

2025



Lesson 07

Low Code LLM App Builder (Part 1)



Flowise

Introduction

- Learn to develop low code LLM application using open-source low-code tool.
- Many tools are available. They are: 1) Flowise, 2) LLMStack, 3) Superagent, 4) Langflow, 5) Dify etc.
- Flowise AI is a low-code/no-code platform that simplifies the development of application powered by Large Language Models (LLMs).
- Flowise provides a visual and user-friendly drag-and-drop GUI interface, making it accessible to both technical and non-technical users.
- Individuals with little programming experience can effortlessly create these LLM applications without the need to write any code.
- Advantageous for organizations striving to rapidly build prototypes and develop LLM applications in an agile manner.

Flowise & LLM Frameworks

- The LangChain/LlamaIndex frameworks made it easy to build LLM applications.
- By abstracting objects like chains, this framework gives us the power to compose complex workflow to solve interesting tasks like Chatbots over documents, personal assistants, semantic search engines etc.
- Framework requires you to code in Python or JavaScript programming language.
- Flowise goes further (together with LangChain/LlamaIndex) by providing an interactive User Interface (UI).





Flowise / Hugging Face Space

Flowise Installation (Hugging Face)

- Familiarity with installation procedures fosters confidence in handling the software, boosting your technical proficiency and adaptability.
- Many ways to install/deploy Flowise (Chatflow).
- We will be using **Hugging Face** to host Flowise.



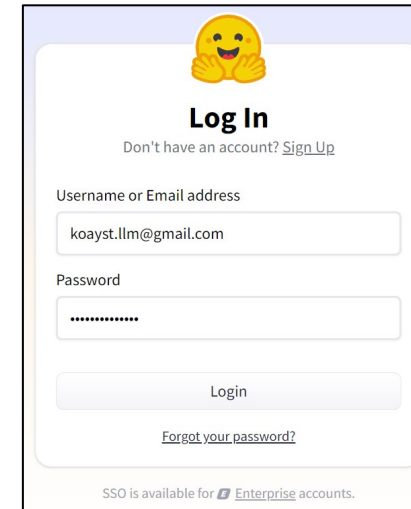
Hugging Face



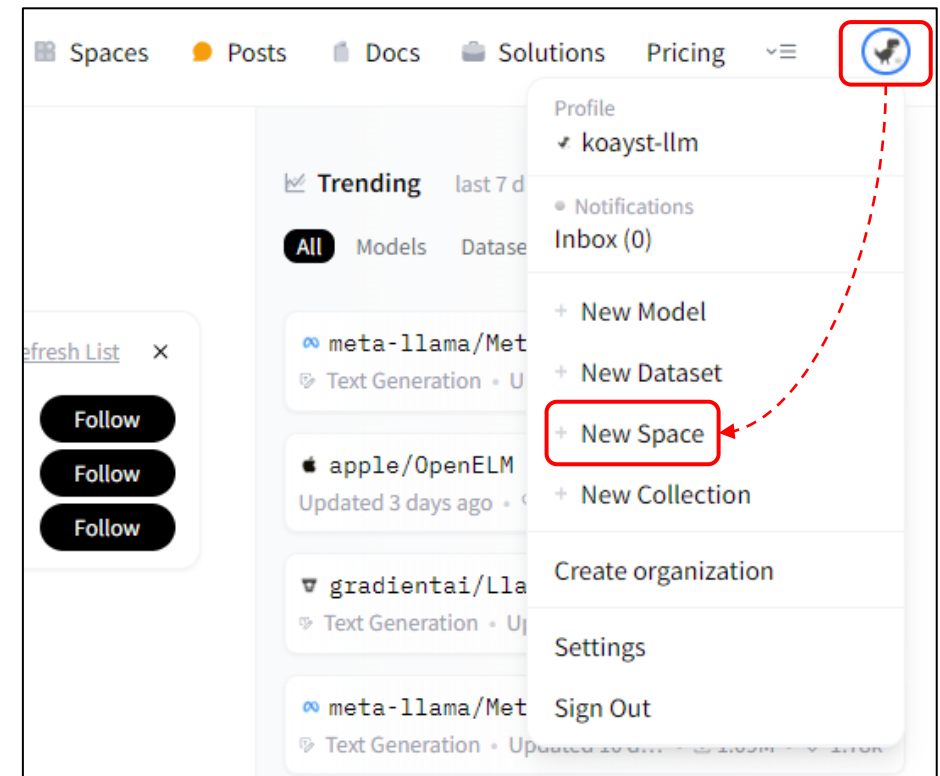
Login Hugging Face

- Login in to Hugging Face.
<https://huggingface.co/login>
- Sign up a Hugging Face account if you don't have one. It's FREE !
- Follow the instruction as published in Flowise to create "Space" to run Flowise.

<https://docs.flowiseai.com/configuration/deployment/hugging-face>



The image shows the Hugging Face login page. At the top is a yellow emoji icon with its hands clasped. Below it is the text "Log In" and a link "Don't have an account? Sign Up". There are two input fields: "Username or Email address" with the value "koayst.llm@gmail.com" and "Password" with masked characters. A "Login" button is below the password field, and a link "Forgot your password?" is to its right. At the bottom, it says "SSO is available for Enterprise accounts."



Step 1

Instruction:
Read from left
to right, top to
bottom

1

Create a
new
space

Hugging Face Search models, data Models Datasets Spaces Docs Enterprise Pricing

Create a new Space

Spaces are Git repositories that host application code for Machine Learning demos. You can build Spaces with Python libraries like [Streamlit](#) or [Gradio](#), or using [Docker](#) images.

Owner koayst-llm / **Space name** flowise

Short description Short Description

License unknown

Select the Space SDK
You can choose between Streamlit, Gradio and Static for your Space. Or [pick Docker](#) to host any other app.

Streamlit Gradio 3 templates **Docker 15 templates** Static 3 templates

Select the Space SDK
You can choose between Streamlit, Gradio and Static for your Space. Or [pick Docker](#) to host any other app.

Streamlit Gradio 3 templates **Docker 13 templates** Static 3 templates

Choose a Docker template:

Blank JupyterLab Argilla Livebook

LabelStudio AimStack AutoTrain Shiny (R)

Shiny (Python) ZenML ChatUI Panel

Giskard Quarto

Space hardware Free

CPU basic · 2 vCPU · 16 GB · FREE

You can switch to a different hardware at any time in your Space settings. You will be billed for every minute of uptime on a paid hardware.

Public
Anyone on the internet can see this Space. Only you (personal Space) or members of your organization (organization Space) can commit.

Private
Only you (personal Space) or members of your organization (organization Space) can see and commit to this Space.

Create Space

Step 2

2

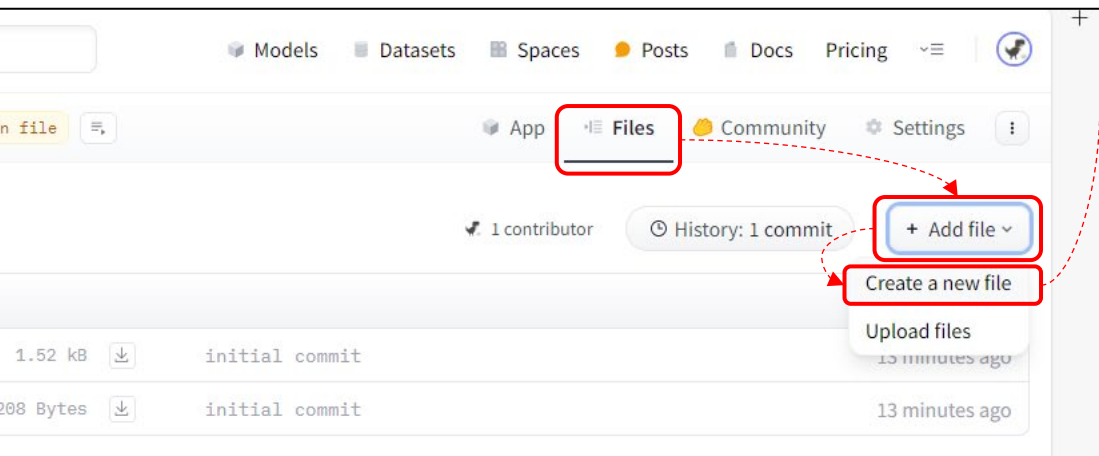
Set the environment variables

The screenshot shows the Hugging Face Spaces interface for a Space named 'flowise'. The 'Settings' tab is selected, and the 'Variables and secrets' section is expanded. A 'New variable' dialog box is open, showing the 'Name' field set to 'PORT' and the 'Value (public)' field set to '7860'. The 'Save' button is highlighted. Red dashed arrows indicate the flow of the process: from the 'Settings' tab to the 'Variables and secrets' section, and from the 'New variable' dialog box to the 'Variables and secrets' section. The 'Variables and secrets' section shows a list of variables, including 'PORT' with a 'View' and 'Delete' button. The 'Factory rebuild' button is also visible.

Instruction:
Read from left to right, top to bottom

3

Create a Dockerfile



Go to:

<https://docs.flowiseai.com/configuration/deployment/hugging-face> to copy the Dockerfile text

Create a Dockerfile

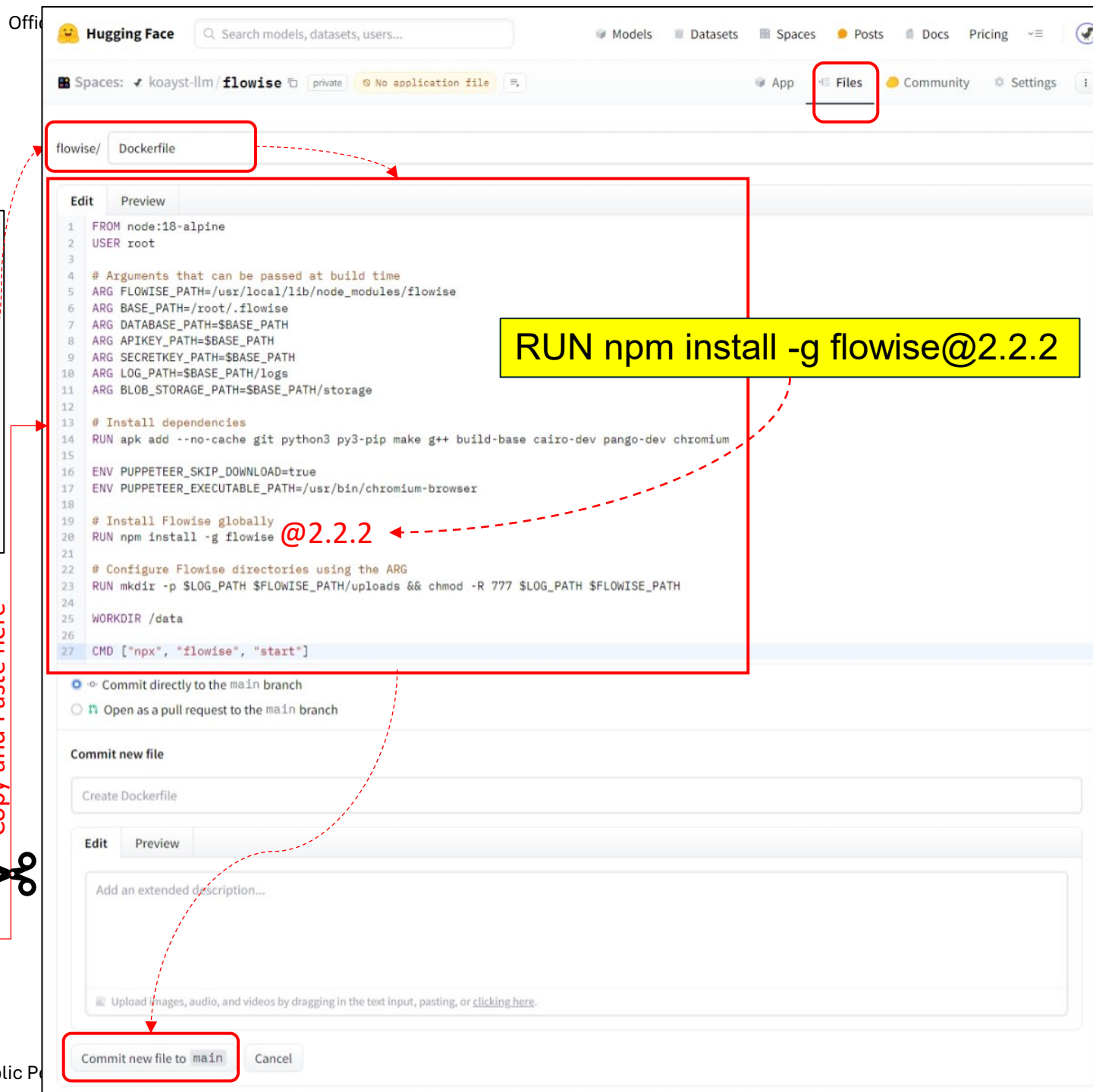
- At the files tab, click on button **+ Add file** and click on **Create a new file** (or Upload files if you prefer to)
- Create a file called **Dockerfile** and paste the following:

```
FROM node:18-alpine
USER root

# Arguments that can be passed at build time
ARG FLOWISE_PATH=/usr/local/lib/node_modules/flowise
ARG BASE_PATH=/root/.flowise
ARG DATABASE_PATH=$BASE_PATH
ARG APIKEY_PATH=$BASE_PATH
ARG SECRETKEY_PATH=$BASE_PATH
```

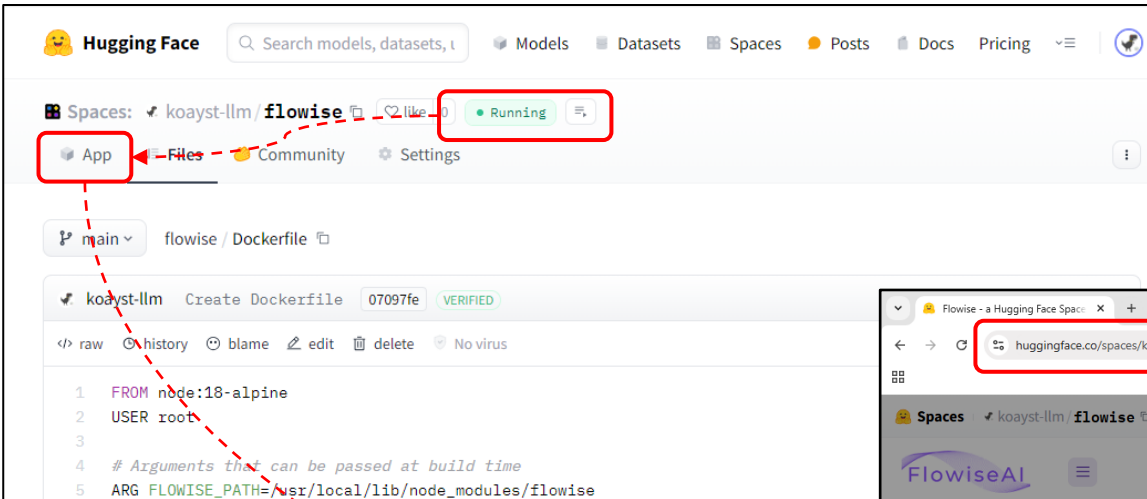
Copy

Copy and Paste here

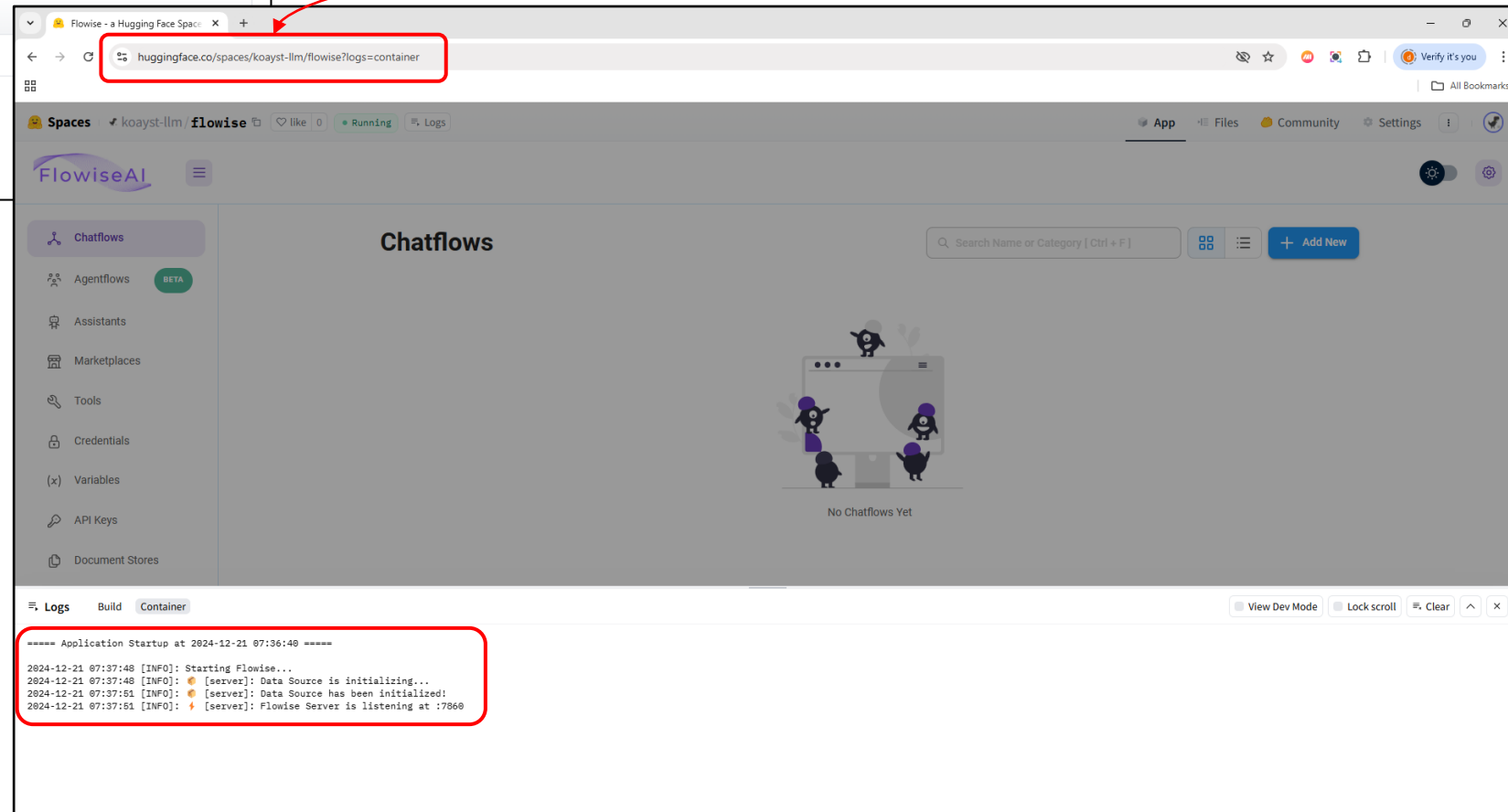


Step 4: Running Flowise

Take note of this URL. You can share it with others to use your Flowise




Be patient: Once Flowise starts running, click the “grey” area to start using Flowise



Delete Space

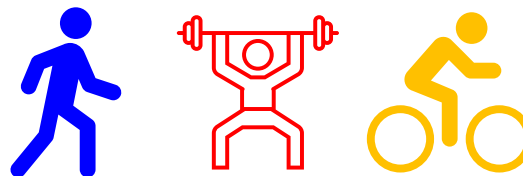
- Once you are done with Flowise, a good practice is to delete the Space from Hugging Face.
- Go to Setting of your Flowise-Space, search for your Space, scroll to the end and type the “word” to confirm the deletion.

 **Delete this space**

This action **cannot** be undone. This will permanently delete the **koayst-llm/flowise** space repository and all its files.

Please type **koayst-llm/flowise** to confirm.

[I understand, delete this space](#)





Installation

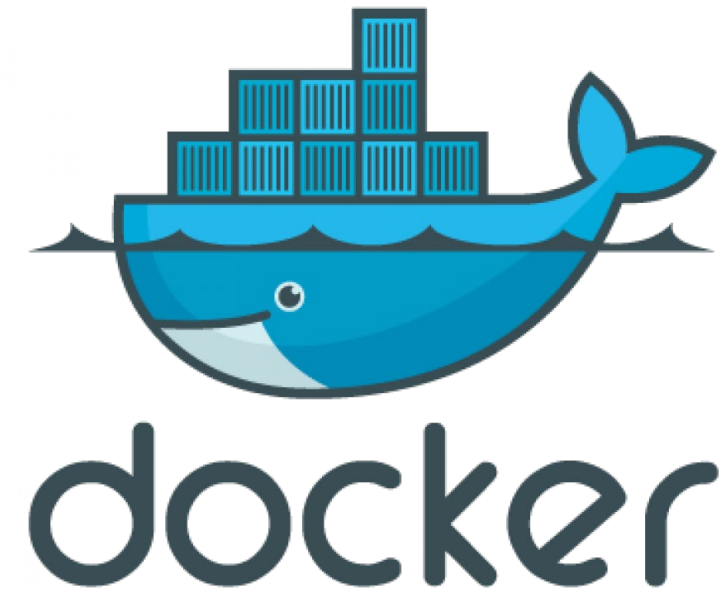
Laptop

(optional)

Installation and Setup

- Prerequisite
 - Install latest NodeJS.
- Setup
 - **Method 1:**
 - Install Flowise via npm.
 - Start Flowise.
 - Method 2:
 - Git clone the open-source project.
 - Docker Compose
 - Docker-compose up the local “.env” to run Flowise
 - Docker-compose stop to stop running Flowise
 - Docker Image
 - Build and Run Docker image

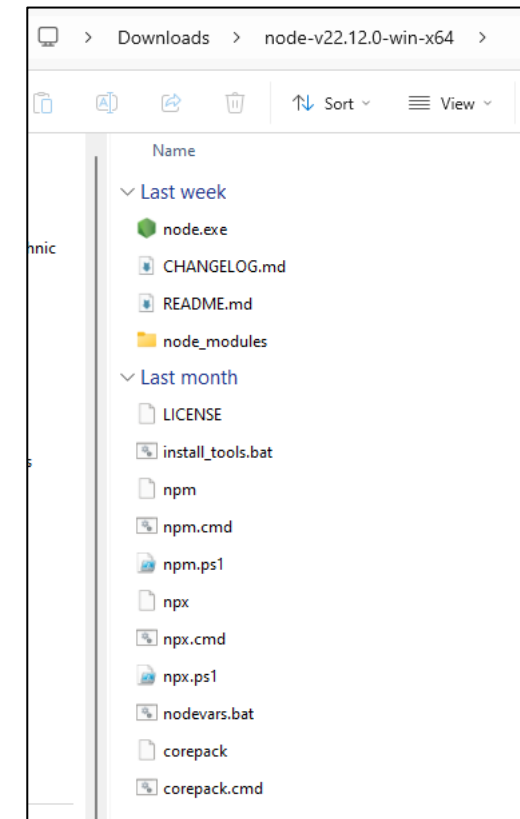
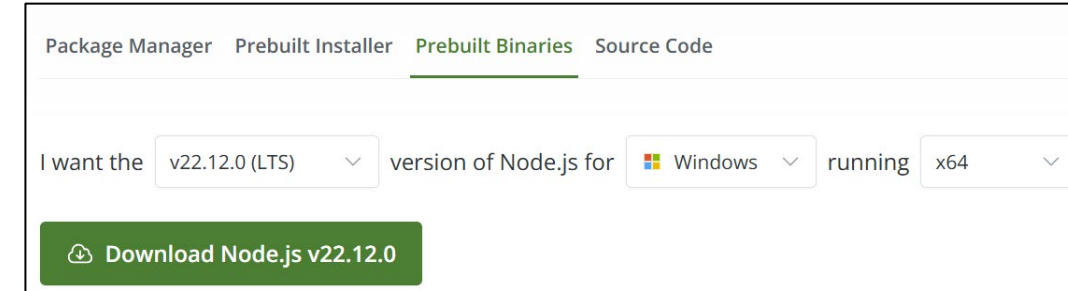
Reference: <https://docs.flowiseai.com/getting-started>



Installation & Setup

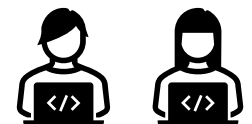
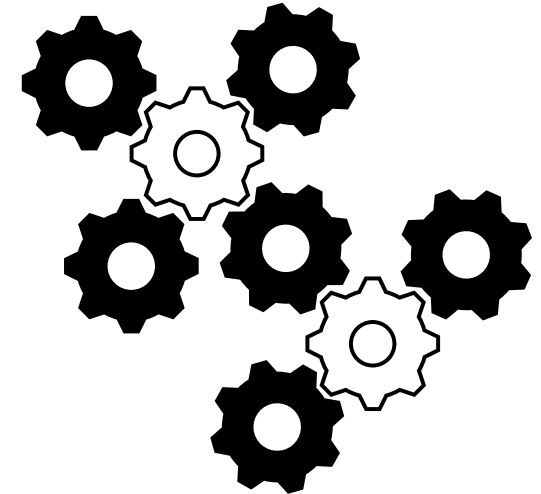
The steps listed are meant for Windows OS

- 1) Download the latest prebuilt binary NodeJS®:
 - v22.12.0 (LTS)
<https://nodejs.org/en/download/prebuilt-binaries>
 - 2) Unzip the just downloaded file to a directory: node-v22.12.0-win-x64.
 - 3) Open a Command Prompt and navigate to the directory: node-v22.12.0-win-x64.
- LTS (Long Term Support) version v20.15.1 is recommended.
 - For production environments where stability and compatibility are crucial, the LTS version provides a reliable choice. The current version offers the latest features and improvements for developers who want to explore new functionalities.



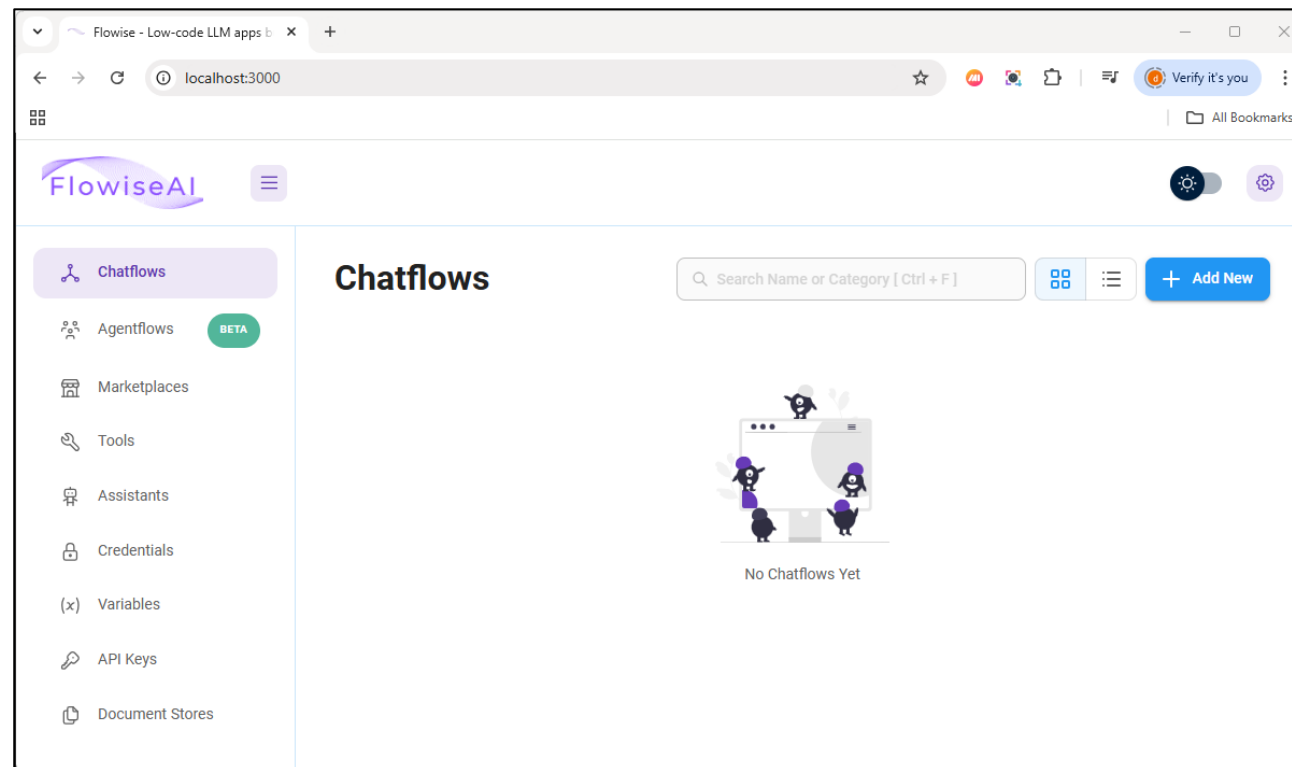
Installation & Setup

- 1) Assuming you have unzipped the zip file to directory C:\Users\<Your_username>\Downloads\node-v22.12.0-win-x64, set the environment variable by:
 - `set PATH=C:\Users\Your_username\Downloads\node-v22.12.0-win-x64;%PATH%`
- 2) Install Flowise
 - `npm install -g flowise@2.2.1`
 - `npm fund (optional)`
- 3) Start Flowise (default port is 3000)
 - `npx flowise start --DEBUG=true`
- 4) To keep Flowise updated
 - `npm update -g flowise`
- 5) To Uninstall Flowise when you are done with using Flowise
 - `npm uninstall -g flowise`



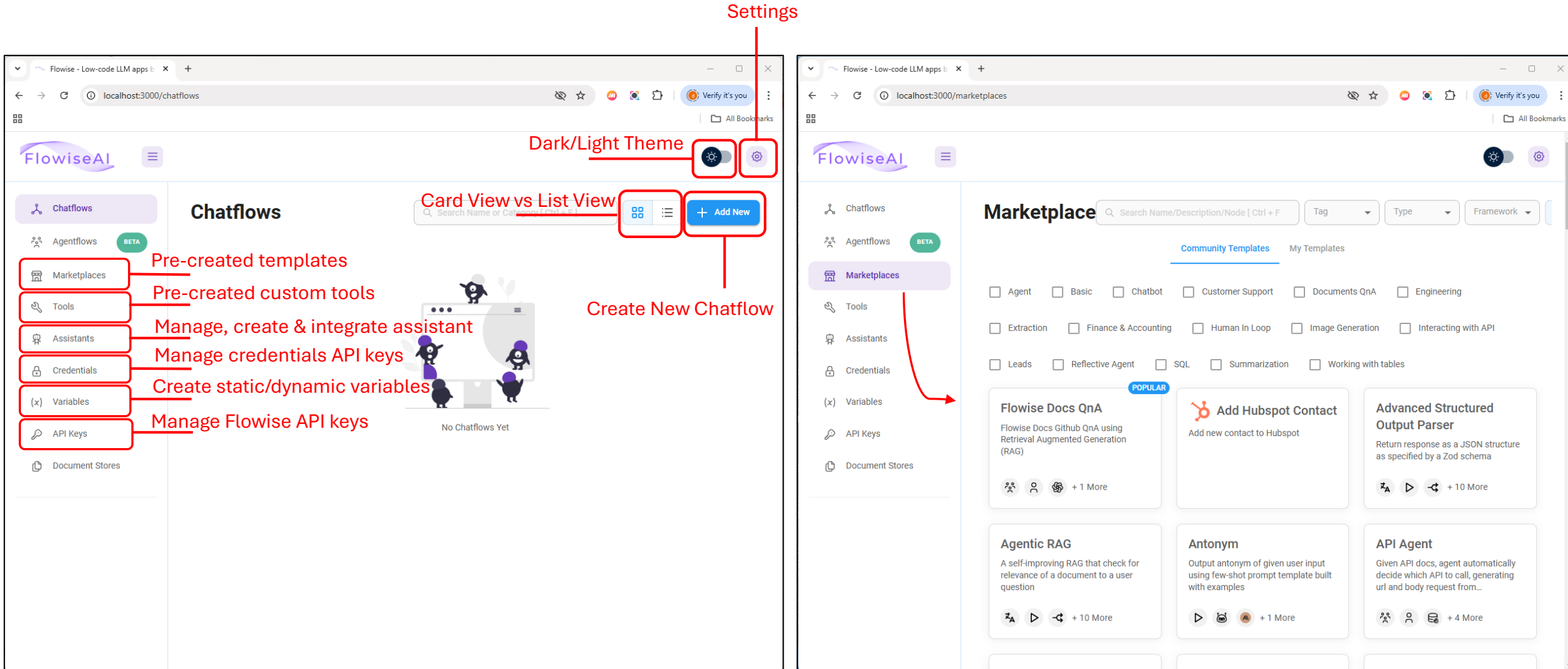
Installation Test

- Open `http://localhost:3000` with a browser
- **CTRL-C** at Command Prompt to shutdown Flowise



```
C:\Users\<Your_Username>\Downloads\node-v22.12.0-win-x64>npx flowise start --DEBUG=true
2024-12-09 14:46:12 [INFO]: Starting Flowise...
(node:3344) [DEP0040] DeprecationWarning: The `punycode` module is deprecated. Please use a userland alternative
instead.
(Use `node --trace-deprecation ...` to show where the warning was created)
2024-12-09 14:46:13 [INFO]: ⚡ [server]: Flowise Server is listening at 3000
2024-12-09 14:46:13 [INFO]: 📦 [server]: Data Source is being initialized!
2024-12-09 14:49:01 [INFO]: 📦 [server]: Data Source has been initialized!
2024-12-09 14:53:18 [INFO]: Shutting down Flowise...
Terminate batch job (Y/N)? y
```

Flowise Dashboard



Settings

Dark/Light Theme

Card View vs List View

Create New Chatflow

Pre-created templates

Pre-created custom tools

Manage, create & integrate assistant

Manage credentials API keys

Create static/dynamic variables

Manage Flowise API keys

Marketplace

Flowise Docs QnA

Add Hubspot Contact

Advanced Structured Output Parser

Agentic RAG

Antonym

API Agent

Accounts & API Keys

- To facilitate your learning:
 - You need to possess some API keys to conduct your experiment.
 - Sometimes you will need to register for an account with other service provider to obtain an API key to test run or complete your app development.
 - It will be easier if you have a Google account to register during account registration.
 - Please kindly handle the API keys with care.
 - Do not share your API keys with others.






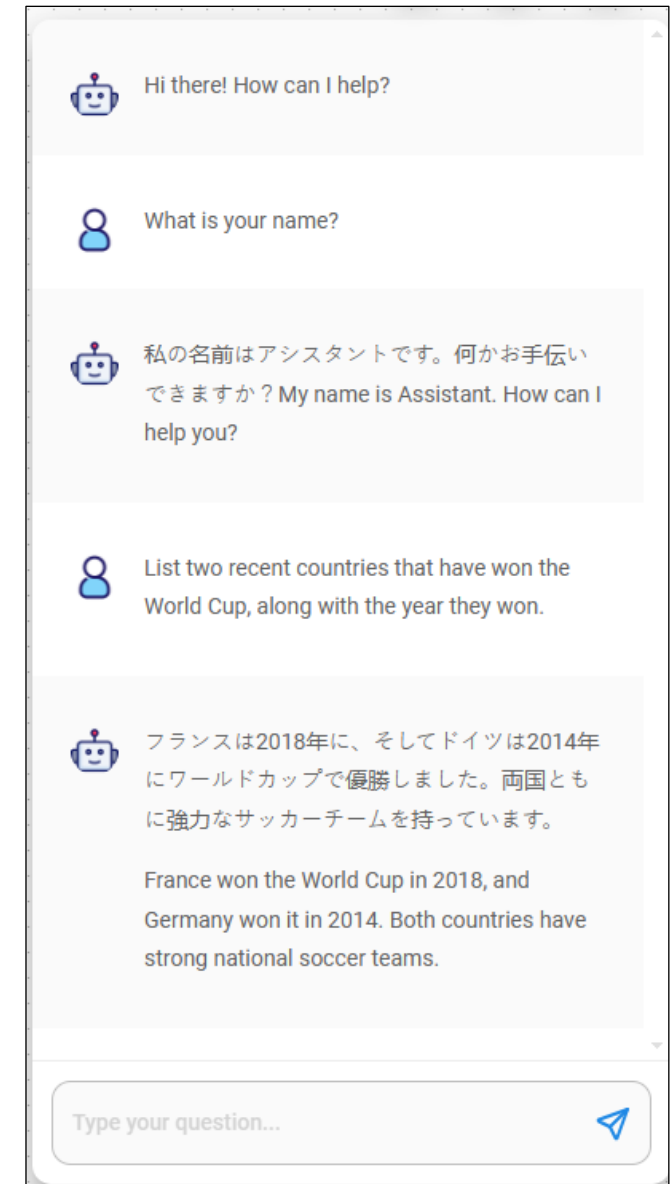
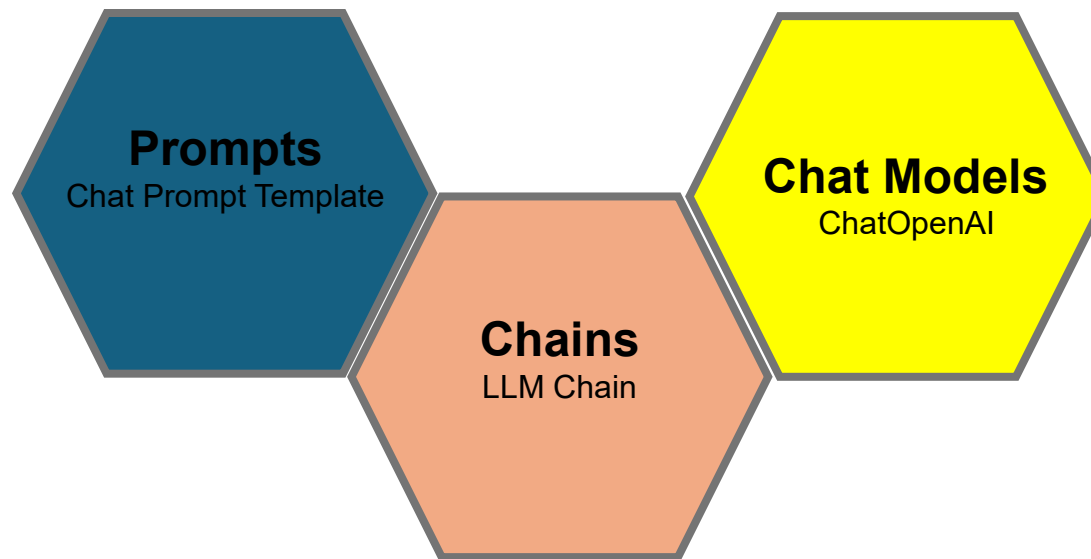
Activity: Translator

Activity: Translator

- Flowise was successfully installed on your computer/laptop.
- Let's create our first LLM application using Flowise.
- Familiarise ourselves with the Flowise UI.
- Learn how to use Flowise components to build a simple Chatbot application.
- Experiment by asking questions in English and receiving responses in both English and Japanese sentences.
- You are encouraged to explore switching to different languages once the basic functionality is understood and running properly.

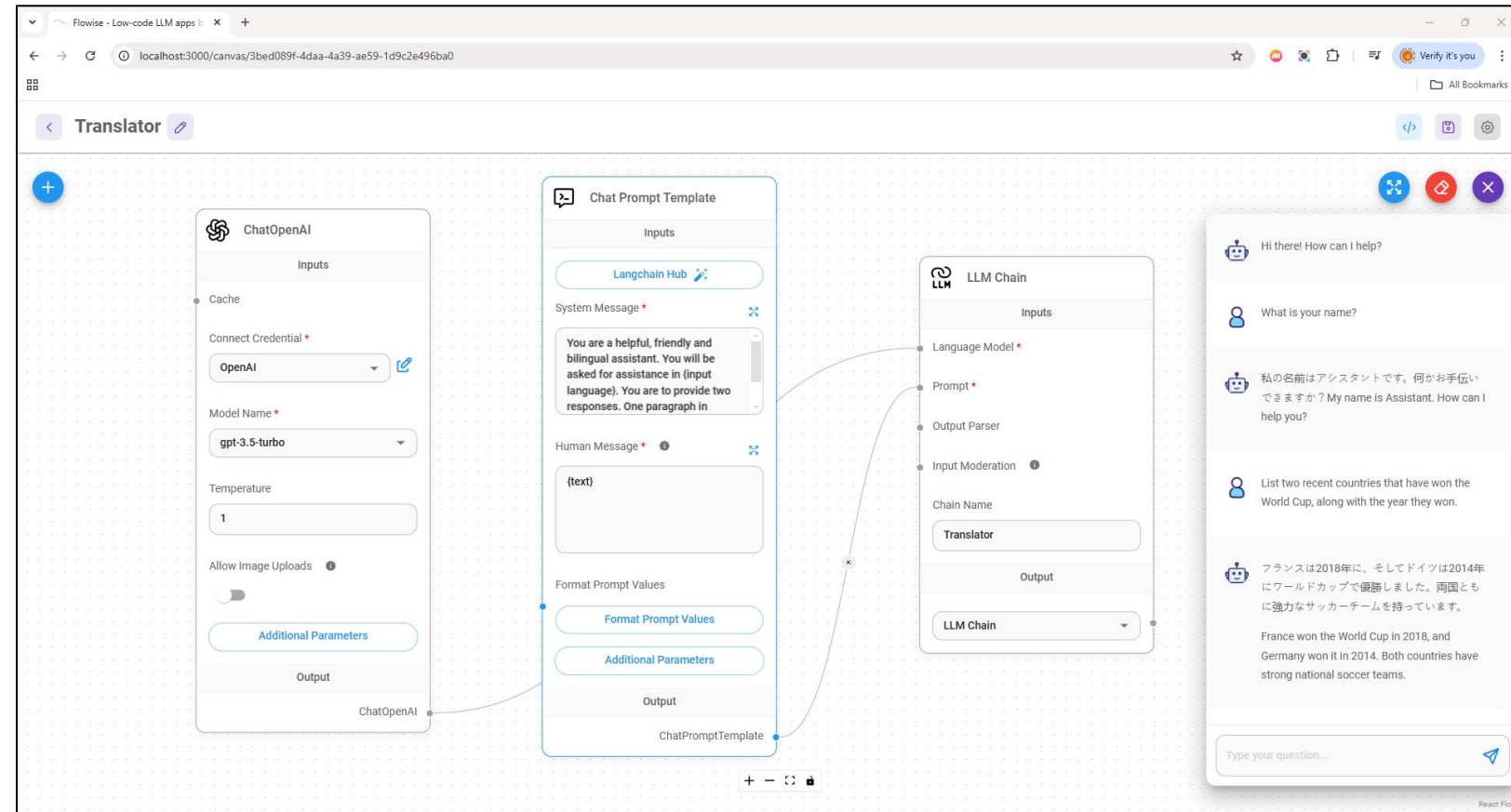
Activity Map: Translator

- You will need the following:
 - ☒ OpenAI API Key [ OpenAI]
 - ☒ cURL [`curl://`]
- Flowise setup








First LLM Application

- Let's build our first AI LLM application using Flowise.
- This exercise aims to familiarise you with Flowise UI.
- We will build a simple English-to-Japanese translator.
- Feel free to try another language after you completes the exercise.

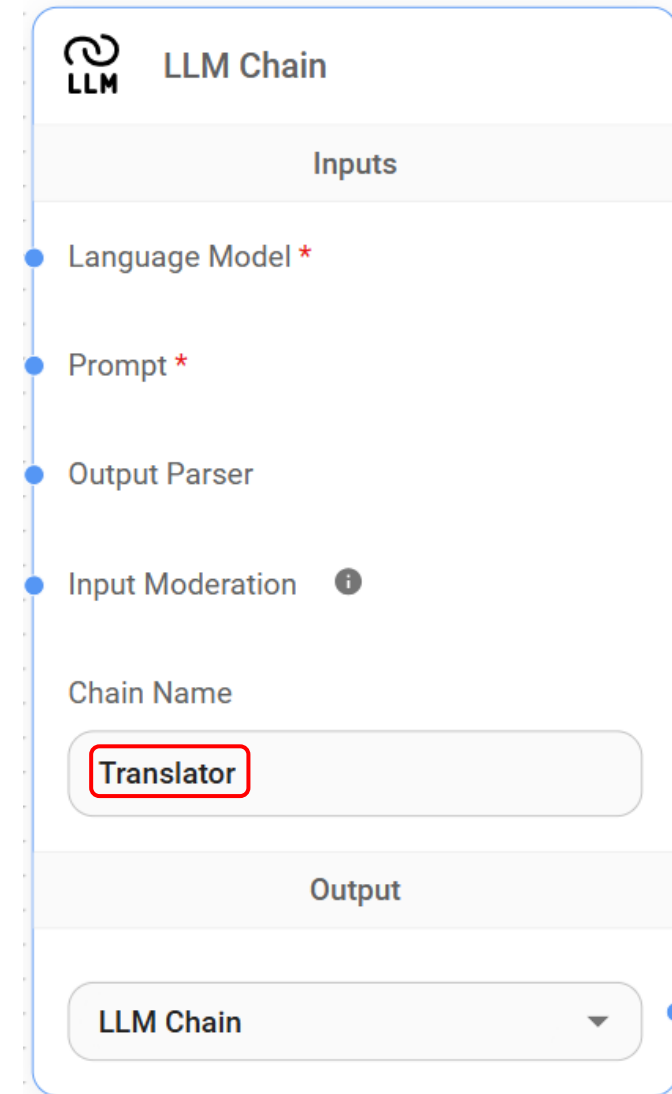
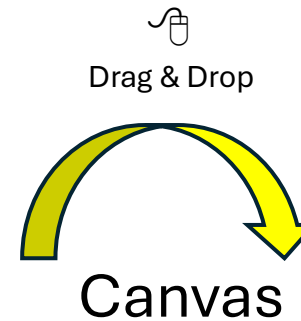
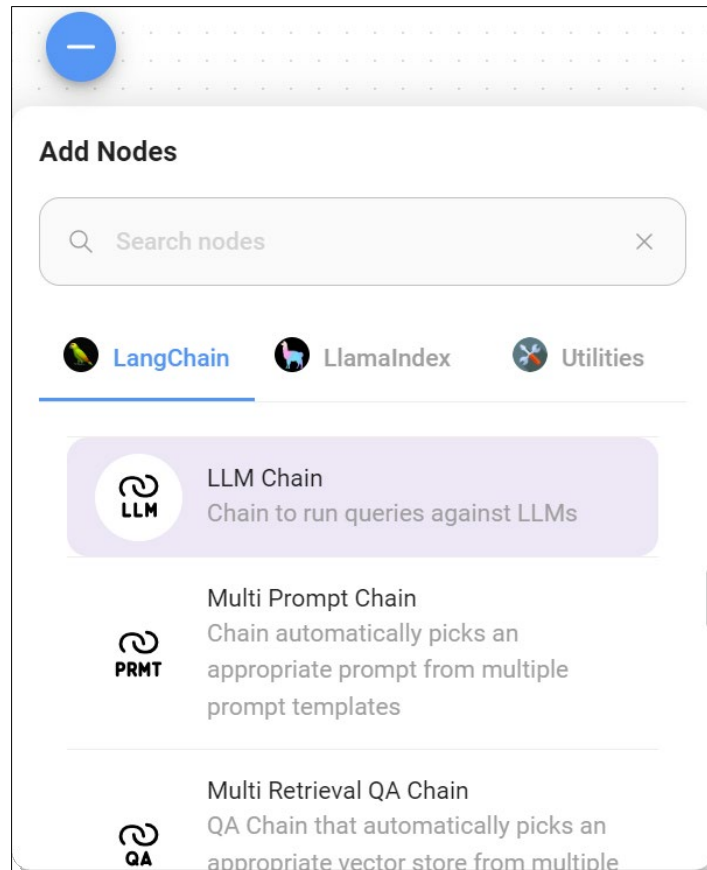


First LLM Application

- All Chatflows should contain at least one agent or one chain.
- Steps:
 - At Chatflow dashboard, click “Add New”    , a new “Untitled Chatflow” canvas is created. You will create the AI app using the blank canvas by drag and drop.
 - Click  to save the new Chatflow. Name your saved Chatflow as “Translator”.
 - Click  and look under “Chains”, drag “LLM Chain” and drop it to the blank canvas.

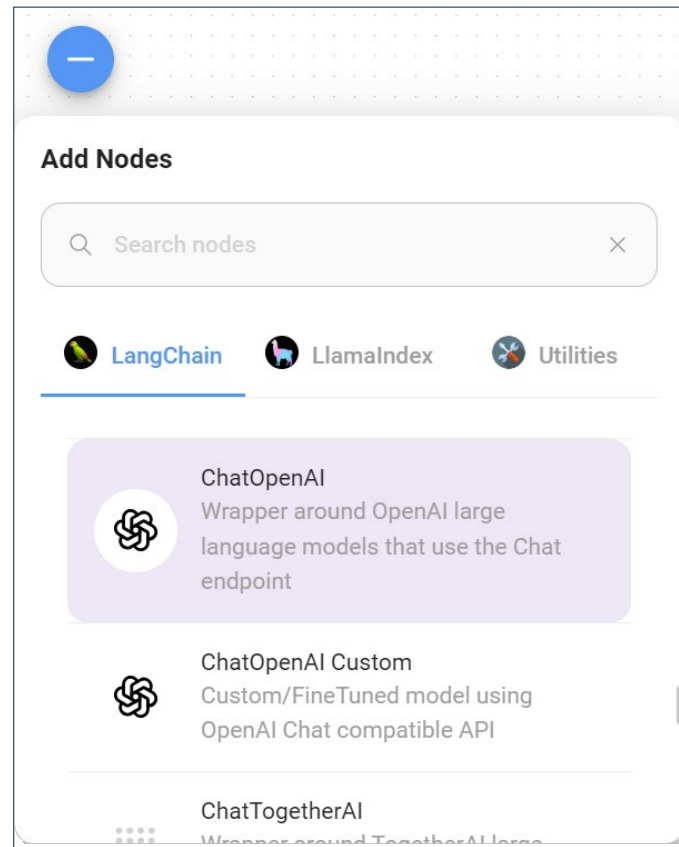
Translator

- Create a **Chains → LLM Chain** node.

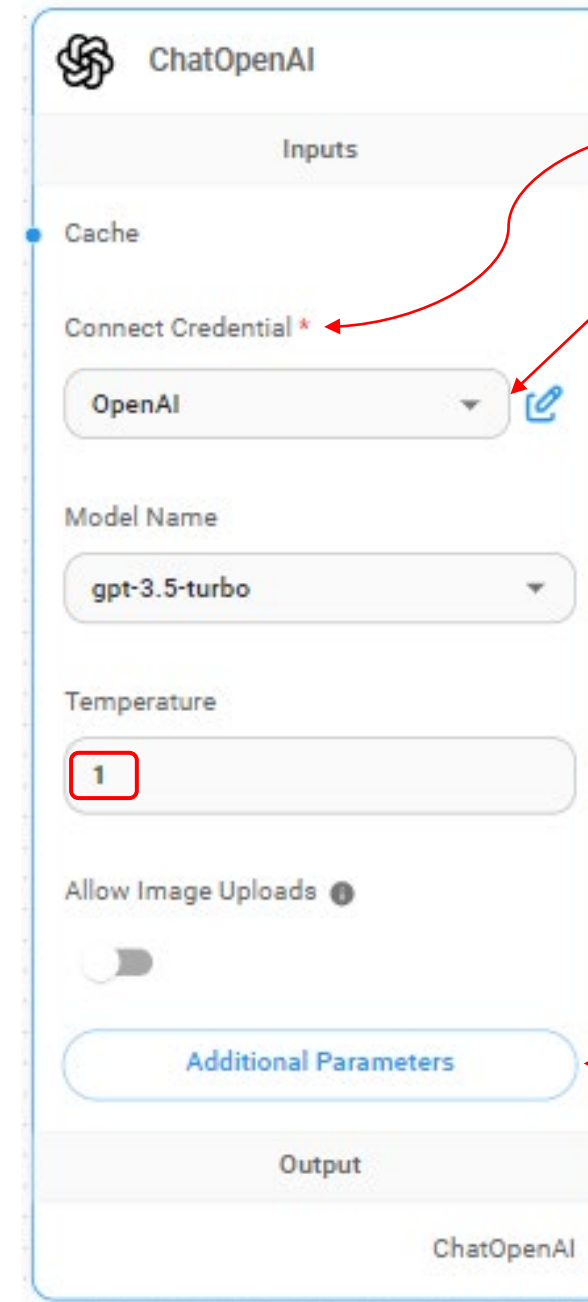


Translator

- Create a **Chat Models** → **ChatOpenAI** node.

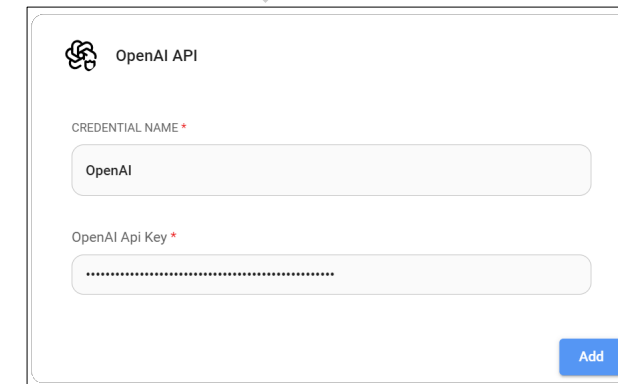
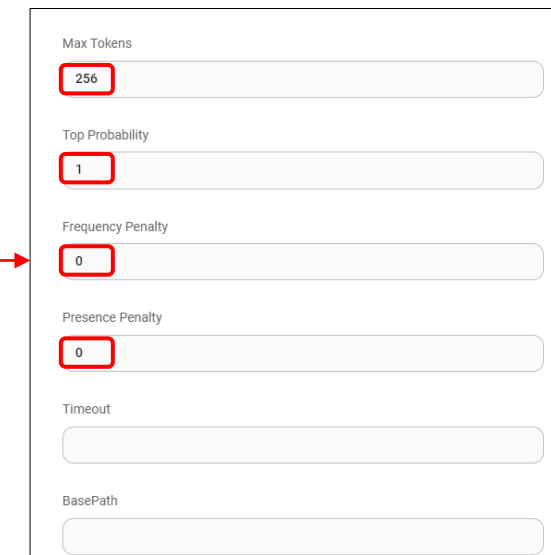


Drag & Drop
Canvas



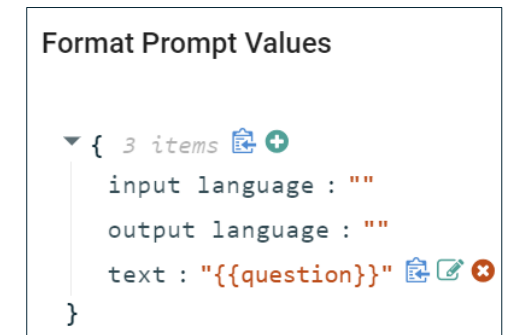
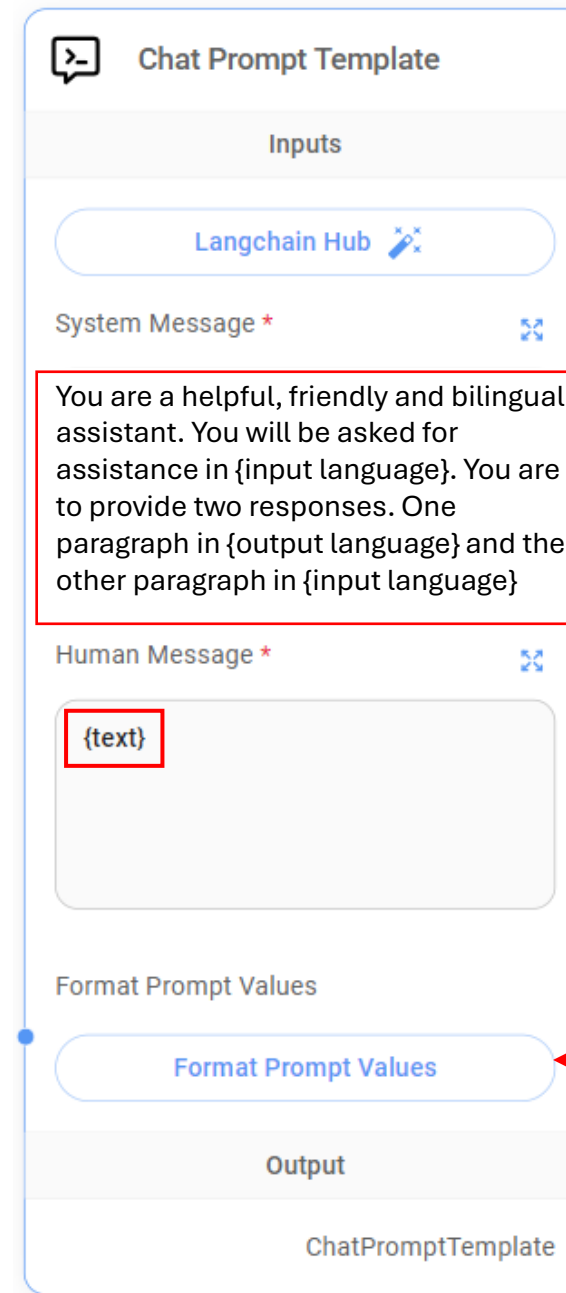
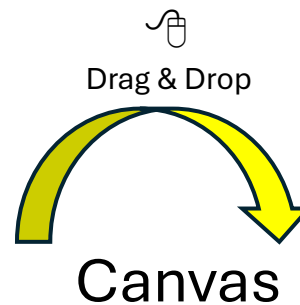
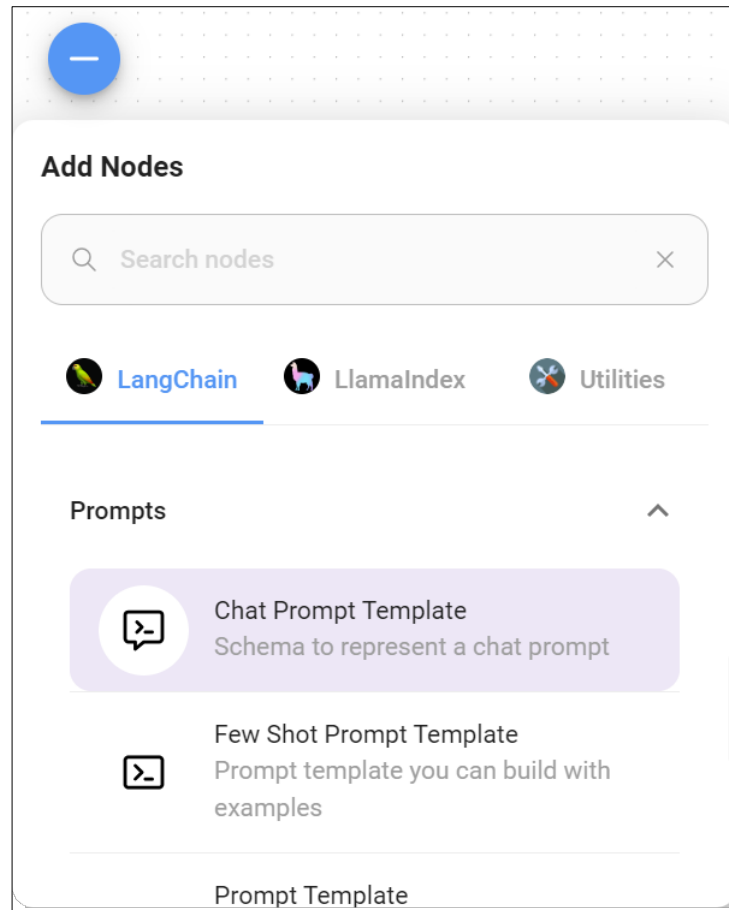
“*” means Mandatory

- Select “- Create New -”

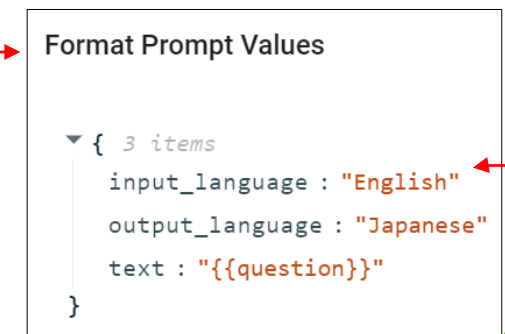



Translator

- Create a **Prompt** → **Chat Prompt Template** node.

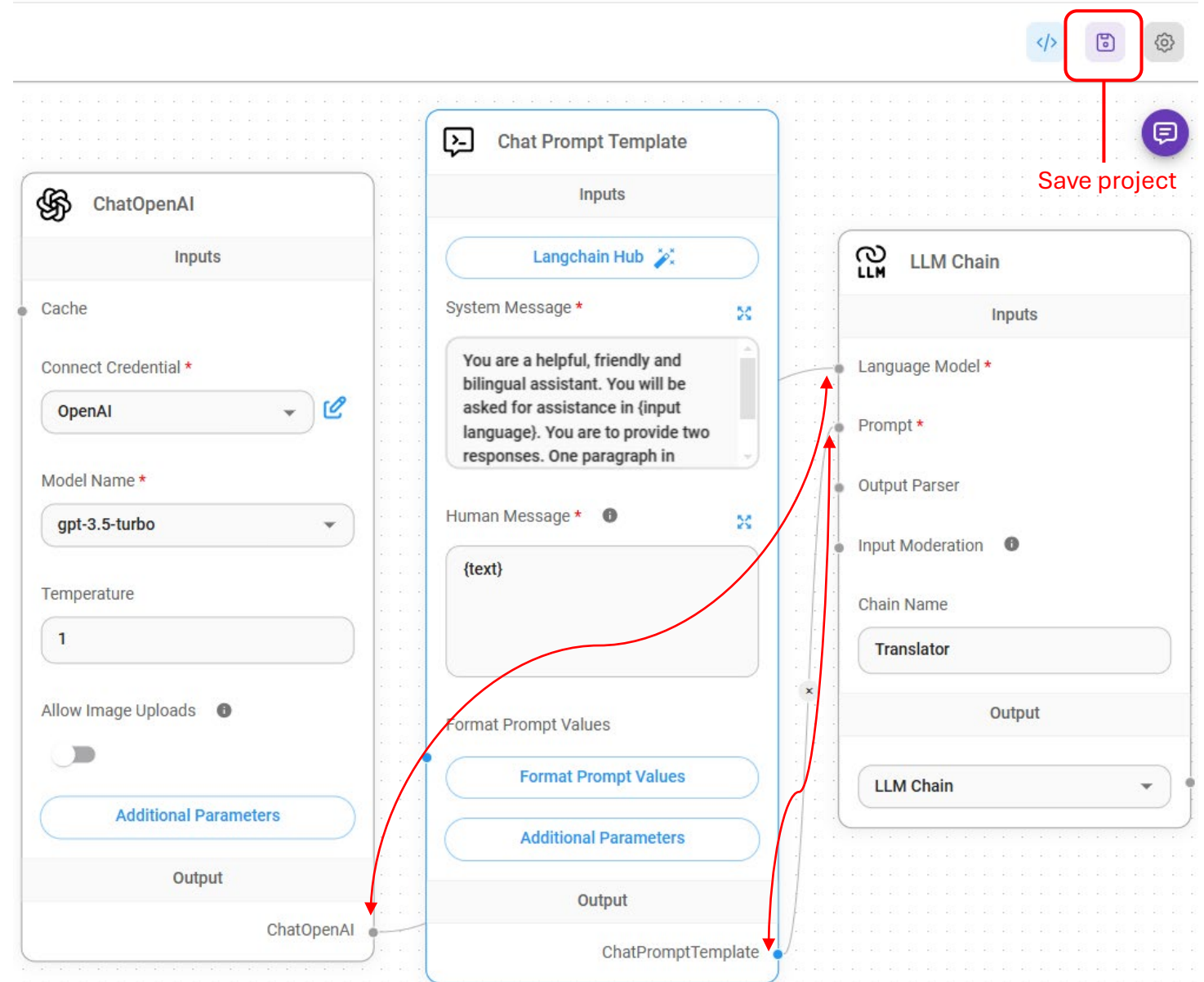
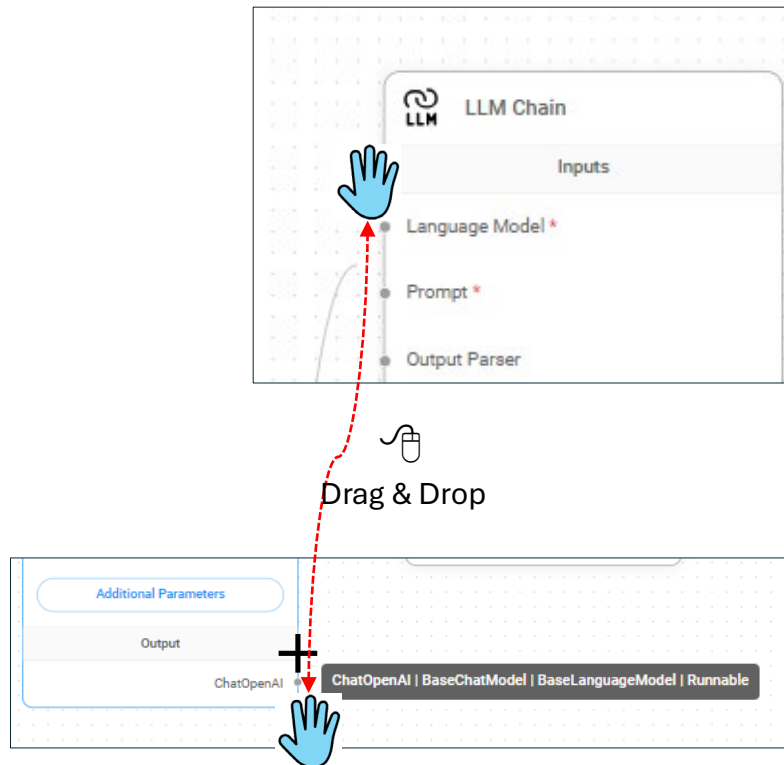


No need to enter the quotes




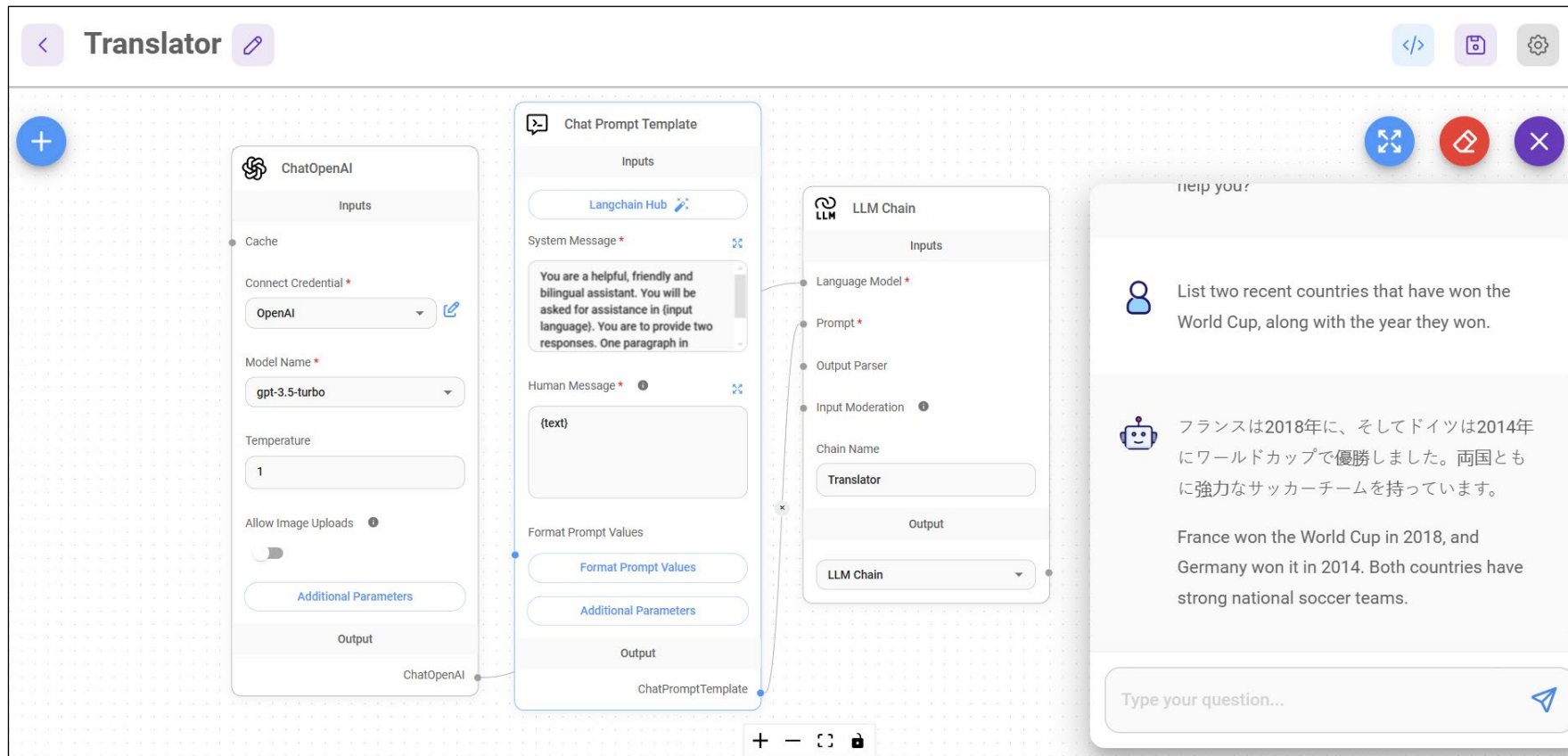
Translator

- Steps
 - Connect all the nodes.
 - Save the project.



Translator

- Anytime you want to test the translator Chatflow after you made changes, remember to **SAVE** the project first. Flowise does not auto-save your work.
- Click  to start chatting with the newly created AI LLM app.



Text Placeholders

System Message

You are a helpful, friendly and bilingual assistant. You will be asked for assistance in `{input language}`. You are to provide two responses. One paragraph in `{output language}` and the other paragraph in `{input language}`

Human Message

`{text}`

Format Prompt Values

Additional Parameters

Format Prompt Values

`{3 items`

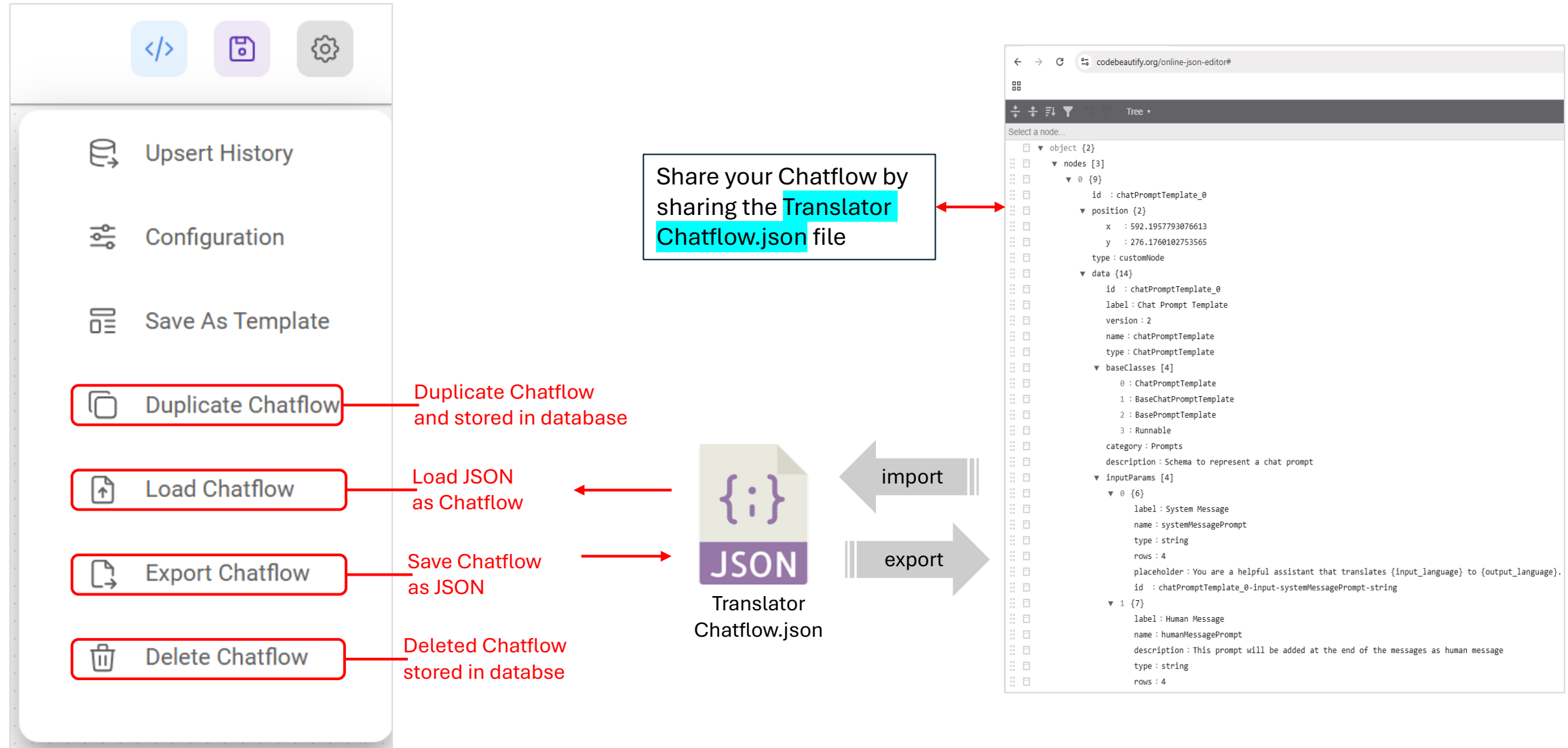
`input language:"English"`

`output language:"Japanese"`

`text:"{{question}}"`

`}`

Saving Translator Chatflow





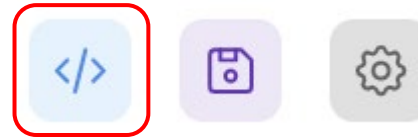
Flowise Deployment

Activity: API Endpoints

- Flowise provides several options for deployment.
- Issue curl command to chat with the LLM.
- Embed “<embed> HTML element” to “install” a chatbot in the HTML page to chat with the LLM.

API Endpoint (**cURL**)

- There are several methods to call your app. As shown below, the app can be called via Embed, Python, Javascript , cURL etc.



Embed in website or use as API

 Embed  Python  JavaScript  **CURL**  Share Chatbot

No Authorization ▼

```
curl http://localhost:3000/api/v1/prediction/3bed089f-4daa-4a39-ae59-1d9c2e496ba0 \
-X POST \
-d '{"question": "Hey, how are you?"}' \
-H "Content-Type: application/json"
```



Copy command

☐ Show Override Config

Read [here](#) on how to stream response back to application

API Endpoint (**cURL**)

- The command text provided is meant to run on “Unix/Linux” based operating system.
- Minor modifications are required to get it run in a Command Prompt under Window OS.

```
curl http://localhost:3000/api/v1/prediction/3bed089f-4daa-4a39-ae59-1d9c2e496ba0 -X POST -d "{\"question\":\"Hey, how are you?\"}" -H "Content-Type: application/json"
```

cURL Installation

Steps:

- 1) Go to <https://curl.se/windows/> to download curl for Windows i.e. curl for 64-bit i.e. curl-8.11.0_4-win64-mingw.zip.
- 2) Unzip the just downloaded file to a directory: curl-8.11.0_4-win64-mingw.
- 3) Open a Command Prompt and navigate to the directory: curl-8.11.0_4-win64-mingw/bin.
- 4) Run → `curl http://localhost:3000/api/v1/prediction/3bed089f-4daa-4a39-ae59-1d9c2e496ba0 -X POST -d "{\"question\": \"Hey, how are you?\"}\" -H \"Content-Type: application/json\"`

*Note: For the pink region shown above, you MUST replace it with your APP ID

cURL Observation

```

C:\Users\Your_username>cd downloads

C:\Users\Your_username\Downloads>cd curl-8.11.0_4-win64-mingw

C:\Users\Your_username\Downloads\curl-8.11.0_4-win64-mingw>cd bin

C:\Users\Your_username\Downloads\curl-8.11.0_4-win64-mingw\bin>dir
Volume in drive C has no label.
Volume Serial Number is 2AA1-E26C

Directory of C:\Users\Your_username\Downloads\curl-8.11.0_4-win64-mingw\bin

11/06/2024  07:09 AM    <DIR>          .
11/06/2024  07:09 AM    <DIR>          ..
11/26/2024  01:58 PM                236,849 curl-ca-bundle.crt
11/06/2024  07:09 AM            3,638,888 curl.exe
11/06/2024  07:09 AM                2,353 libcurl-x64.def
11/06/2024  07:09 AM            3,188,840 libcurl-x64.dll
               4 File(s)            7,066,930 bytes
               2 Dir(s)  116,778,516,480 bytes free

C:\Users\Your_username\Downloads\curl-8.11.0_4-win64-mingw\bin>curl
http://localhost:3000/api/v1/prediction/3bed089f-4daa-4a39-ae59-1d9c2e496ba0 -X POST -d '{"question\": \"Hey,
how are you?\"}' -H 'Content-Type: application/json'
{"text":"こんにちは！お元気ですか？何かお手伝いできますことがありますか？Hello! How are you doing? How can I assist you
today?","question":"Hey, how are you?","chatId":"ca55aeb7-6f70-41bb-9370-
a710556bd0d2","chatMessageId":"04fc1199-90f2-4f49-95a1-
d40b95986039","isStreamValid":false,"sessionId":"ca55aeb7-6f70-41bb-9370-a710556bd0d2"}
C:\Users\Your_username\Downloads\curl-8.11.0_4-win64-mingw\bin>

```

API Endpoint (**Embed**)

- `<embed>` is a HTML element.
- Embeds external content at specified point in the HTML page.
- The content is provided by an external application or other source of interactive content such as a browser plug-in.
- Use NodeJS Express to demonstrate the working of Embed (Popup HTML).



API Endpoint (**Embed**)



Embed in website or use as API

 **Embed**  Python  JavaScript  CURL  Share Chatbot

No Authorization ▼

Popup Html Fullpage Html Popup React Fullpage React

Paste this anywhere in the <body> tag of your html file.

You can also specify a [version](#): `https://cdn.jsdelivr.net/npm/flowise-embed@<version>/dist/web.js`

```
<script type="module">
  import Chatbot from "https://cdn.jsdelivr.net/npm/flowise-embed/dist/web.js"
  Chatbot.init({
    chatflowid: "3bed089f-4daa-4a39-ae59-1d9c2e496ba0",
    apiHost: "http://localhost:3000",
  })
</script>
```

Replace chatflowid with
yours found in **index.html**

☐ Show Embed Chat Config

API Endpoint (**Embed**)

- 1) Open a **Command Prompt** and navigate to the directory: **node-v22.12.0-win-x64**
- 2) Make directory: **mkdir projects\simpleHTTP**
- 3) Navigate to directory: **cd projects\simpleHTTP**
- 4) Install Express by:
 - a) **..\..\npm init**
(press “Enter” for all the questions and “yes” to complete the initialization)
 - b) **..\..\npm install express -save**
 - c) **..\..\npm fund**
- 5) Copy **index.js**, **index.html** and **flowise.png** to **projects\simpleHTTP**
- 6) Run Express: **..\..\node index.js**
- 7) Go to browser and enter URL: **http://localhost:5500**

Observation

```
C:\Users\Your_username\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP>..\..\npm init
```

This utility will walk you through creating a package.json file.

It only covers the most common items, and tries to guess sensible defaults.

See `npm help init` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.

package name: (simplehttp)

version: (1.0.0)

description:

entry point: (index.js)

test command:

git repository:

keywords:

author:

license: (ISC)

About to write to C:\Users\Your_username\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP\package.json:

Observation

```
{  
  "name": "simplehttp",  
  "version": "1.0.0",  
  "main": "index.js",  
  "scripts": {  
    "test": "echo \"Error: no test specified\" && exit 1"  
  },  
  "author": "",  
  "license": "ISC",  
  "description": ""  
}
```

Is this OK? (yes) yes

C:\Users\Your_username\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP>

Observation

```
C:\Users\koay_seng_tian\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP>dir
```

```
Volume in drive C has no label.
```

```
Volume Serial Number is 2AA1-E26C
```

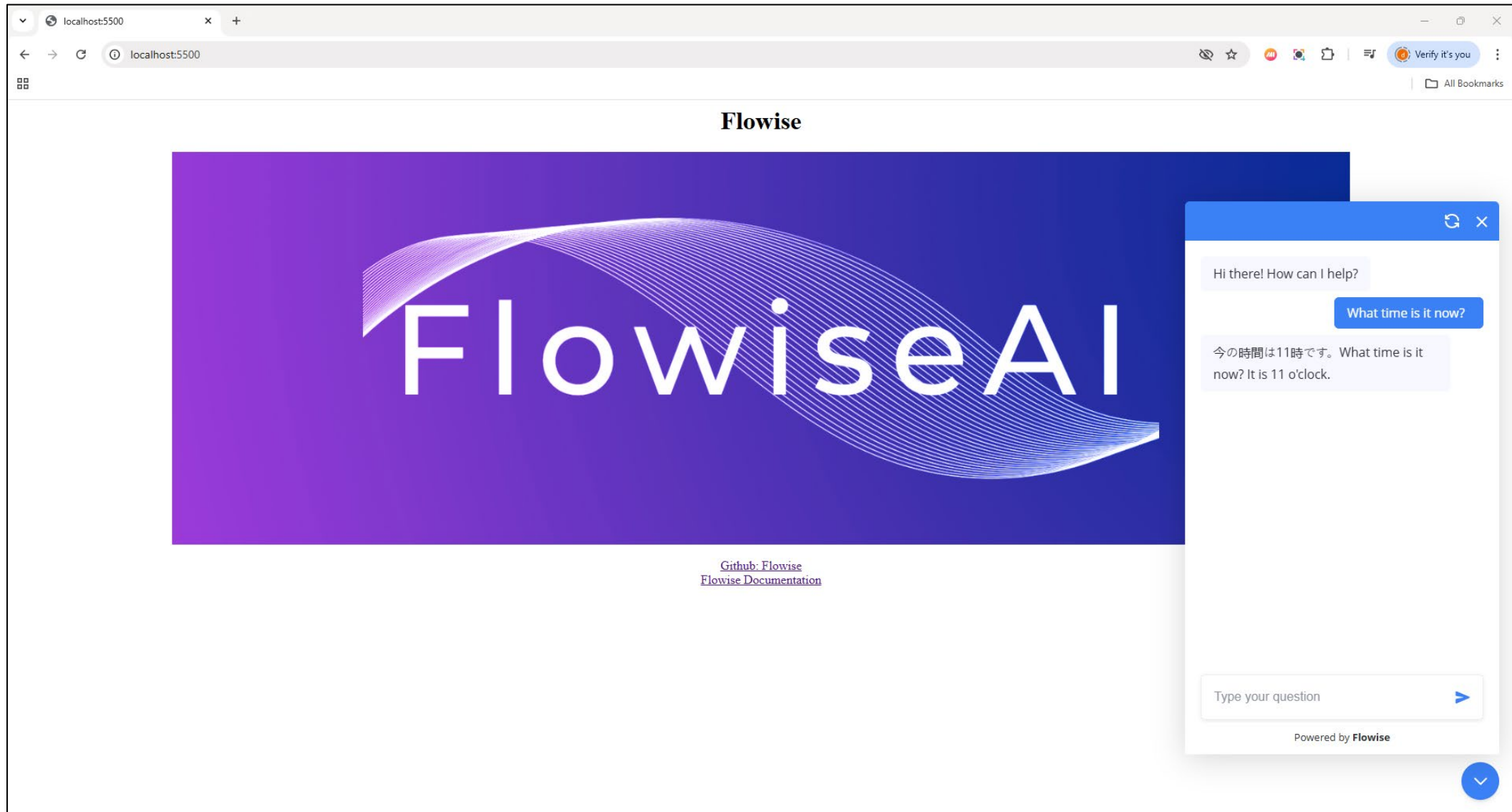
```
Directory of C:\Users\koay_seng_tian\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP
```

```
12/10/2024  11:22 AM    <DIR>          .
12/10/2024  11:09 AM    <DIR>          ..
04/07/2024  04:09 PM             536,083 flowise.png
04/07/2024  06:01 PM             764 index.html
04/07/2024  04:03 PM             302 index.js
12/10/2024  11:20 AM    <DIR>        node_modules
12/10/2024  11:20 AM             28,154 package-lock.json
12/10/2024  11:20 AM             256 package.json

        5 File(s)          565,559 bytes
        3 Dir(s)  117,105,750,016 bytes free
```

```
C:\Users\koay_seng_tian\Downloads\node-v22.12.0-win-x64\projects\simpleHTTP>
```

Result



Reference

- Flowise

<https://docs.flowiseai.com/>

Quiz 1

- <https://forms.office.com/r/S9AUdfYA8r>

GenAI & LLM App Development (1)



Thank you!