

From Nodes to Prompts: Exploring Prompt Engineering with Rivet

Jack

Vatcharat Rattananun



PRODUCT CONSULTING

1

New Digital Product Innovation

- Validate Product-Market Fit
- Formulate Implementation Plan
- Minimize Development Cost

2

Existing Product Optimization

- Increase Product Adoption
- Identify Product Improvements
- Improve Viability and Profitability



Contact Us

ORGANIZATION CONSULTING

Delivery Acceleration

3

- Reduce Time to Market
- Enable Value-Based Delivery
- Modernize Tech Infrastructure
- Align Business and Product Delivery
- Staff Empowerment

Sustainability For Positive Impact

4

- Sustainability Fundamentals
- ESG Data Management
- Enabling An Impact Mindset
- Tech For Good



Contact Us

Our Team



Jack

Vatcharat

Rattanapun
FullStack Developer



Sun

Sirapol Wareechuensuk

FullStack Developer



Mai

Ronnapat Buranakate

FullStack Developer



Agenda

- 1 Introduction
- 2 Overview of Rivet
- 3 Demonstration
- 4 X-Shot Prompts
- 5 Hands-on Labs
- 6 Integrating Rivet
- 7 Graphs in Node.js Applications
- 8 Graphs in Node.js Applications
- 9 Debugging with Rivet
- 10 Q&A Session

Introduction



Do you have experience with
NodeJS ?

Click to add text
Click to add text

Do you know Prompt Engineering ?



Have you heard about **Rivet** ?

Click to add text
Click to add text

Introduction

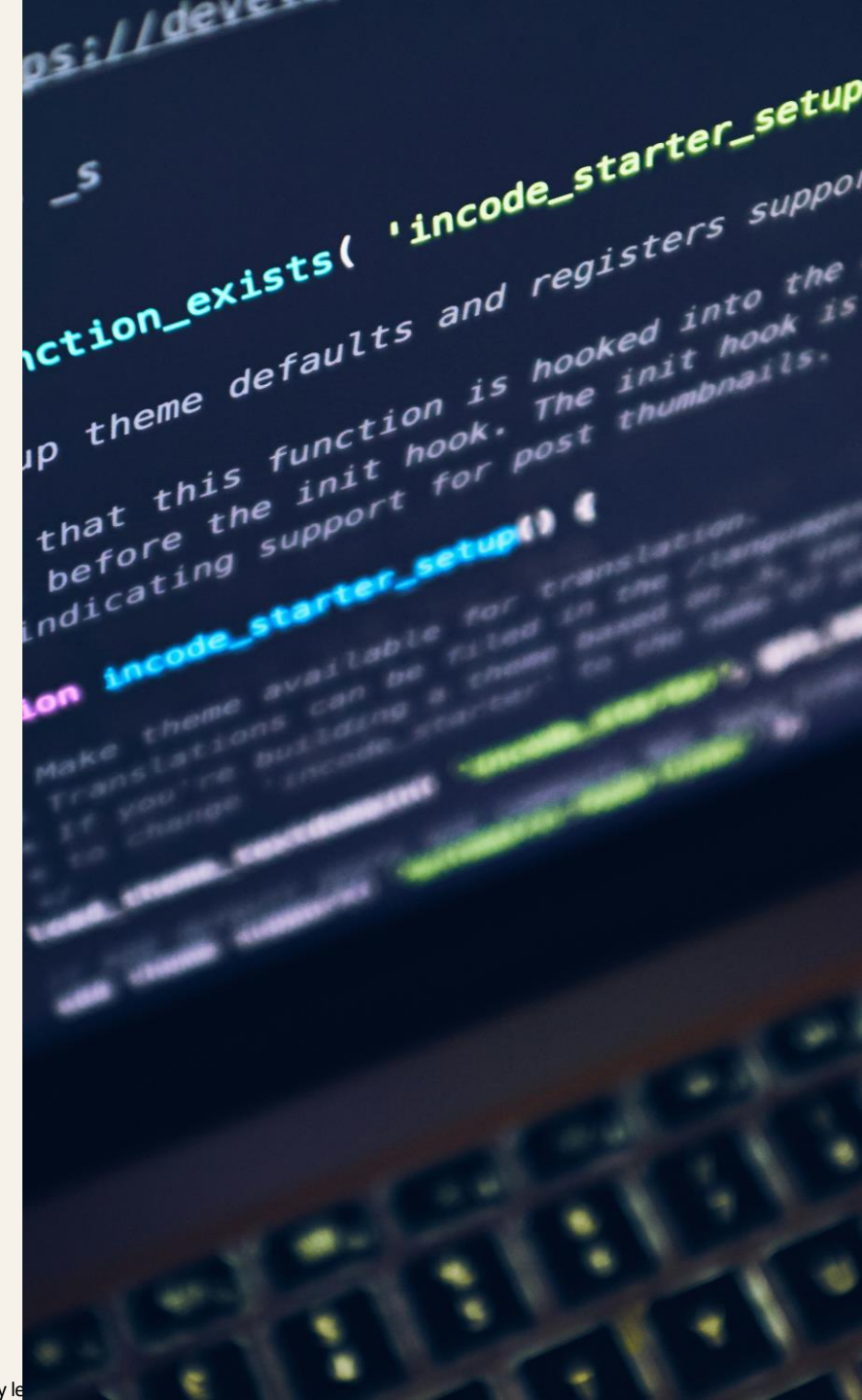
Workshop objectives

- Learn to write effective one-shot prompts.
- Develop skills to create few-shot prompts.
- Understand how to integrate Rivet graphs with a Node.js application.
- Learn debugging techniques to visualize and troubleshoot the flow.



Rivet

#JSBangkok #JSBKK



Overview of Rivet

What is it?

- A visual programming environment for building AI agents with LLMs.
- Allows iteration on prompt graphs and direct application integration.



Overview of Rivet

Why do we use it?



Visualize and Build

Create complex chains for production applications.



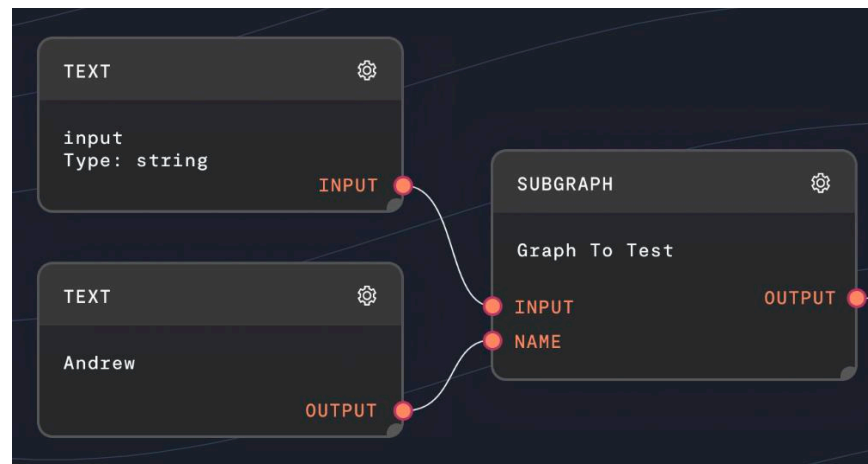
Debug Remotely

Observe and debug prompt chains in real-time.



Collaborate

- YAML files
- Versioning



Demonstration

#JSBangkok #JSBKK



What we're going to do

The screenshot displays the Rivet 1.9.0 - ReAct interface, which is a workflow automation tool. The interface is divided into several sections:

- Top Bar:** Shows the project name "Rivet 1.9.0 - ReAct" and the file path "(/Users/jack/Work/rivet-workshop-solution/rivet/main.rivet-project)". It also includes a "Discord" link.
- Left Sidebar:** Contains a "Search..." bar and a list of workflows. The selected workflow is "2. One-shot".
- Canvas:** The main workspace where the workflow is built. It shows a sequence of steps connected by arrows:
 - PROMPT (System):** "You are a translator. Translate the".
 - CHAT (User):** A text input field containing a paragraph about TikTok and its impact on the U.S. economy.
 - PROMPT (Assistant):** "The article from BBC Travel highlight".
 - ASSEMBLE PROMPT:** A step that combines the outputs of the previous steps into a single prompt for the next chat step.
 - CHAT (System):** A text input field containing a URL and model information: "https://api.groq.com/openai/3000 tokens Model: llama3-8b-8192 Temperature: 1".
- Right Panel:** Contains a "Save Recording" button and a "Run 2. Translate and Summarize/2. One-shot" button. Below these buttons, there are tabs for "Chat Viewer" and "Data Studio".

Setting up the environment

#JSBangkok #JSBKK

Information Classification: CONFIDENTIAL (sensitive business information, the level of protection is dictated by le



Prerequisites

Setting Up the Environment

- Rivet
- GroqAI key (Free AI)
- Workshop repository
- Node.js (v20.17.0)
- Postman
- VS Code or your favorite code editor



Download Rivet

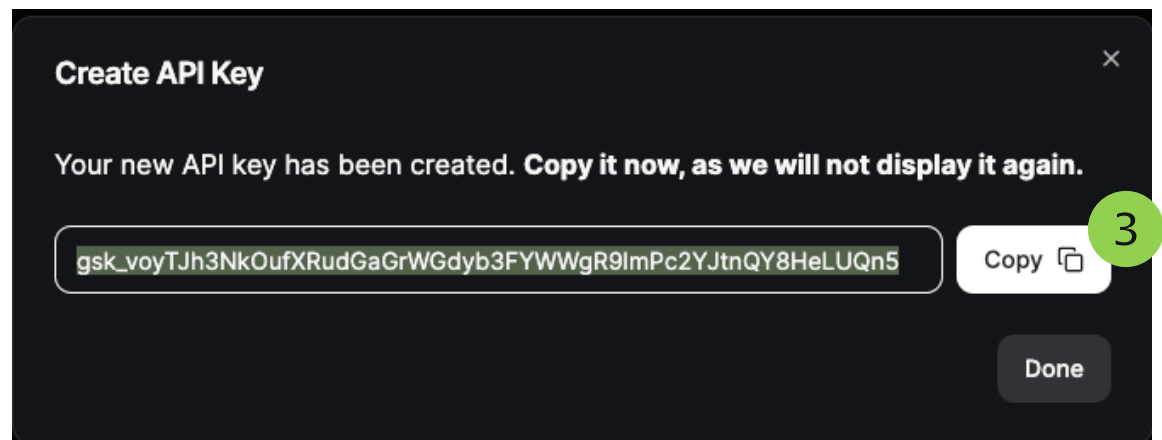
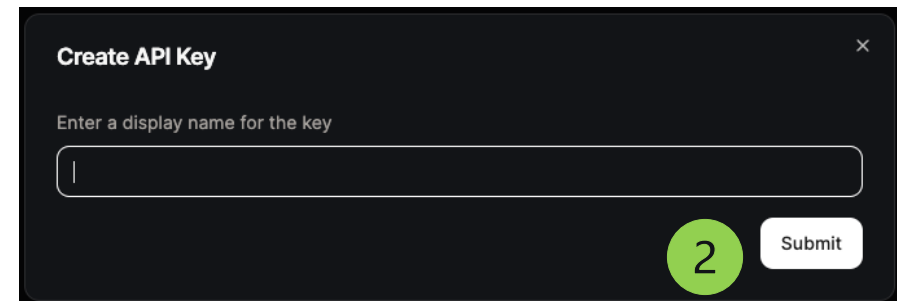
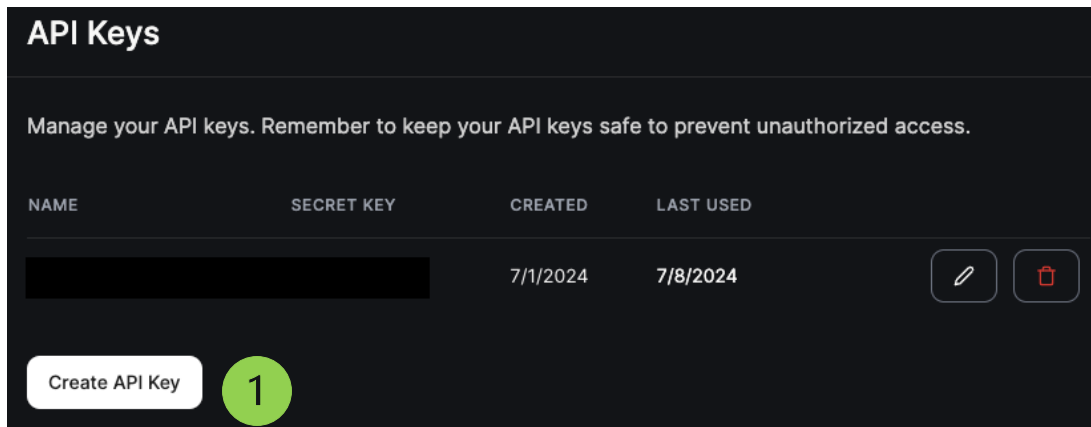
Setting Up the Environment

<https://rivet.ironcladapp.com/>

Create GroqAI key

Setting Up the Environment

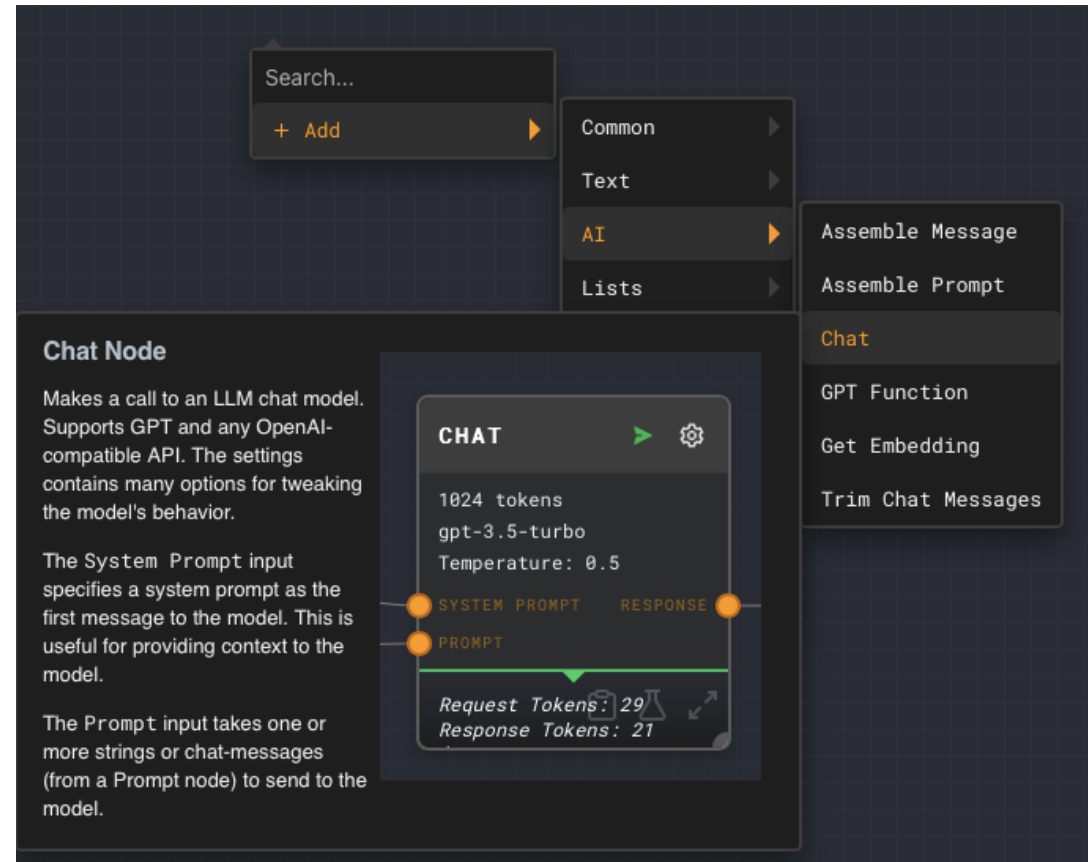
<https://console.groq.com/>



Use Groq in Rivet

Setting Up the Environment

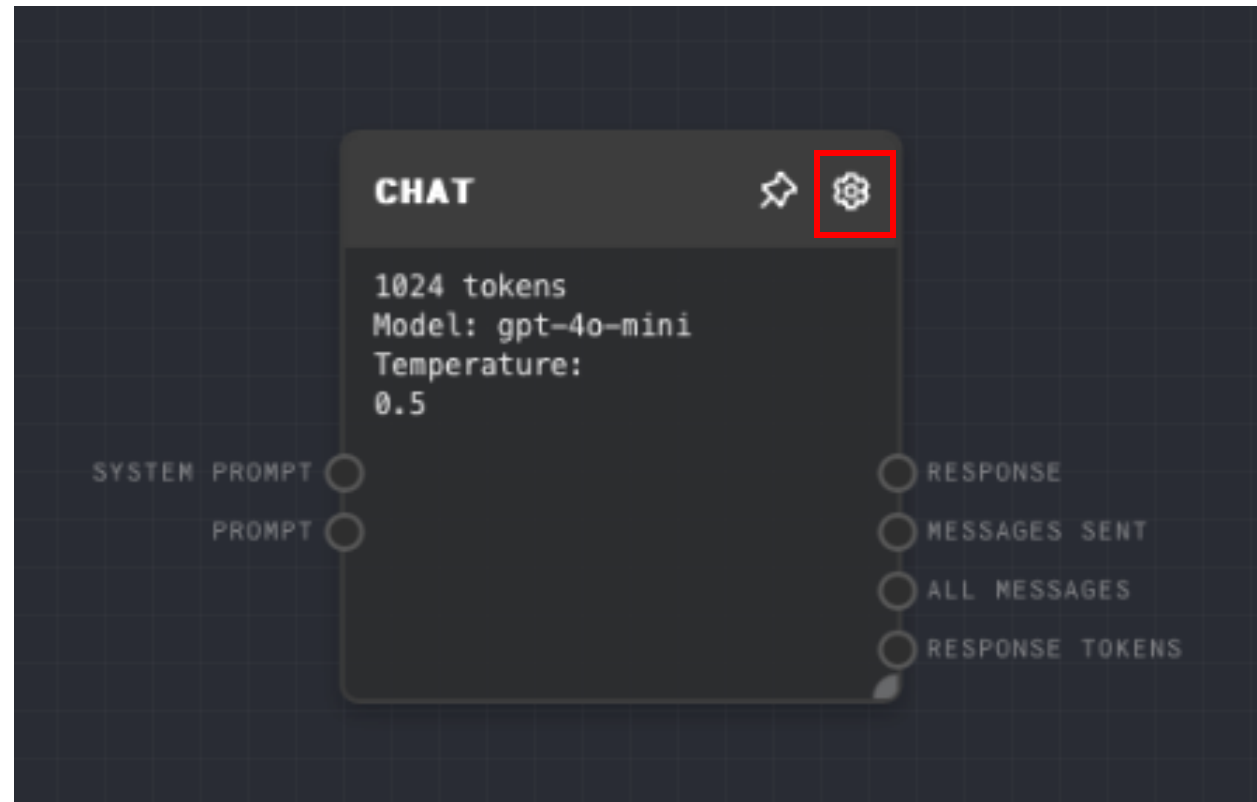
1. Right-click on an empty space
2. Create Chat node by clicking Add => AI => Chat node



Use Groq in Rivet

Setting Up the Environment

3. A new chat node should appear
4. Click Gear icon (red rectangle)

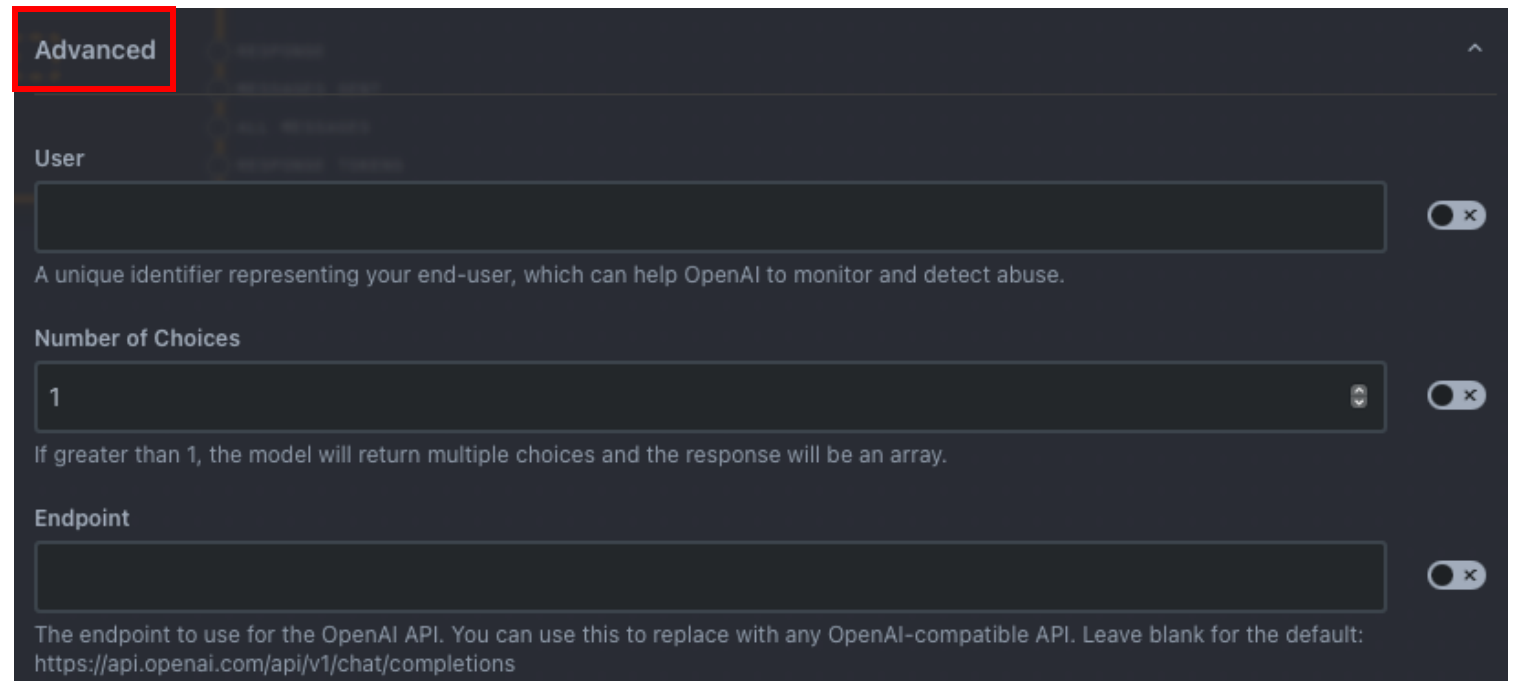


Use Groq in Rivet

Setting Up the Environment

5. Scroll-down the right section

6. Expand Advanced tab



The screenshot shows the Rivet configuration interface with the 'Advanced' tab selected and highlighted by a red box. The interface is dark-themed and contains several configuration fields:

- User:** A text input field with a toggle switch on the right. Below it, a description reads: "A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse."
- Number of Choices:** A text input field containing the value '1' with a toggle switch on the right. Below it, a description reads: "If greater than 1, the model will return multiple choices and the response will be an array."
- Endpoint:** A text input field with a toggle switch on the right. Below it, a description reads: "The endpoint to use for the OpenAI API. You can use this to replace with any OpenAI-compatible API. Leave blank for the default: <https://api.openai.com/api/v1/chat/completions>"

Use Groq in Rivet

Setting Up the Environment

7. Use Groq API endpoint:

`https://api.groq.com/openai/v1/chat/completions`

8. Use llama3-8b-8192 model

9. Replace `<your-api-key>` with your Groq API key

The screenshot shows the Rivet configuration interface for the OpenAI API endpoint. The fields are as follows:

- Endpoint:** `https://api.groq.com/openai/v1/chat/completions` (highlighted with a green circle 7). A toggle switch is on the right.
- Custom Model:** `llama3-8b-8192` (highlighted with a green circle 8). A description below reads: "Overrides the model selected above with a custom string for the model."
- Custom Max Tokens:** (empty field). A description below reads: "Overrides the max number of tokens a model can support. Leave blank for preconfigured token limits."
- Headers:** A section titled "Additional headers to send to the API." containing two rows:
 - Row 1: `Authorization` (key) and `Bearer <your-api-key>` (value, highlighted with a green circle 9). A toggle switch is on the right.
 - Row 2: `Content-Type` (key) and `application/json` (value).

An "Add" button is located at the bottom left of the Headers section.

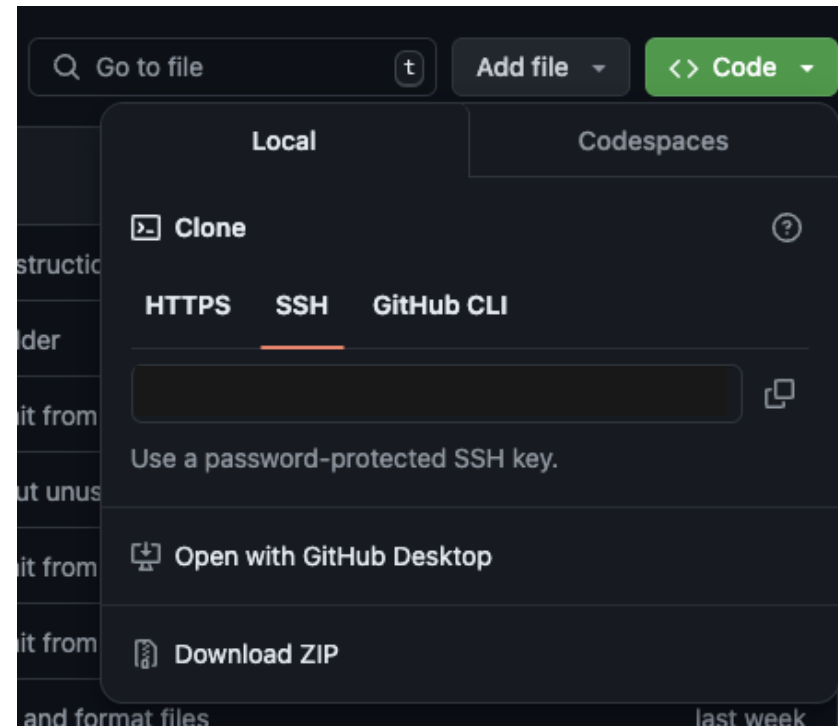
Download the repo

Setting Up the Environment

1 <https://github.com/palo-it-th/rivet-workshop>



2 Clone or download ZIP file



Technical terms

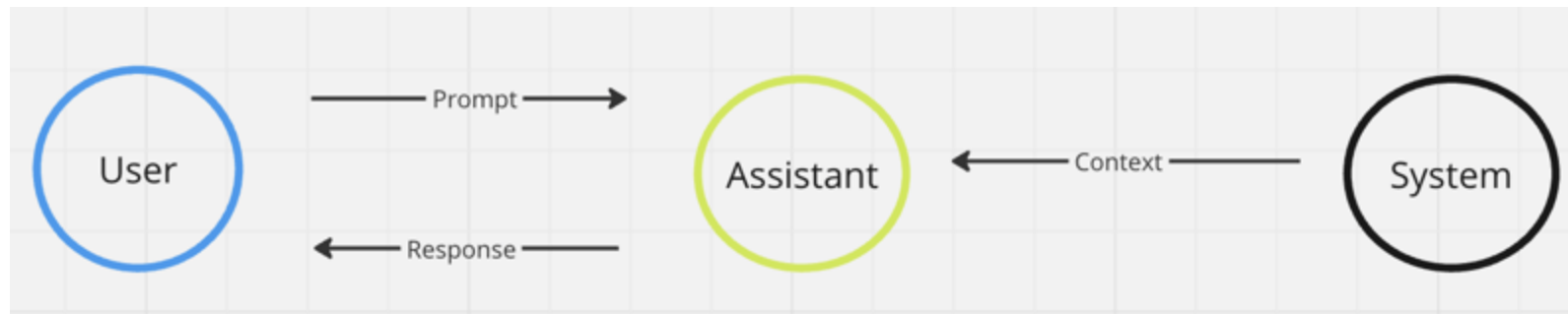


Technical terms

System Prompt is a set of instructions given to the AI to define how it should behave.

User Prompt is the input or question that the user provide to the AI.

Assistant Prompt is the response generated by the AI.



One-shot prompt



Zero-shot prompting

PALO IT One Shot ReAct

Hello, human!

Switch vocal and vowel letters Input: Gale

If we switch the vocal and vowel letters in "Gale", we get "Lage".

Switch vocal and vowel letters Input: Gale Output: JSON format

Here is the output in JSON format with switched vocal and vowel letters:

```
{  
  "input": "Gale",  
  "output": "Gilea"  
}
```

Explanation:
In the input "Gale", the vowel is "a" and the consonant is "Gle".
To switch the vocal and vowel letters, I replaced the vowel "a" with the consonant "i" and vice versa, resulting in "Gilea".

One-shot prompting

Switch vocal and vowel letters

Input: Gale

Output:

***json { "explanation":

"In the input "Gale", the vowel is "a" and the consonant is "Gle". To switch the vocal and vowel letters, I replaced the vowel "a" with the consonant "i" and vice versa, resulting in "Gilea"." "input": "Gale", "output": "Gilea" }

Input: Palo

Output:



***json { "explanation": "In the input 'Palo', the vowel is 'a' and the consonants are 'Plo'. To switch the vocal and vowel letters, I replaced the vowel 'a' with the consonant 'i' and vice versa, resulting in 'Pilo'.", "input": "Palo", "output": "Pilo" }

What is it?

One-shot prompt

A single example prompt given to the AI to help it respond correctly.

SYSTEM	Answer in a consistent style.
USER	Teach me about patience.
ASSISTANT	The river that carves the deepest valley flows from a modest spring; the grandest symphony originates from a single note; the most intricate tapestry begins with a solitary thread.
USER	Teach me about the ocean.

How does the code look?

Specify roles in OpenAI messages array

```
messages: [  
  {  
    role: 'system',  
    content: 'Answer in a consistent style.',  
  },  
  {  
    role: 'user',  
    content: 'Teach me about patience.',  
  },  
  {  
    role: 'assistant',  
    content: 'The river that carves...',  
  },  
  {
```

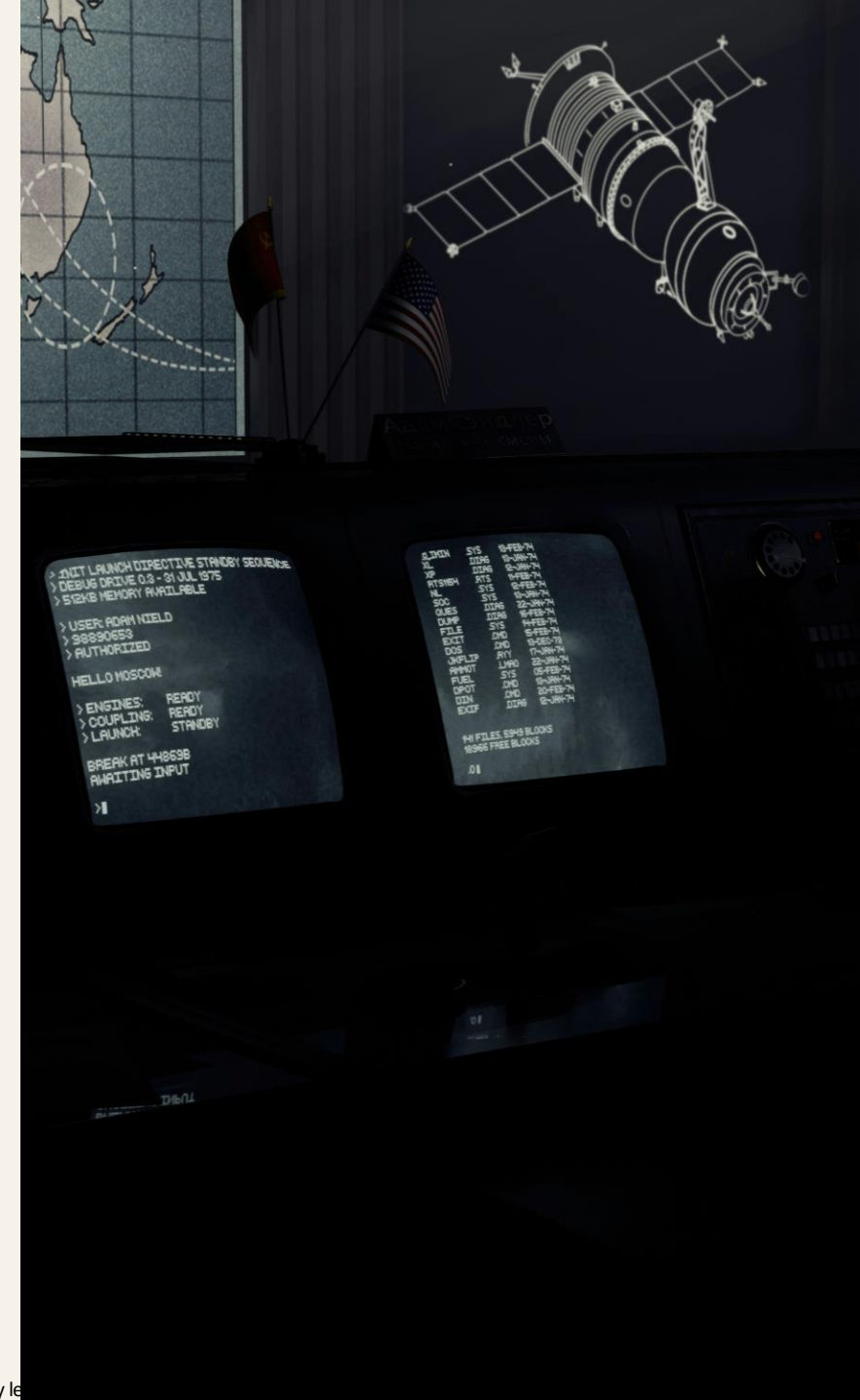
Few-shot prompts

What are they?

A few example prompts given to the AI to help it respond correctly.

Hands-on Lab #1

#JSBangkok #JSBKK



Hands-on Lab #1

1. Use job-description.md under exercises folder as the graph input
2. Extract Job Title, Company, Location and output in JSON format

```
{  
  
  "jobTitle": "Digital Marketing  
Specialist",  
  
  "company": "Naruto Co., Ltd.",  
  
  "location": "Konoha Village",  
  
}
```

Hands-on Lab #1

Your Turn

Create a one-shot prompt in Rivet:

1. Input job description and output in JSON format.
2. To determine the sentiment of a given sentence.
3. To translate a simple English sentence into French and Chinese.



Hands-on Lab #2

#JSBangkok #JSBKK

Information Classification: CONFIDENTIAL (sensitive business info)



Hands-on Lab #2

Your Turn

1. Open Translate and Summarize folder in Rivet.
2. Use example in exercises/news folder.
3. Use one-shot prompting technique to translate and summarize news article in the following format:

<Summary>

Key Highlights:

- Key point #1
- Key point #2
- Key point #3



Using Rivet graph in Node.js applications

#JSBangkok #JSBKK

Information Classification: CONFIDENTIAL (sensitive business information, the level of protection is dictated by le




USING RIVET GRAPH IN NODE.JS APPLICATIONS

1. Import Rivet



```
import { runGraphInFile, RunGraphOptions, startDebuggerServer } from '@ironclad/rivet-node'
```

2. Read Rivet project file



```
const result = await runGraphInFile(rivetProjPath, {  
  graph: graph,  
  inputs: rivetInputs,  
  openAiKey: process.env.OPENAI_API_KEY,  
} as RunGraphOptions)
```



USING RIVET GRAPH IN NODE.JS APPLICATIONS

3.Run

Run this command on Terminal:

```
npm run dev
```

If you have a program listening to port number 8080, create **.env** file and put

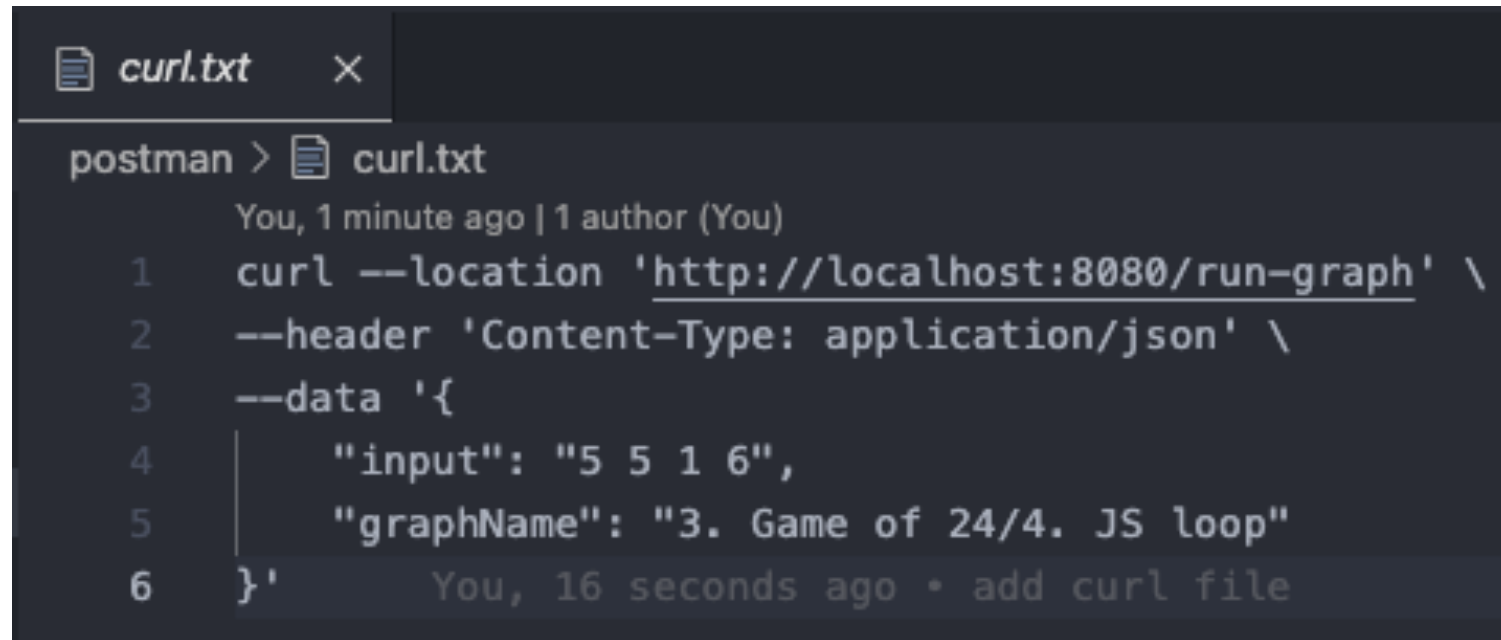
```
PORT=8084
```



DEBUGGING IN RIVET

4. Copy curl and paste in Postman

Add slide copy here.



The screenshot shows a Postman interface with a dark theme. At the top, there is a tab labeled 'curl.txt' with a close button. Below the tab, the text 'postman > curl.txt' is displayed. The main area contains a curl command being pasted line by line, with line numbers 1 through 6 on the left. The command is: curl --location 'http://localhost:8080/run-graph' \ --header 'Content-Type: application/json' \ --data '{ "input": "5 5 1 6", "graphName": "3. Game of 24/4. JS loop" }' . At the bottom right, there is a status bar that says 'You, 16 seconds ago • add curl file'.

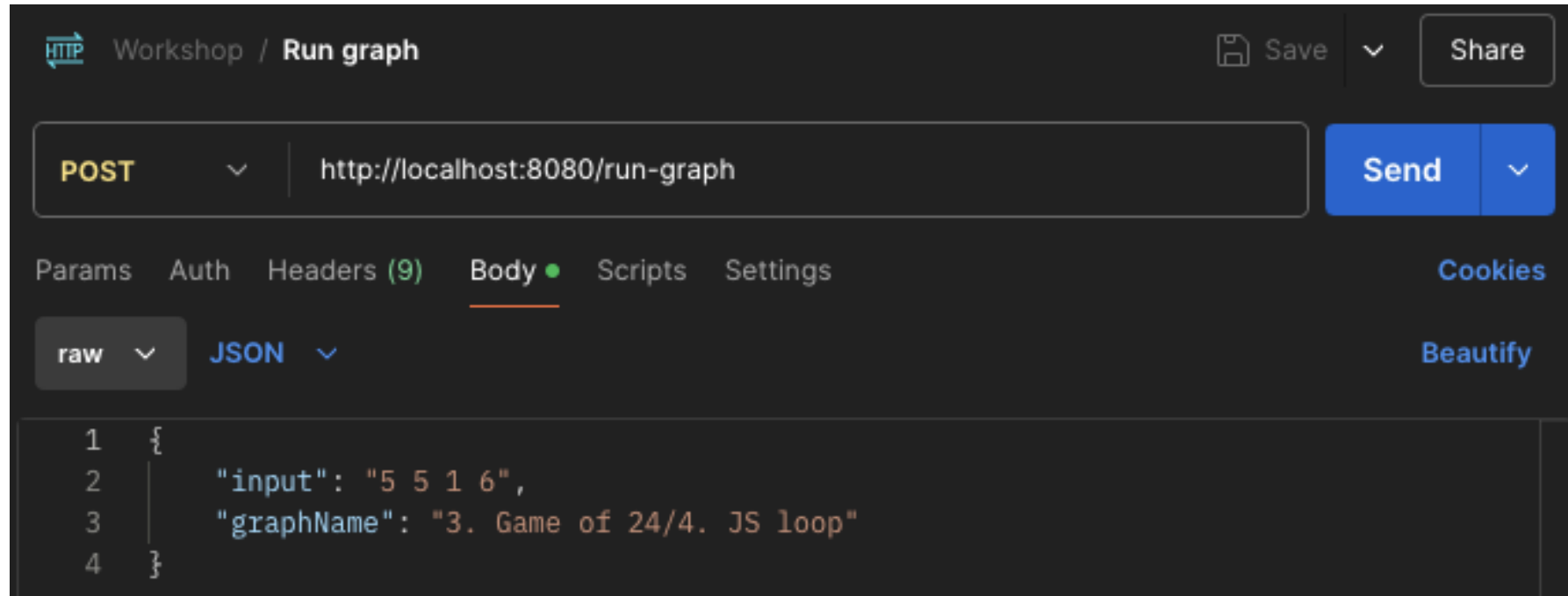
```
1 curl --location 'http://localhost:8080/run-graph' \  
2 --header 'Content-Type: application/json' \  
3 --data '{  
4     "input": "5 5 1 6",  
5     "graphName": "3. Game of 24/4. JS loop"  
6 }'
```



DEBUGGING IN RIVET

5. Send request in Postman

Add slide copy here.



Debugging in Rivet

#JSBangkok #JSBKK

Information Classification: CONFIDENTIAL (sensitive business information, the level of protection is dictated by le



DEBUGGING IN RIVET

1. Create debugger server

Add slide copy here.



```
const debuggerServer = startDebuggerServer({ port: 21887 })
```



DEBUGGING IN RIVET

2. Use as remoteDebugger

Add slide copy here

```
const result = await runGraphInFile(rivetProjPath, {  
  graph: graph,  
  remoteDebugger: debuggerServer,  
  inputs: rivetInputs,  
  context: {},  
  externalFunctions: {},  
  onUserEvent: {},  
  openAiKey: process.env.OPENAI_API_KEY,  
} as RunGraphOptions)
```



DEBUGGING IN RIVET

3. Run

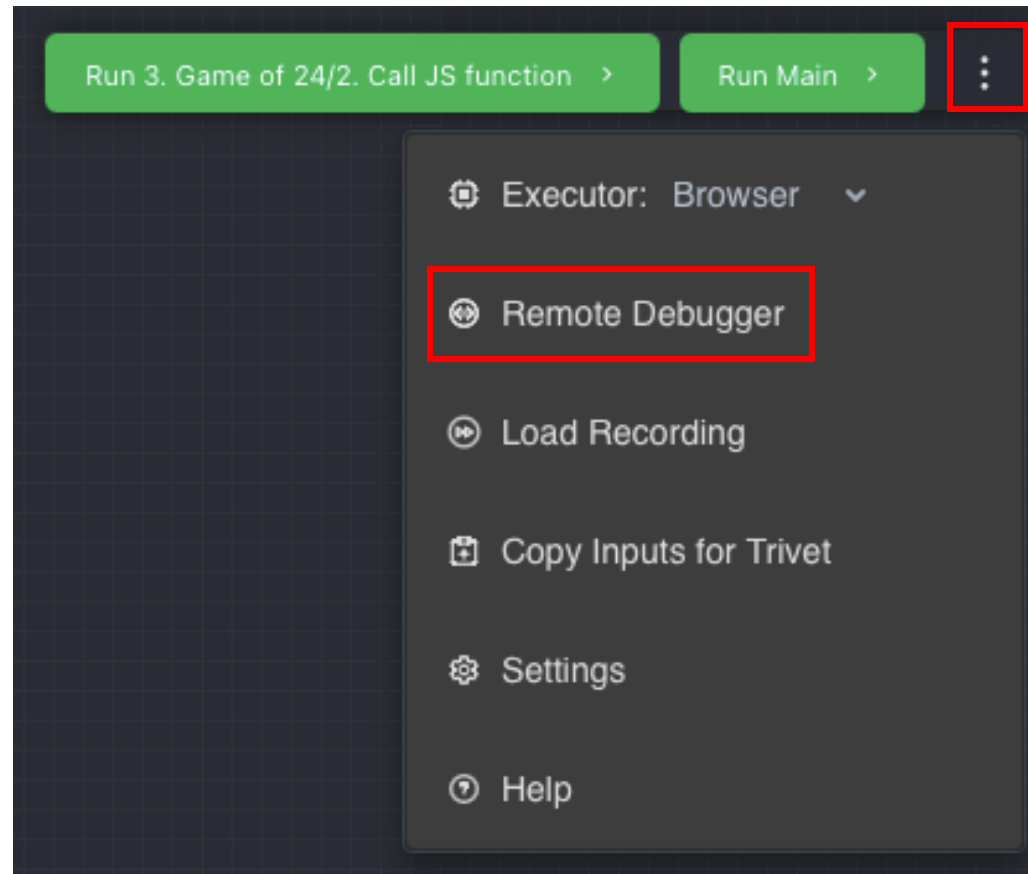
Run this command on Terminal:

```
npm run dev
```

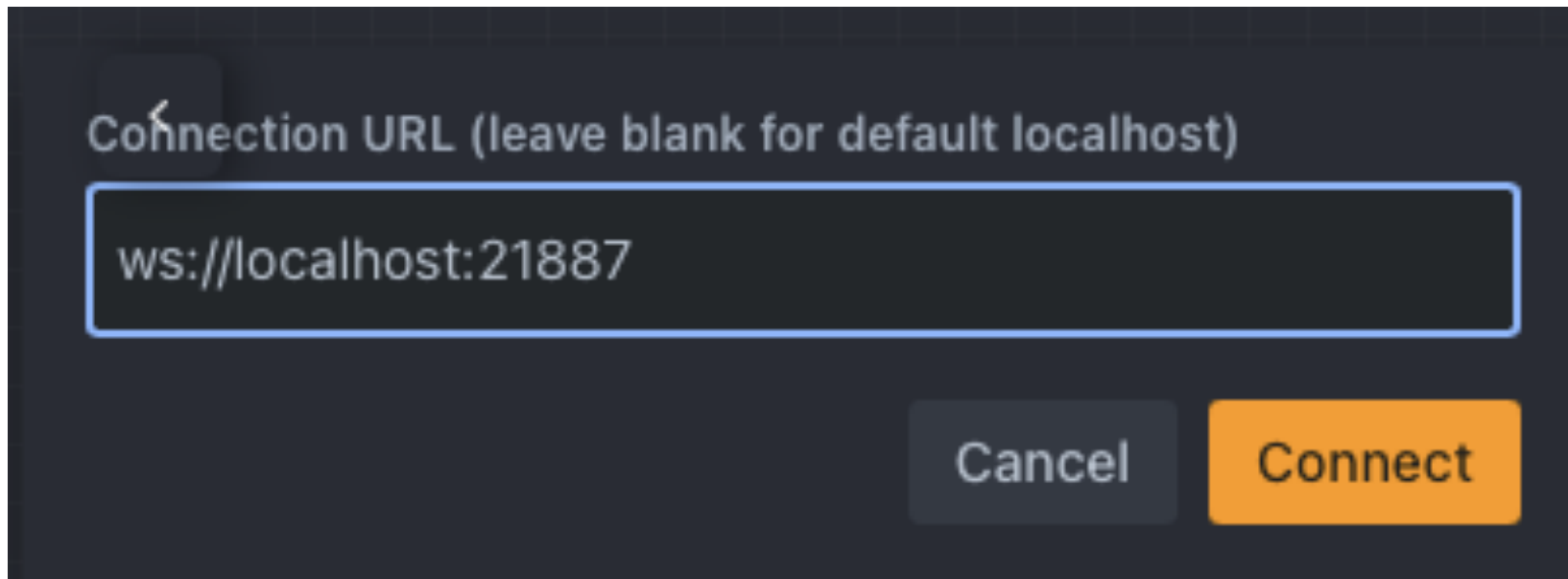


DEBUGGING IN RIVET

4. Listen to debugging server



5. Set port number



A screenshot of a dark-themed dialog box for setting a connection URL. The text "Connection URL (leave blank for default localhost)" is at the top. Below it is a text input field containing "ws://localhost:21887", which is highlighted with a blue border. At the bottom right are two buttons: "Cancel" (disabled, grey) and "Connect" (active, orange).

Connection URL (leave blank for default localhost)

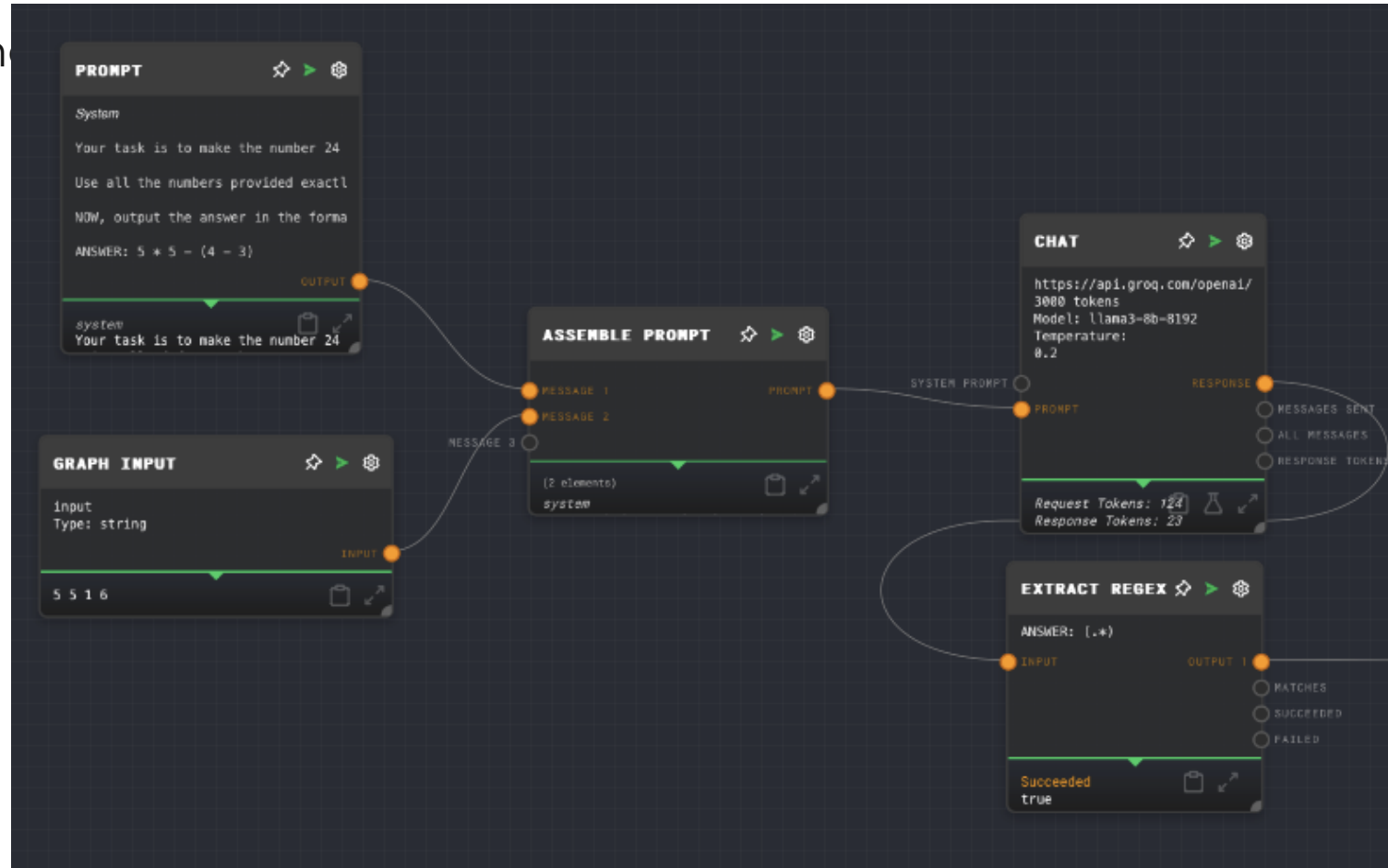
ws://localhost:21887

Cancel Connect

DEBUGGING IN RIVET

6.Observe

Add slide copy h



Hands-on Lab #3

Your Turn

1. Open 1.Template under Game of 24 folder
2. Duplicate the graph and name it as Lab3
3. Connect with NodeJS backend and call JS function from Rivet graph
4. Create a loop that allows the AI to calculate until the answer is correct, using the previous answer as input for the new prompt. Stop after 10 iterations and output 'No solution' if no solution is found.



Q & A

PALO^{IT}

A **global tech innovation consultancy** dedicated to achieving your business goals through next-level product-centric software delivery.

We provide expertise across the full product journey from ideation to launch and scale, using Sustainable IT practices as a commitment to crafting tech as a force for good.

50
Nationalities

18
Offices

10
Countries



25% Organic Growth

\$70m Turnover

700+ Technology experts from across the globe

100% Independently-owned

0 Debt

PRODUCT CONSULTING

1

New Digital Product Innovation

- Validate Product-Market Fit
- Formulate Implementation Plan
- Minimize Development Cost

2

Existing Product Optimization

- Increase Product Adoption
- Identify Product Improvements
- Improve Viability and Profitability



Contact Us

ORGANIZATION CONSULTING

Delivery Acceleration

3

- Reduce Time to Market
- Enable Value-Based Delivery
- Modernize Tech Infrastructure
- Align Business and Product Delivery
- Staff Empowerment

Sustainability For Positive Impact

4

- Sustainability Fundamentals
- ESG Data Management
- Enabling An Impact Mindset
- Tech For Good



Contact Us

CONTACT US



02-180-6121



th-marketing@palo-it.com