HITACHI
Inspire the Next

Boosting Productivity of Node-RED with Large Language Model

October 21, 2024

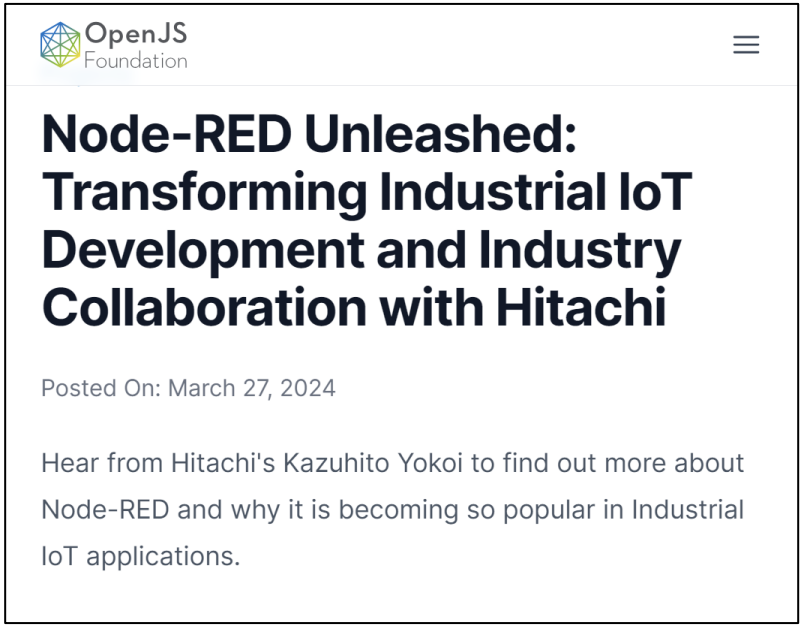
Kazuhito Yokoi
Hitachi Academy

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



My interview on
OpenJS Foundation blog

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

HITACHI
Inspire the Next

Contents

HITACHI
Inspire the Next

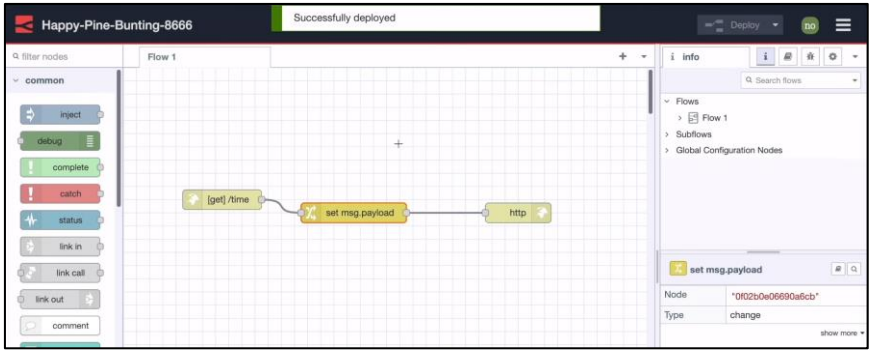
What is Node-RED?

What is Node-RED?

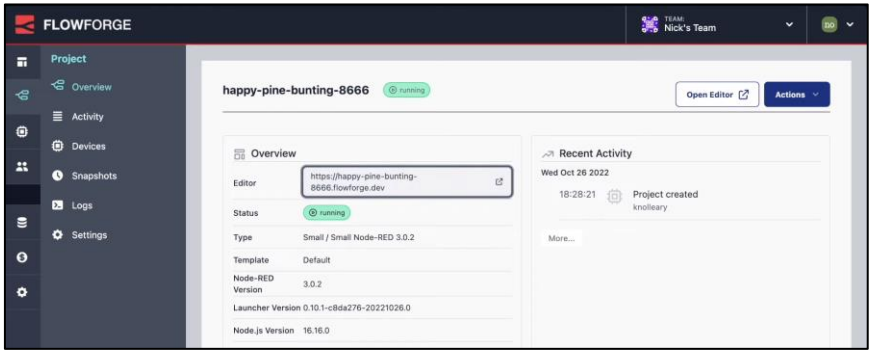
HITACHI
Inspire the Next

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/

The Number of Downloads

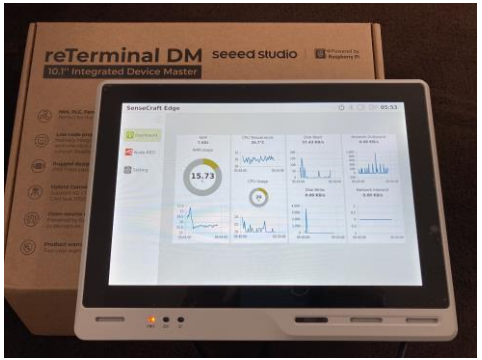
HITACHI
Inspire the Next

In 10 years, the number of downloads from the npm has gradually increased.

https://npm-stat.com/charts.html?package=node-red&from=2015-01-01&until=2024-12-31

Node-RED Community

- 229 contributors to Node-RED project
- 3rd party 5081 nodes available
- Pre-installed edge devices
  - reTerminal DM, Seeed Studio
  - EdgePi
- Node-RED events
  - Node-RED conference
  - Regular meetups
- Node-RED user companies
  Siemens, Schneider, NEC, Fujitsu, HPE, Hitachi and others



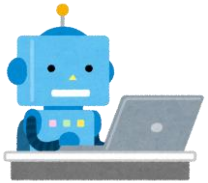
https://event.ospn.jp/slides/OSC2021-OnlineSpring/20210305osc_hitachi_yokoi.pdf

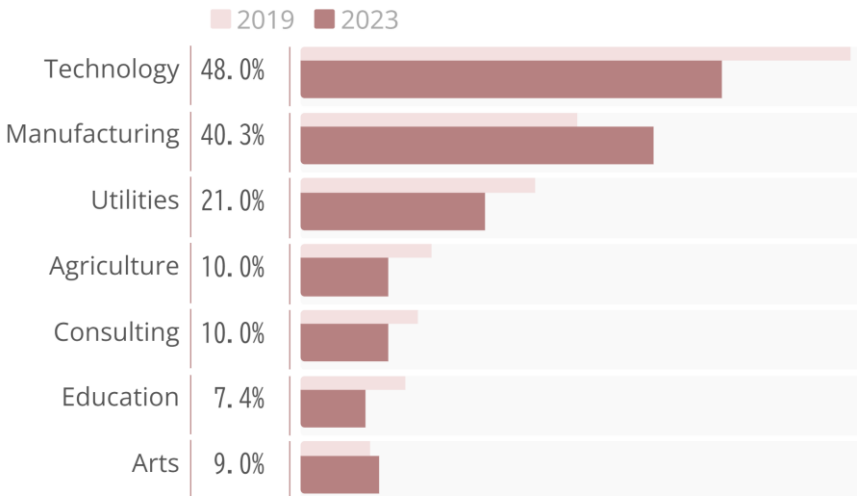
Node-RED Use Cases

HITACHI
Inspire the Next

Node-RED is widely used in the following industries and use cases:

- Technology
  - Chatbot
  - API Gateway
- Manufacturing Industry
  - Factory Automation
  - Data Visualization
- Utilities
  - Energy Management
  - Predictive Maintenance
- Agriculture
  - Smart Agriculture





Community survey
about major industries

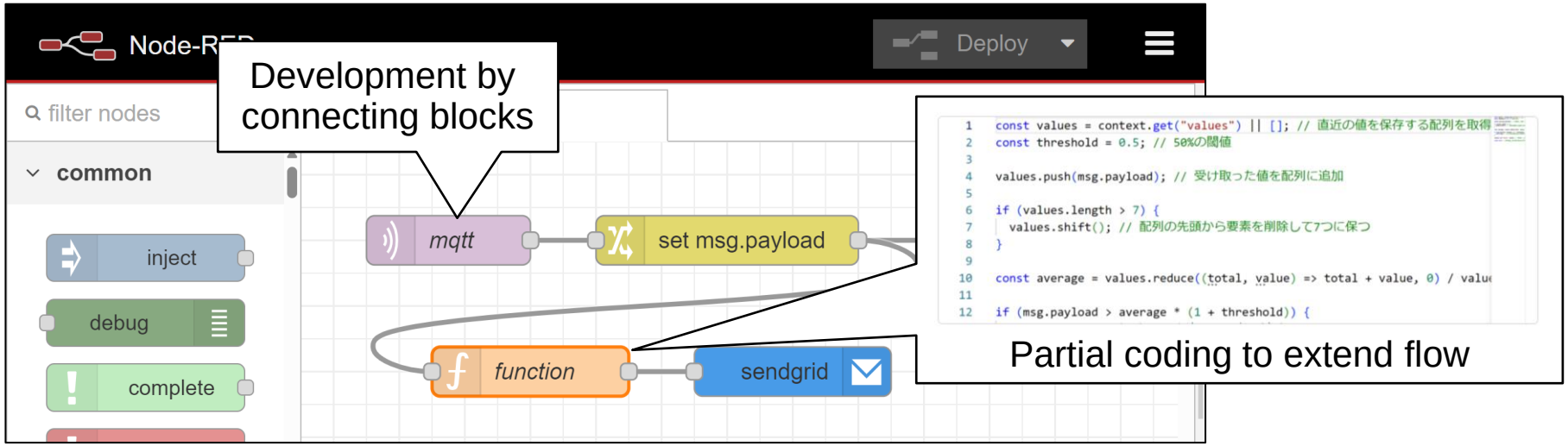
https://nodered.org/about/community/survey/2023/

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

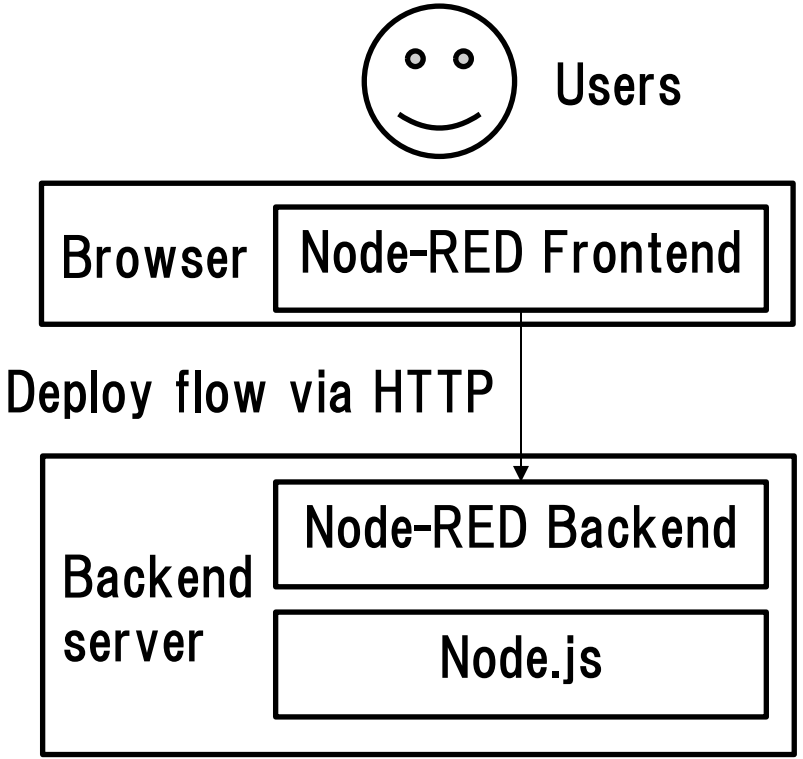


Architecture of Node-RED

HITACHI
Inspire the Next

Node-RED consists of a front-end UI and a back-end server.

- In flow development, users use the Node-RED front-end UI in their browser.
- The Node-RED back-end runs on Node.js as an HTTP server.



How to Install Node-RED

HITACHI
Inspire the Next

Type a few commands in terminal on the local PC.

(1) Install Node.js with Node Version Manager

```
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.40.0/install.sh | bash
nvm install 22
```

(2) Restart the terminal

(3) Install Node-RED using npm command

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

(4) Start the Node-RED server

```
node-red
```

-> You can access the Node-RED from "http://localhost:1880".

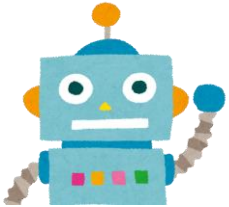
Adding AI Functionality to Increase Productivity in Development

HITACHI
Inspire the Next

To increase the productivity of Node-RED, we have recently adopted technologies that support coding by AI.

I will introduce the following background technologies with demonstrations:

- Large Language Model
- Granite Code
- Code completion
- Plugin for Node-RED



Coding support by AI

```
1  const values = context.get("values") || []; // 直近の値を保存する配列を取得
2  const threshold = 0.5; // 50%の閾値
3
4  values.push(msg.payload); // 受け取った値を配列に追加
5
6  if (values.length > 7) {
7    values.shift(); // 配列の先頭から要素を削除して7つに保つ
8  }
9
10 const average = values.reduce((total, value) => total + value, 0) / value
11
12 if (msg.payload > average * (1 + threshold)) {
```

Coding to extend flow

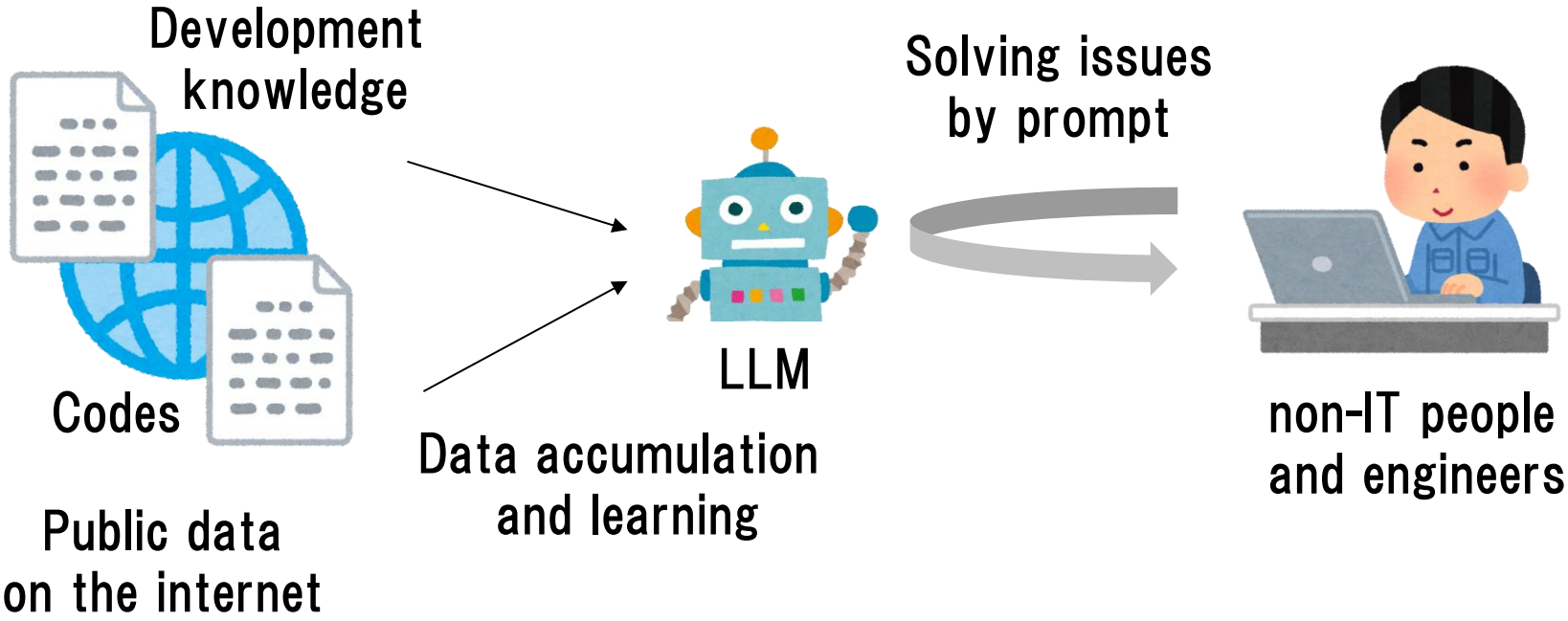
HITACHI
Inspire the Next

Large Language Model (LLM)

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.

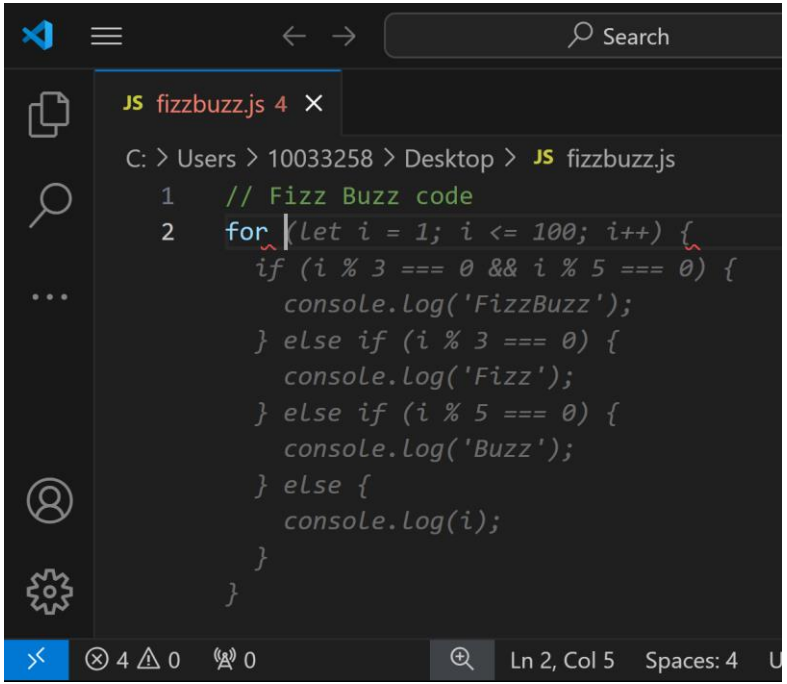


Large Language Model for Coding

HITACHI
Inspire the Next

LLM for software coding tasks to increase software development productivity

- Key functionalities
  - Code generation from comment
  - Code explanation from code
  - Code fixing
- Major LLMs
  - Public Service LLMs (GitHub Copilot)
  - Local LLMs (Granite Code)

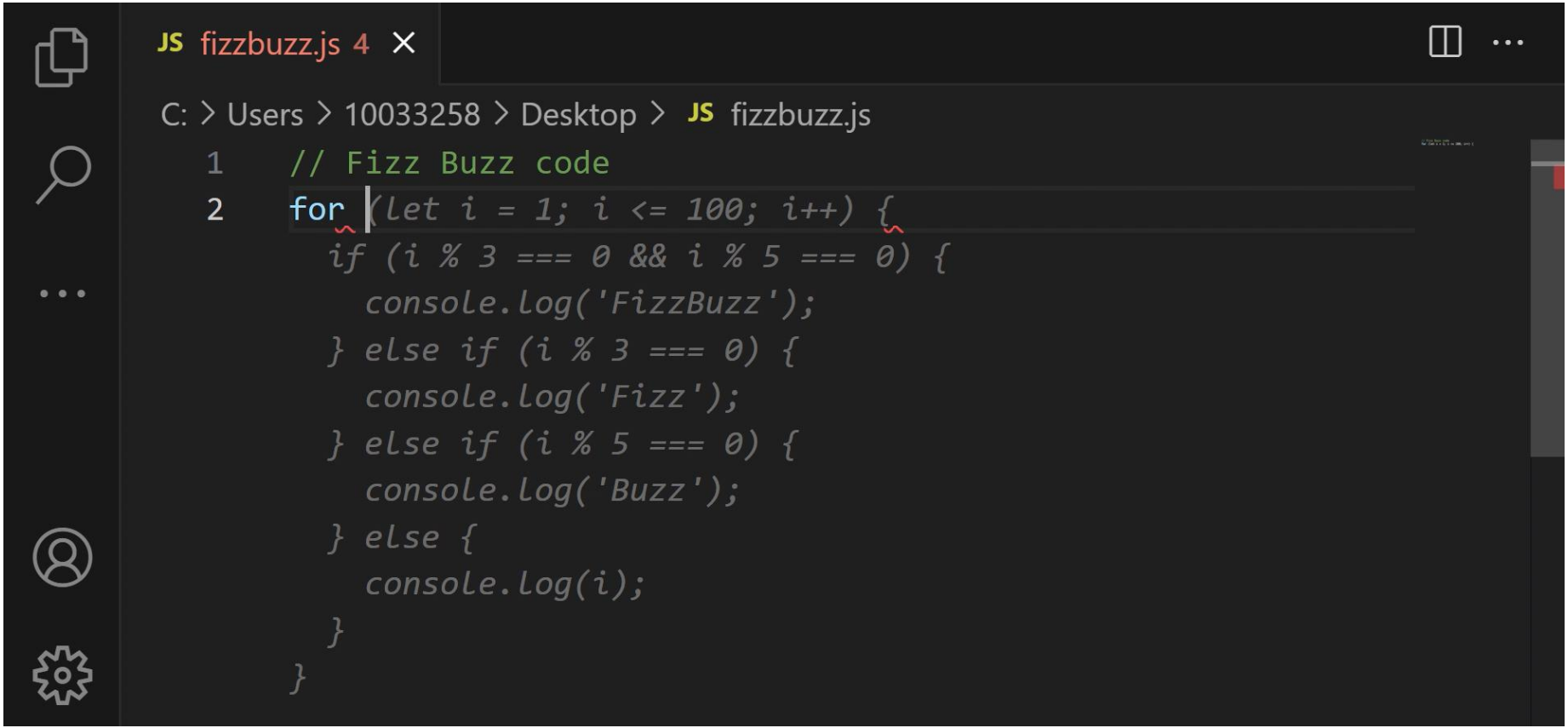


Code generation on GitHub Copilot

https://github.com/features/copilot

Demonstration: Code Generation Using GitHub Copilot

HITACHI
Inspire the Next



Comparison of Public Service LLMs and Local LLMs

HITACHI
Inspire the Next

As of 2024, we evaluated two types of LLMs as follows.

#	Item	Public Service LLMs	Local LLMs
1	Cost	Monthly or PAYG	Free
2	Speed	Good	Depends on PC spec
3	Offline use	Impossible	Possible
4	Quality	Good	Depends on PC spec

(Green color means positive)

Local LLMs are suitable for the following cases:

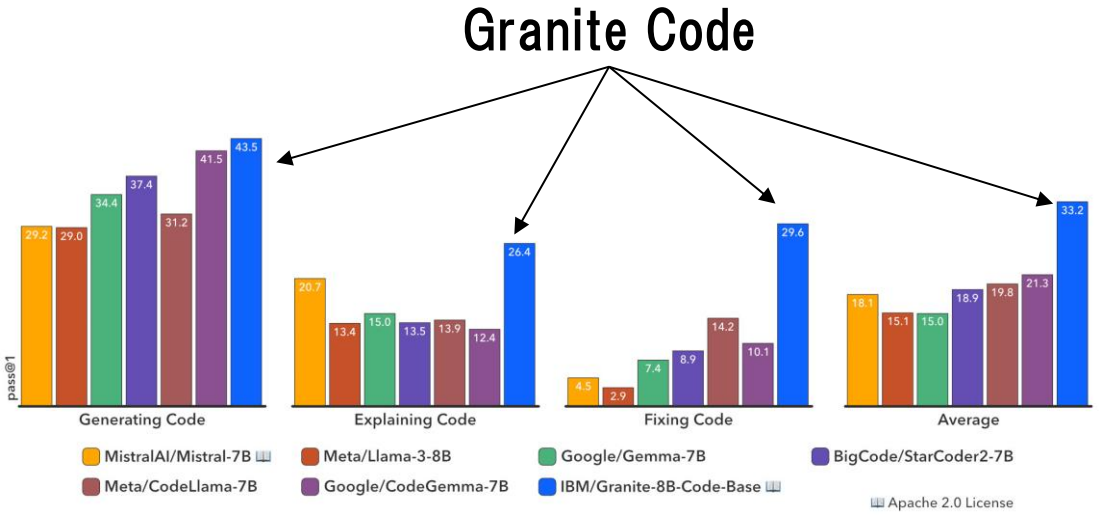
- Development like testing and documentation, which uses LLM many times
- Projects under the compliance policies such as export control

Granite Code Model

HITACHI
Inspire the Next

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



Quality Comparison with Local LLMs

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

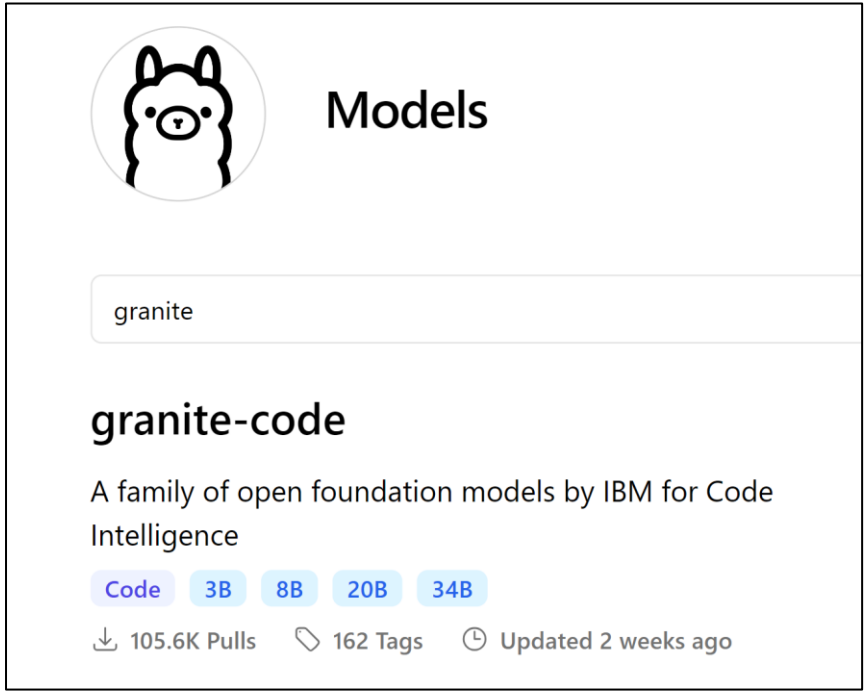
Ollama

HITACHI
Inspire the Next

Backend server software for running LLMs on the local PC

- Available models
  - Mistral
  - CodeLlama
  - Gemma
  - StarCode2
  - Granite Code
- PC specs used in the Granite Code
  - Lenovo ThinkPad T14s Gen 2 (32 GB memory)
  - M1 MacBook Air (8 GB memory)

https://ollama.com/library/granite-code



Granite Code model on Ollama page

How to Use Granite Code Model on Ollama

HITACHI
Inspire the Next

After setting up Ollama, you can execute model through CLI or REST API.

- Install Ollama for Linux environment

```
curl -fsSL https://ollama.com/install.sh | sh
ollama serve
```

- Execute model
  - Command line interface

```
ollama run granite-code:3b-instruct
```

  - REST API

```
curl http://localhost:11434/api/generate -d '{ "model": "granite-code:3b-instruct", "prompt": "Hello world in Node.js", "stream": false }'
```

https://github.com/ollama/ollama/blob/main/docs/api.md#request-no-streaming

HITACHI
Inspire the Next

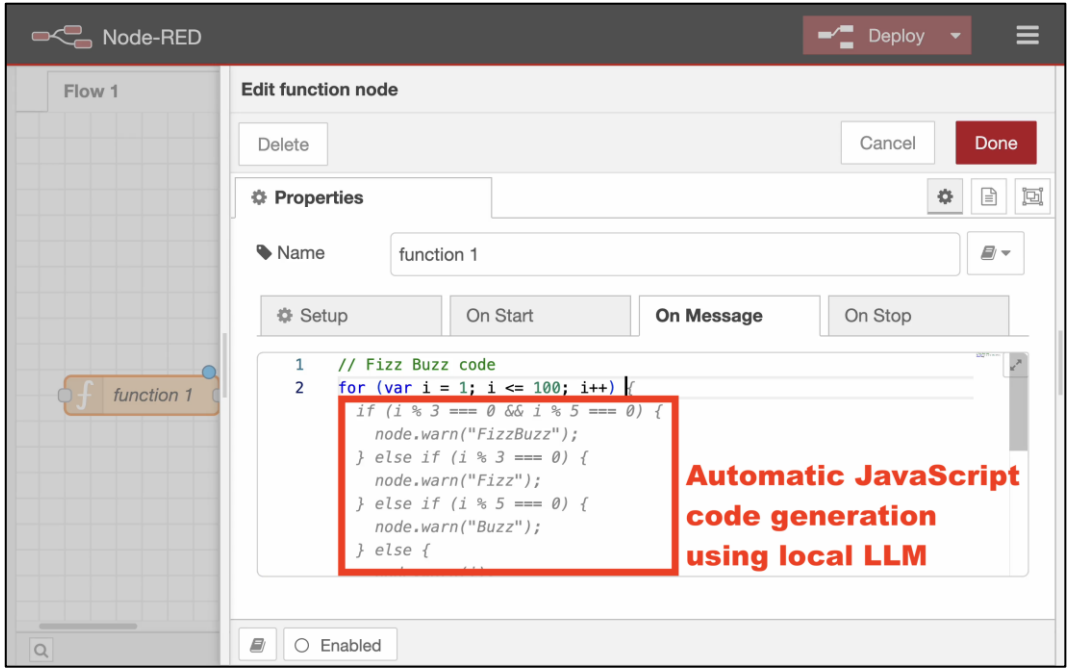
Code Completion Plugin for Node-RED

Code Completion Plugin for Node-RED

HITACHI
Inspire the Next

Plugin to enable code completion in code editor of function node

- Provides the same UX as code completion in GitHub Copilot-enabled VS Code
- Generates JavaScript code from comments using Granite Code Model
- Available as an OSS on Node-RED library



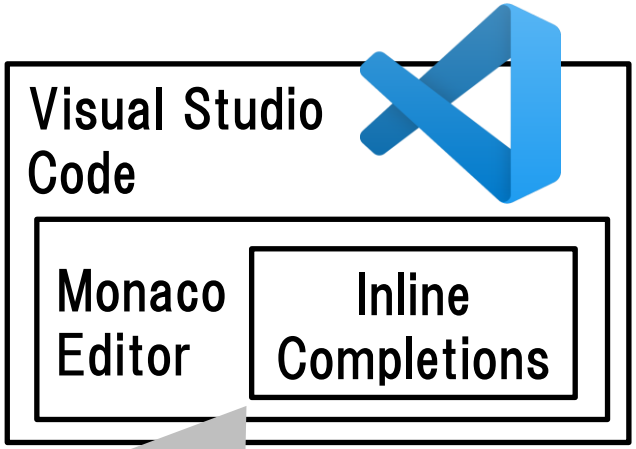
Code editor of function node

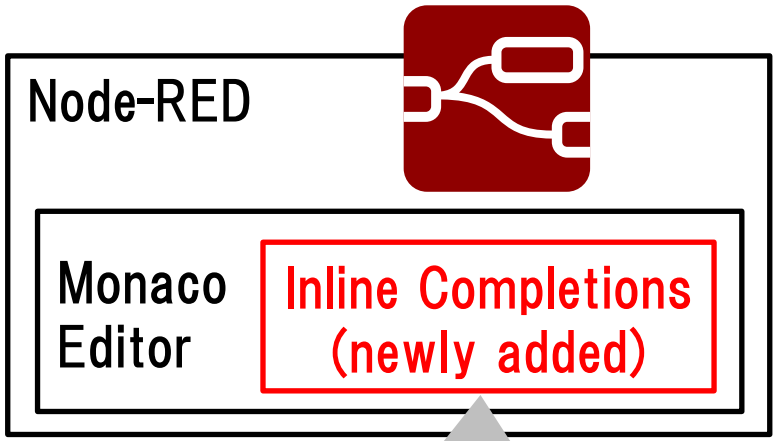
https://flows.nodered.org/node/@kazuhitoyokoi/node-red-contrib-plugin-codecompletion

Code Completion UX in Node-RED

HITACHI
Inspire the Next

Node-RED can be applied for the same UX as Visual Studio Code, because they both have the same Monaco Editor.





[UX in GitHub Copilot]
- Code Suggestion
- Accepting code by tab key

Apply for the same UX for Node-RED

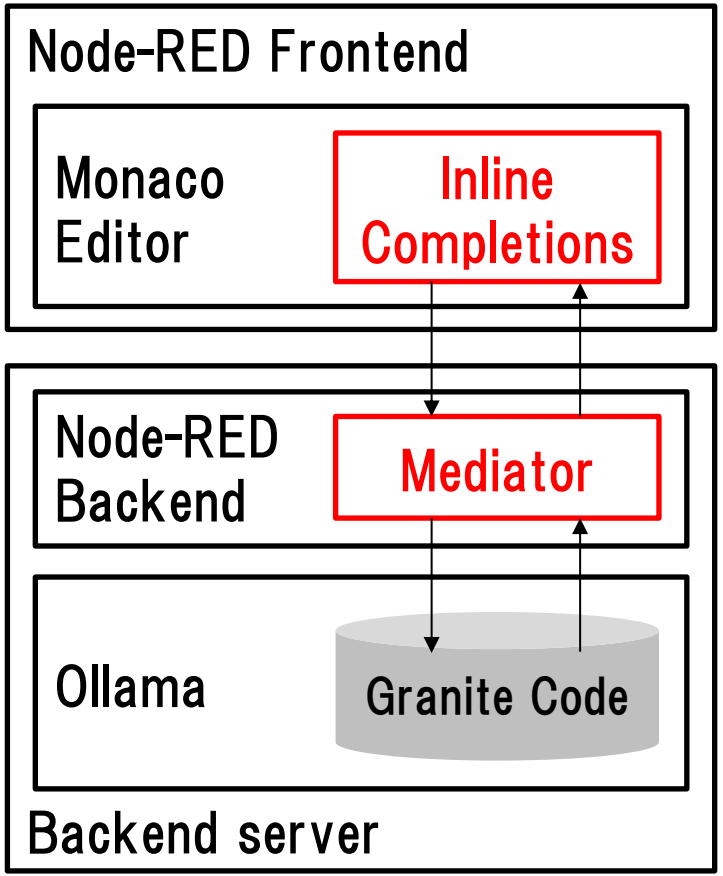
https://microsoft.github.io/monaco-editor/typedoc/functions/languages.registerInlineCompletionsProvider.html

Architecture of Code Completion Plugin

HITACHI
Inspire the Next

Using the Node-RED plugin feature,
the UI and backend components are installed.

- Inline Completions UI
  - Provides code suggestions UX
  - Accesses Mediator endpoint which has the same port as Node-RED backend
- Mediator
  - Provides REST API with authentication for Inline Completions UI
  - Accesses to the Ollama endpoints within the same backend server



How to Install Code Completion Plugin

HITACHI
Inspire the Next

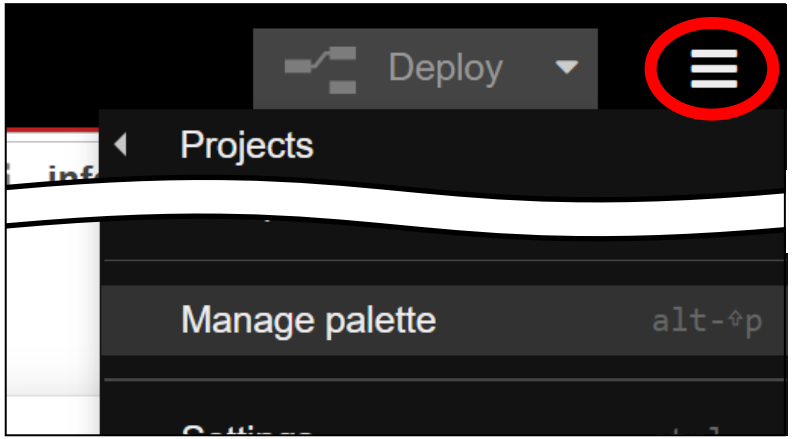
Node-RED has a UI for installing plugins in the browser.

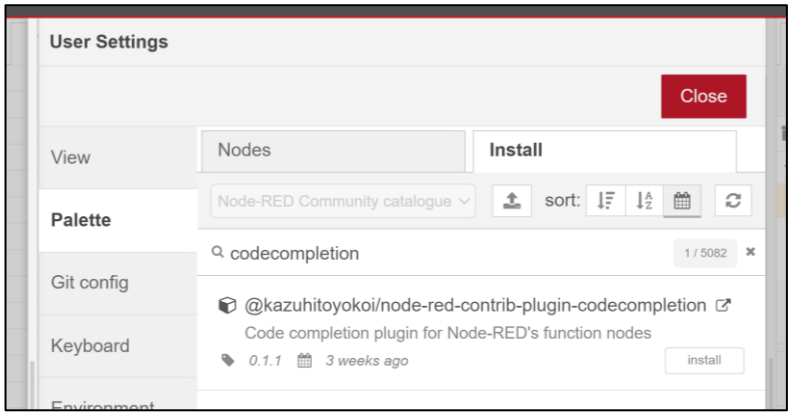
(1) Select "Manage Palette" from the menu

(2) Open the "Install" tab
in the user settings UI

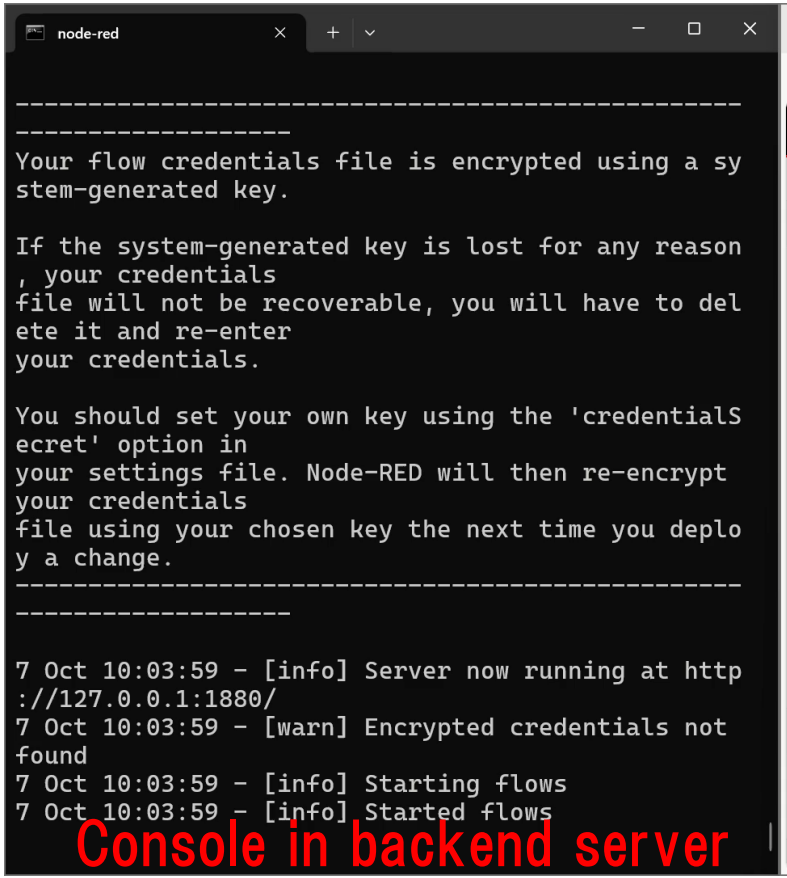
(3) Type "codecompletion"
in the search field

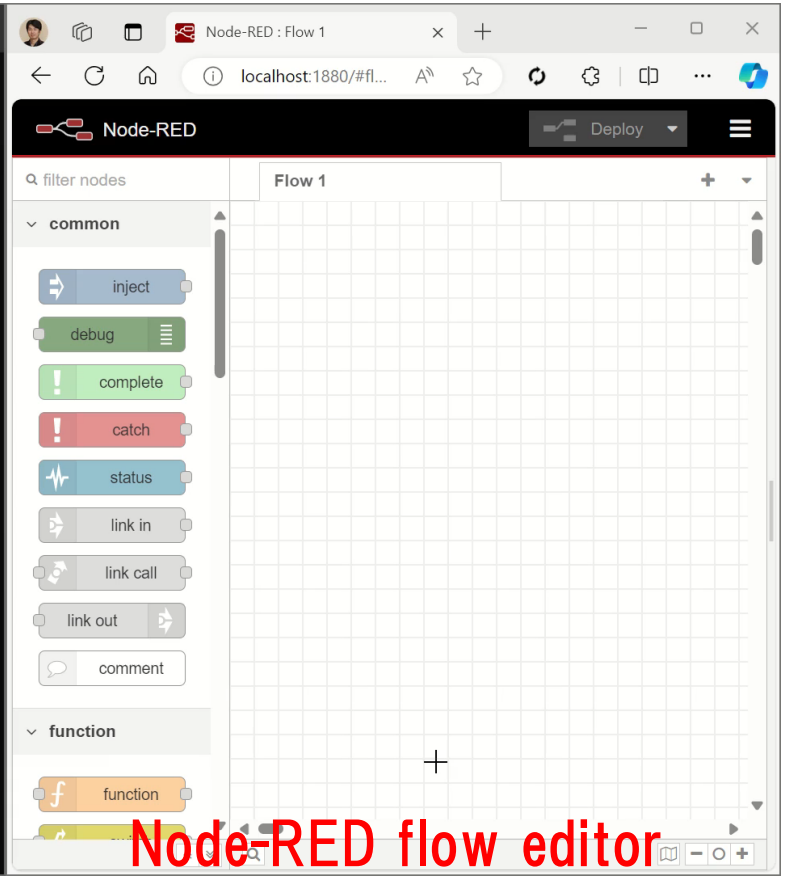
(4) Click the install button of the
"node-red-contrib-plugin-codecompletion"





Downloading Granite Code Model





HITACHI
Inspire the Next

Demonstrations

Demonstration 1: Generating Fizz Buzz Code

HITACHI
Inspire the Next

(1) Typing the comments and loop block

```
1   // Fizz Buzz code
2   for (var i = 1; i <= 100; i++)
3   return msg;
```

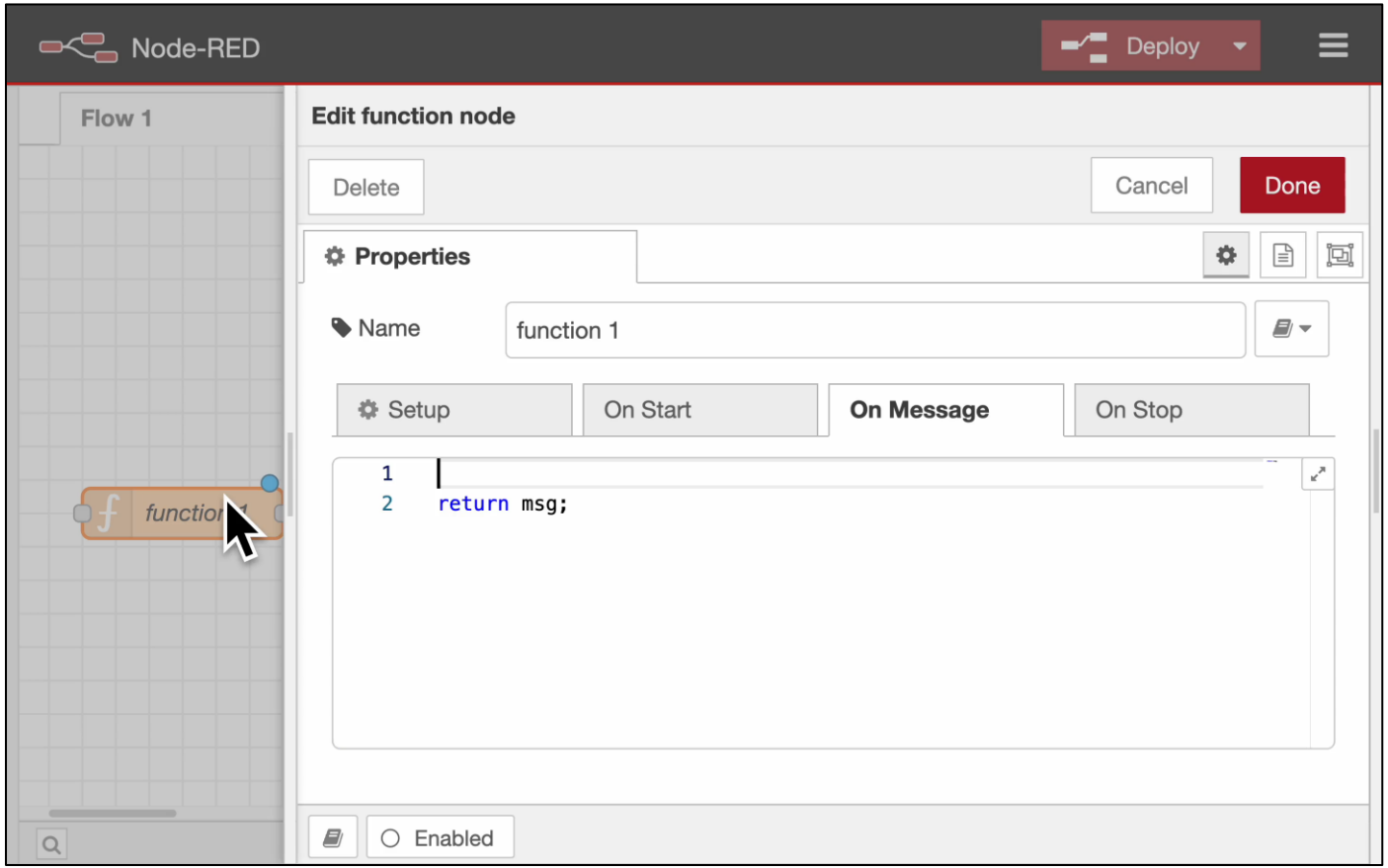
(2) Fizz Buzz Code is suggested in gray text

```
1   // Fizz Buzz code
2   for (var i = 1; i <= 100; i++) {
      if (i % 3 === 0 && i % 5 === 0) {
        node.warn("FizzBuzz");
      } else if (i % 3 === 0) {
        node.warn("Fizz");
      } else if (i % 5 === 0) {
        node.warn("Buzz");
      } else {
        node.warn(i);
```

(3) Applying suggested code by tab key

```
1   // Fizz Buzz code
2   for (var i = 1; i <= 100; i++) {
3     if (i % 3 === 0 && i % 5 === 0) {
4       node.warn("FizzBuzz");
5     } else if (i % 3 === 0) {
6       node.warn("Fizz");
7     } else if (i % 5 === 0) {
8       node.warn("Buzz");
9     } else {
10      node.warn(i);
```

Demonstration 1: Generating Fizz Buzz Code



Demonstration 2: Calculating Distance between Two Locations

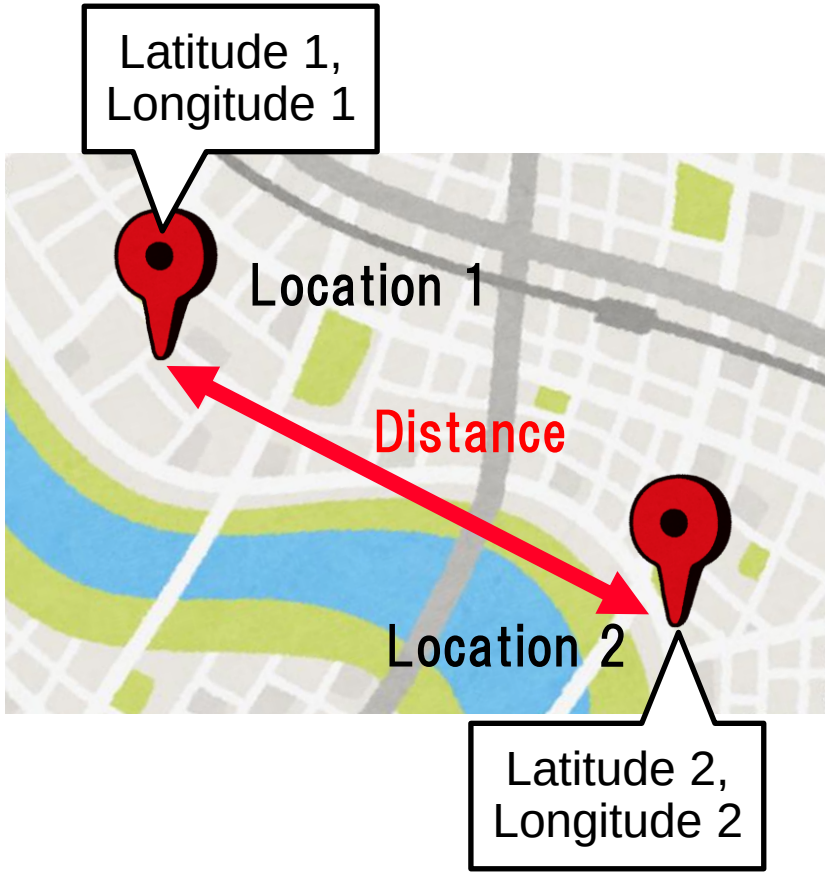
HITACHI
Inspire the Next

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 :(

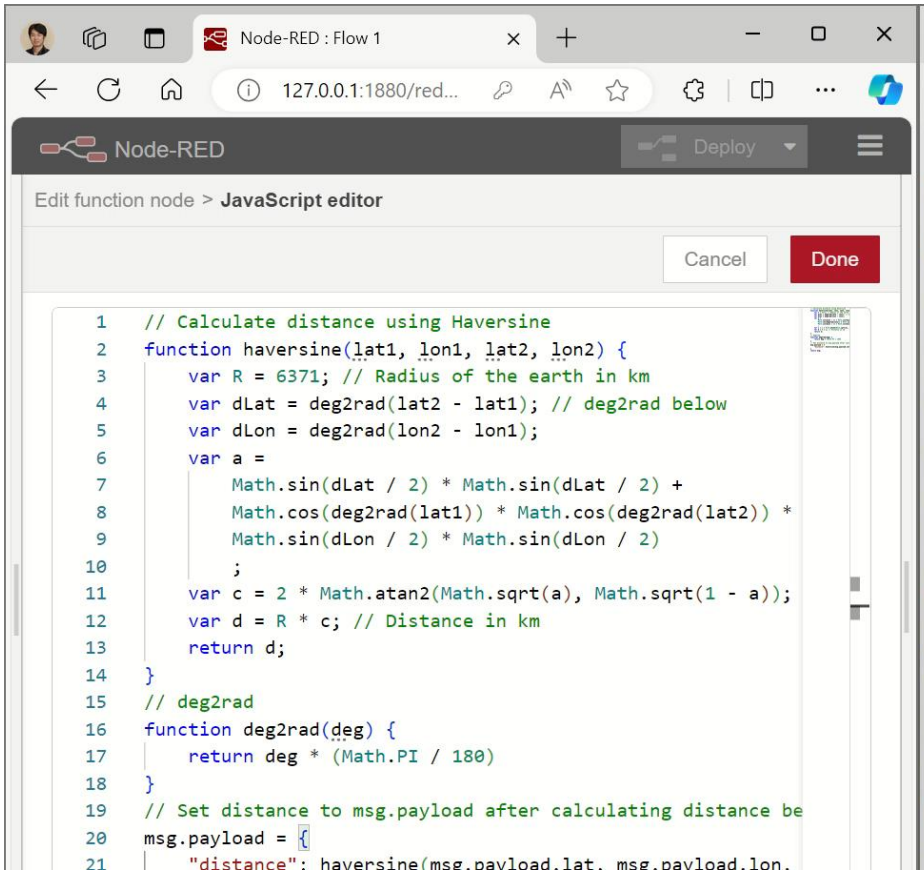


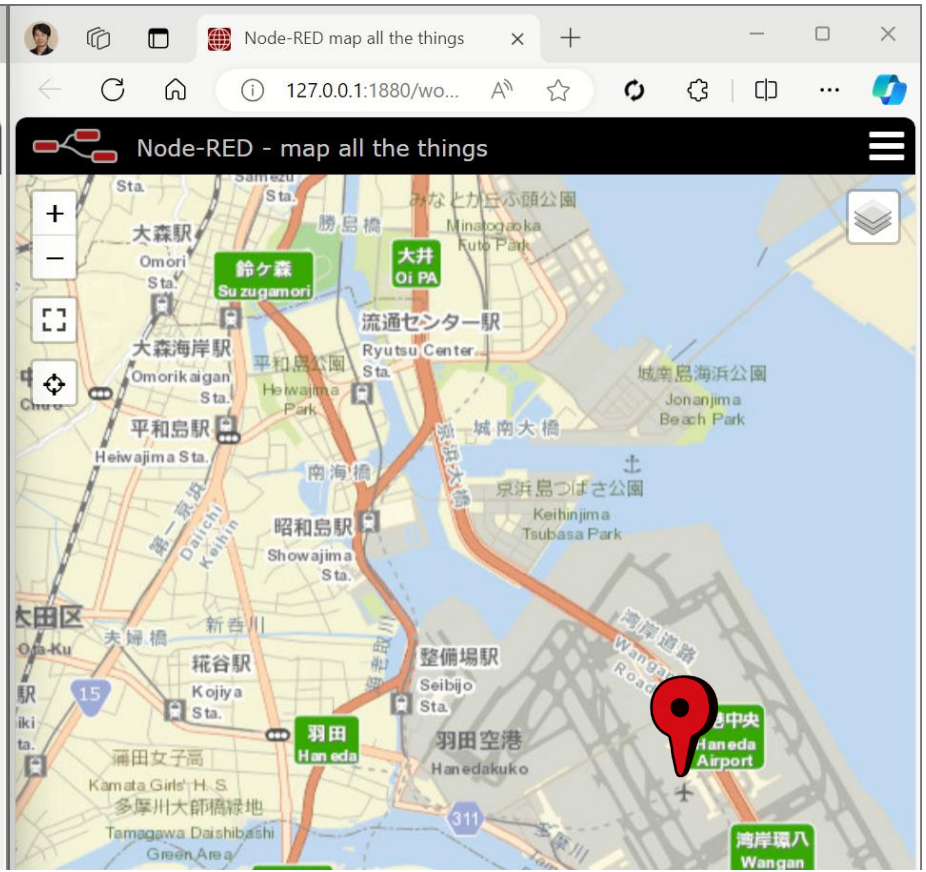
Demonstration 2: Calculating Distance between Two Locations

HITACHI
Inspire the Next

Demonstration 2: Calculating Distance between Two Locations

HITACHI
Inspire the Next



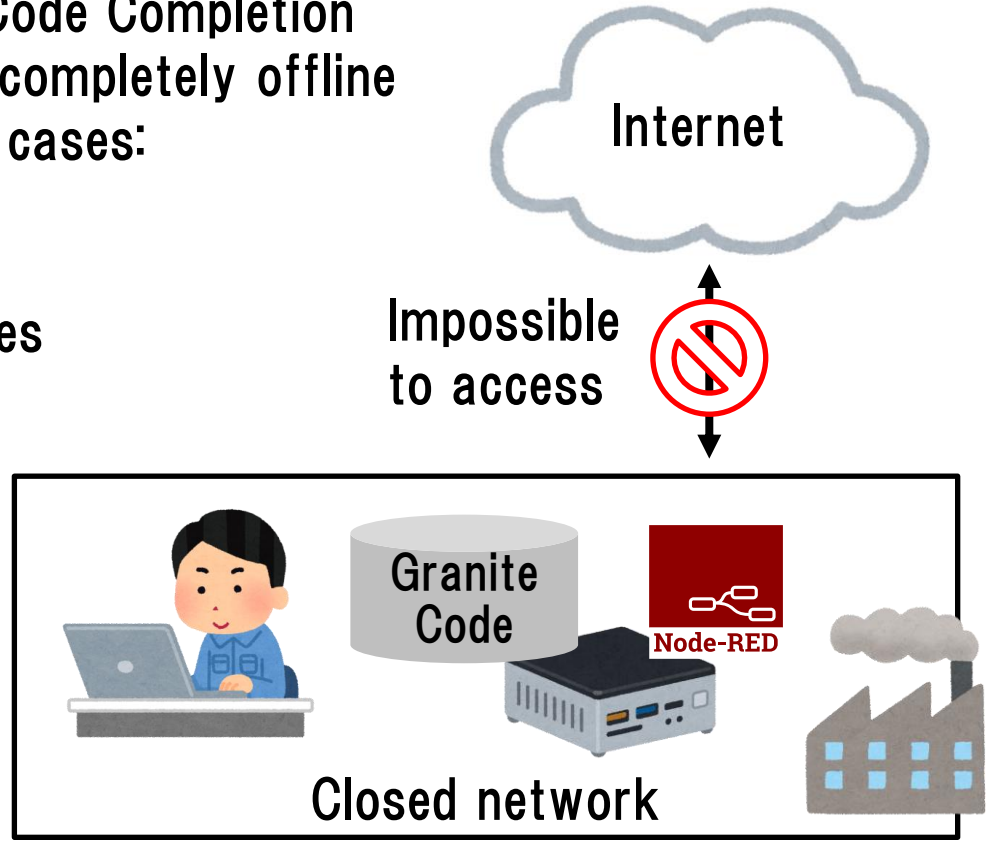


Benefits of Offline Use

HITACHI
Inspire the Next

The combination of Node-RED with Code Completion and the Granite Code Model enables completely offline development that is suitable for use cases:

- Mission-critical systems under the strict compliance policies
- Factory automation systems inside a closed network



HITACHI
Inspire the Next

Conclusion

Conclusion

HITACHI
Inspire the Next

Coding with LLM increases productivity in Node-RED.

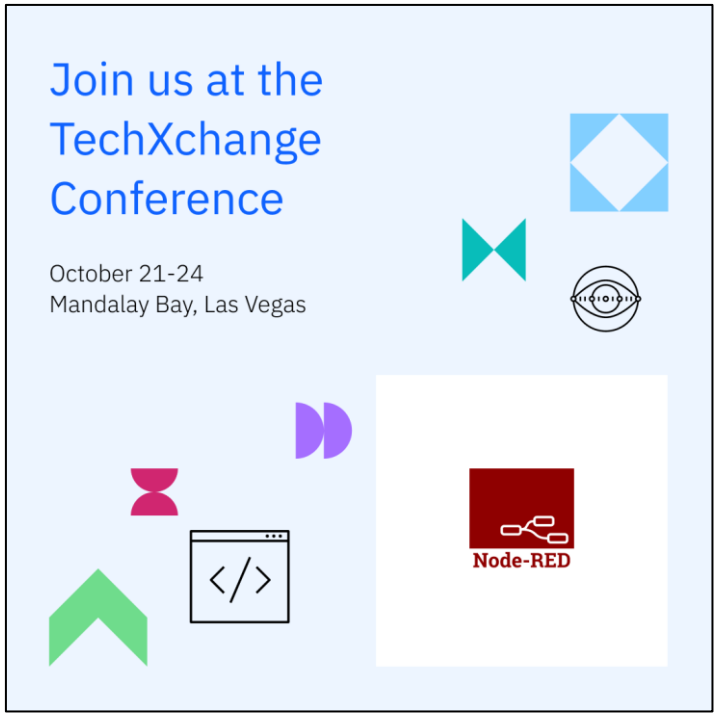
- Code Completion plugin provides the latest developer experience for Node-RED.
- In addition, the Granite code model enables offline use and reduces compliance risks.

Please look forward to the future developments.

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

Boosting Productivity of Node-RED with Large Language Model

October 21, 2024

Kazuhito Yokoi
Hitachi Academy

Trademarks

HITACHI
Inspire the Next

- Node-RED, Node.js, and Node Version Manager are trademarks of OpenJS Foundation.
- reTerminal DM is a trademark of Seeed Studio.
- npm is a trademark of npm, Inc.
- GitHub Copilot is a trademark of GitHub, Inc.
- Granite code, code engine and watsonx.ai are trademarks of International Business Machines Corporation.
- ThinkPad is a trademark of Lenovo Corporation.
- macbook is a trademark of Apple Incorporated.
- Visual Studio code and monaco editor are trademarks of Microsoft Corporation.
- ChatGPT is a trademark of OpenAI, Inc.
- Claude is a trademark of Anthropic PBC.
- Other company, product or service names may be trademark or registered trademark of others.