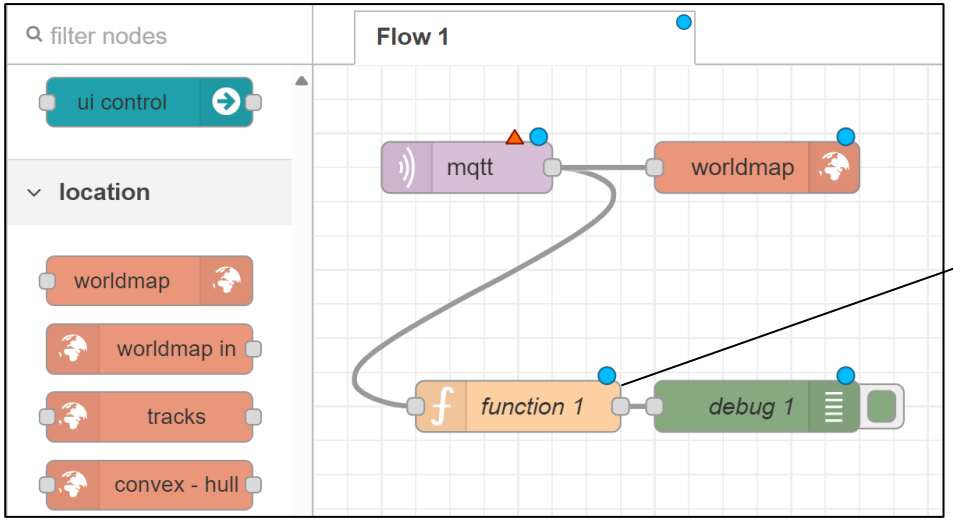
(If there is no code in the result, try typing the same prompt again)

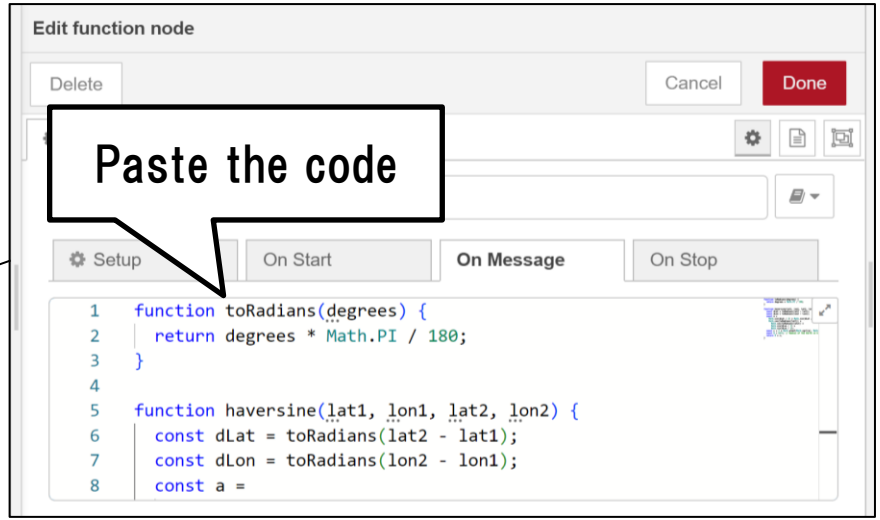


[Hands-on] Creating Flow to Show Distance and Location on Map

HITACHI
Inspire the Next

(1) Add mqtt-in node and red worldmap node and then connect them
(2) Configure mqtt-in node to use shiftr.io MQTT broker with "nodered" topic
(3) Add function node and debug node after mqtt-in node and connect them
(4) Double-click the function node to open the node property UI
(5) Paste the code from the clipboard into the editor of the "On Message" tab





[Hands-on] Modifying code to call generated function

HITACHI
Inspire the Next

(1) Add the following code that specifies two locations at the end of the lines

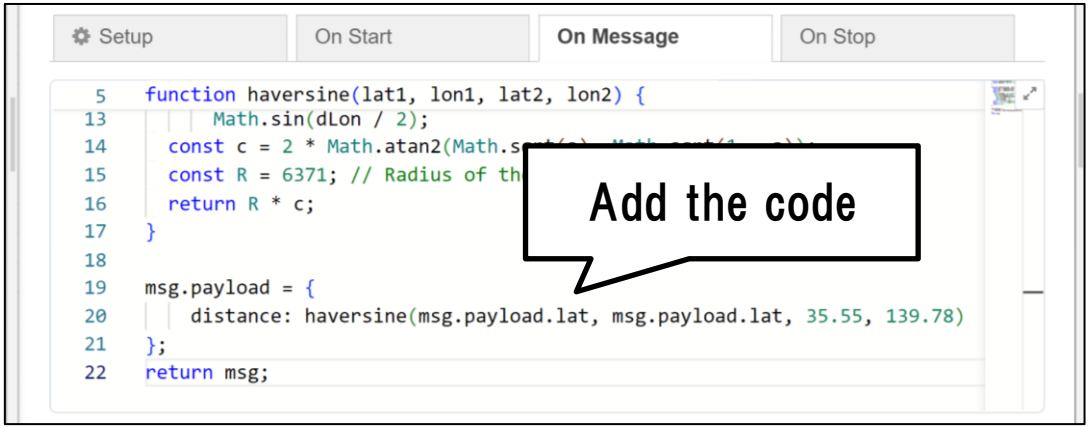
```
msg.payload = {
    distance: haversine(msg.payload.lat, msg.payload.lat, 36.09, -115.15)
};
return msg;
```

Latitude and longitude of the received data

Latitude and longitude of LAS Airport

(2) Click the "Done" button of the function node to apply the code

(3) Hit the "Deploy" button to execute the flow

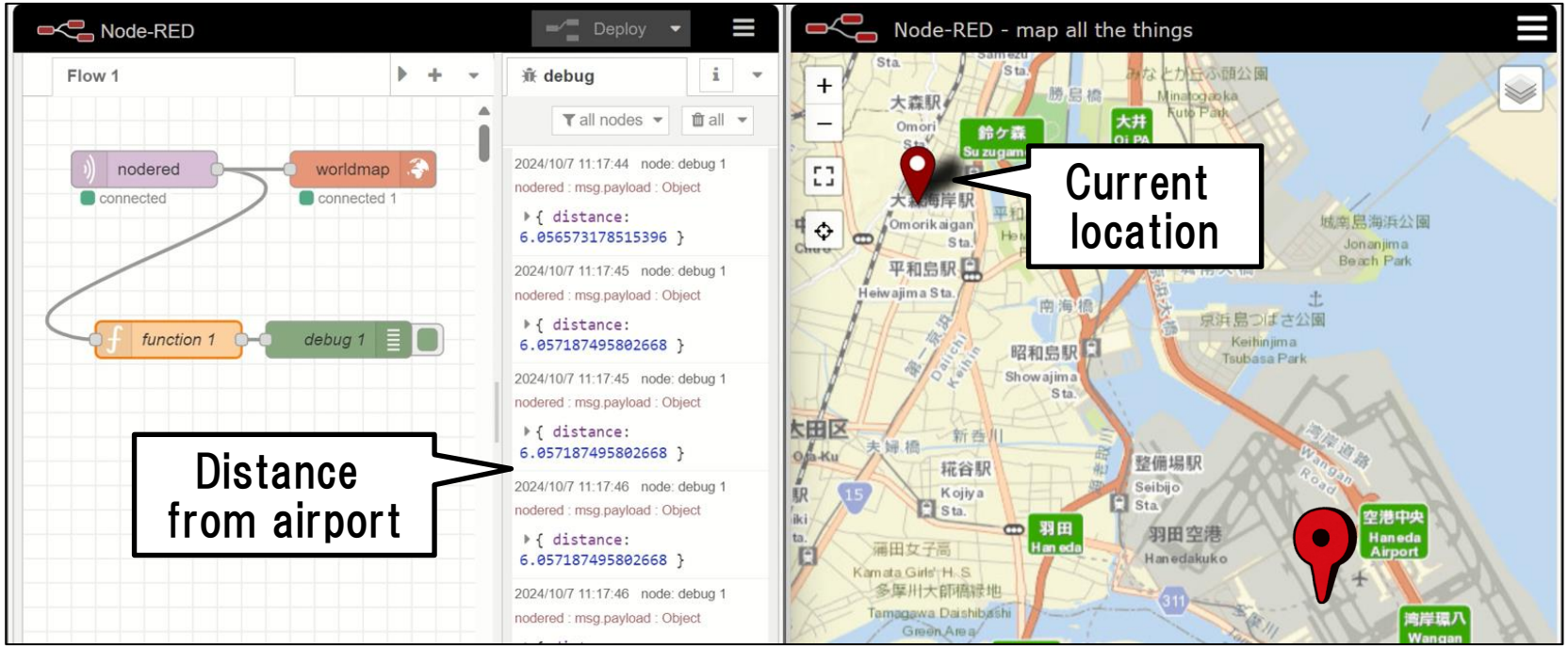


* Screenshot is the case of Haneda Airport

[Hands-on] Executing the Flow

Access the following URL to open map

https://<GitHub Codespaces URL>/worldmap



Conclusion

Conclusion

HITACHI
Inspire the Next

Node-RED is a visual programming tool ideal for rapid development.

- It natively supports IBM open technologies such as MQTT and JSONata.
- Recently, leveraging the latest LLMs like Granite Code and ChatGPT is the hot topic in Node-RED.

Try to develop your own systems by yourself.

[F.Y.I] Delete Codespaces Environment

When you have finished hands-on,
you can delete the Codespaces environment by following steps.

(1) Go to the Codespaces page

https://github.com/codespaces

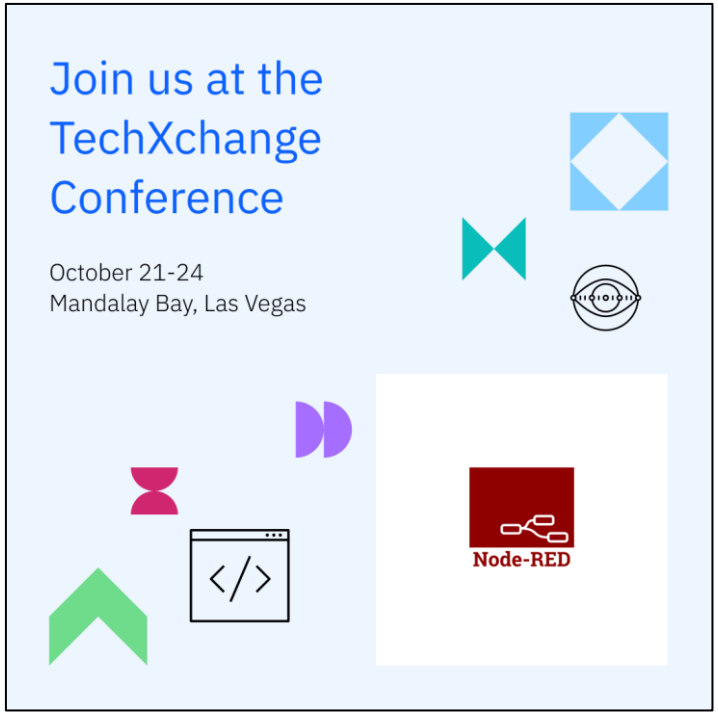
(2) Click the "・・・" of the Codepases

(3) Select "Delete" item on the menu

Node-RED Sessions at TechXchange Conference

HITACHI
Inspire the Next

- Boosting Productivity of Node-RED with Large Language Model
  - October 21 12:00PM - 12:30PM
  - South, Reef D, Level 2
- Node-RED hands-on with Open Technologies
  - October 22 3:00PM - 4:00PM
  - South, Banyan F, Level 3
- Tips for controlling HVAC and lighting with a voice from xR using Code Engine and watsonx.ai!
  - Thursday, October 24 11:30AM - 12:30PM
  - South, Surf C, Level 2



https://reg.tools.ibm.com/flow/ibm/techxchange24/sessioncatalog/page/sessioncatalog?search=Node-RED

HITACHI
Inspire the Next

END

Node-RED Hands-on with Open Technologies

October 22, 2024

Kazuhito Yokoi
Hitachi Academy

Trademarks

- Node-RED is a trademarks of OpenJS Foundation.
- GitHub Copilot is a trademark of GitHub, Inc.
- SendGrid is a trademark of Twilio SendGrid, Inc.
- Granite Code, Code Engine and watsonx.ai are trademarks of International Business Machines Corporation.
- ChatGPT is a trademark of OpenAI, Inc.
- Other company, product or service names may be trademark or registered trademark of others.