



HyperFlow AI Tutorial

미리내소프트 하이퍼플로우AI 튜토리얼

No-code/Low-code Development for Generative AI Application
Kim Jin Soo



1. HyperFlow 시작하기
2. 플로우그래프(Flow-graphs) 소개
3. 플로우그래프를 구축하고 실행하기 (Part 1/2)
4. 플로우그래프를 구축하고 실행하기 (Part 2/2)
5. RAG 기반 지식 DB 및 챗봇 구축
6. 프로덕션 AI앱을 위한 플로우그래프 배포





Getting started with HyperFlow AI

- Create user account
- Setting up an organization
- Add projects

HyperFlow AI 시작



HyperFlow ^β

Sign in as John john@minnus.io

Or sign up with email

Email address
john@hyperflow-ai.com

Password

Must be between 8-20 characters.
Must contain at least one letter.
Must contain at least one number or symbol.

Confirm password

Create account!

Already using HyperFlow?
[Log in now](#)

[Terms of service](#) [Privacy policy](#)

- 계정 생성

HyperFlow AI 시작



Terms of Service
Last Updated: December 23, 2024

1. Introduction

These Terms of Service outline the usage guidelines, data management, service updates, and discontinuation conditions that apply when using HyperFlow Beta Service ("Beta Service") provided by Mirinai Technologies ("the Company"). The Beta Service is a pre-release service still under development. By agreeing to these Terms of Service, users acknowledge that they understand and accept the conditions outlined herein.

2. Welcome to HyperFlow

Thank you for using the HyperFlow Beta Service! These Terms of Service define the relationship between the Company and users registered on the beta waitlist, and provide key information regarding the use of the Beta Service.
The HyperFlow Beta Service can be accessed via various devices, including PCs and mobile devices. The Company provides equal access to all users. However, as the Beta Service is offered for testing purposes prior to its official release, certain features and services may be modified, updated, or discontinued without prior notice.

3. Nature of the Beta Service

The HyperFlow Beta Service is a developmental product, and users acknowledge and agree to the following:

- The Beta Service is provided in a pre-release state and is primarily intended for obtaining feedback on performance and identifying issues.
- The Beta Service is provided "as is" and "as available," with no warranties of any kind from the Company.
- Users accept the following risks associated with using the Beta Service:
 - Unexpected errors or data loss may occur.
 - Service performance and stability may vary.
 - Features may be added, modified, or removed without prior notice.
 - The Beta Service may be interrupted or terminated at any time.

4. Beta User Obligations

As a beta user, you are required to:

1. Provide feedback to the Company regarding issues, errors, or other areas for improvement encountered while using the Beta Service.
2. Report bugs or errors with detailed descriptions and reproducible steps.
3. Respond to surveys or interview requests to share insights for service improvement.
4. Use the Beta Service frequently enough to provide meaningful feedback.

Feedback should include:

- Detailed descriptions of issues encountered
- Steps to reproduce the issues
- Impact of the issues on the intended use of the service
- Suggestions for improvements or additional features
- Observations regarding overall usability and performance

5. Service Stability and Modifications

Users acknowledge and agree to the following:

● 이용약관 확인



HyperFlow AI 시작



- 조직 생성



HyperFlow AI 시작



Organizations encompass...

- Account billing
- User membership & access
- Resource & quota tracking
- API keys and endpoints

HyperFlow AI 시작



HyperFlow β

Create Organization

Organizations

HyperFlow AI

Members

john@hyperflow-ai.com: owner x

+ Invite member

Projects

+ Create Project

owner

This screenshot shows the HyperFlow AI web interface. At the top, there's a navigation bar with links for 'Docs', 'Support', 'Account home', and 'Logout'. Below the navigation is a button to 'Create Organization'. The main area is titled 'Organizations' and contains a single organization entry for 'HyperFlow AI'. This entry includes a section for 'Members' with one listed member ('john@hyperflow-ai.com: owner') and a link to '+ Invite member'. There's also a section for 'Projects' with a link to '+ Create Project'. The 'HyperFlow AI' entry has a status indicator 'owner' at the top right.

- 조직 생성

HyperFlow AI 시작



HyperFlow β

Create Organization

HyperFlow AI

Members

john@hyperflow-ai.com: owner

+ Invite member

Projects

+ Create Project

Create project

John's chatbot

Include examples

Cancel OK

owner

Translated by AI

- 프로젝트 생성

HyperFlow AI 시작



HyperFlow β

Create Organization

Organizations

HyperFlow AI

Members

john@hyperflow-ai.com: owner x

+ Invite member

Projects

John's chatbot →

+ Create Project

owner

A screenshot of the HyperFlow AI web interface. At the top, there's a navigation bar with links for 'Docs', 'Support', 'Account home', and 'Logout'. Below the navigation is a button labeled 'Create Organization'. The main area is titled 'Organizations' and shows a single organization named 'HyperFlow AI'. This organization has one member listed as 'john@hyperflow-ai.com: owner' with a delete icon next to it. There's also a button to '+ Invite member'. Under the 'Projects' section, there's a link to 'John's chatbot' with an arrow icon and a button to '+ Create Project'.

- 프로젝트 생성

HyperFlow AI 시작



A screenshot of the HyperFlow AI web application. The interface includes a header with the logo, title, and navigation links like Docs, Support, Account home, and Logout. Below the header is a toolbar with various icons for file operations. A sidebar on the left contains sections for Flow-graphs, Nodes, and Templates, along with a search bar and examples dropdown. The main area is a large, blank canvas for drawing flow graphs. On the far right, there's a sidebar with a plus sign and minus sign, and a "React for" button.

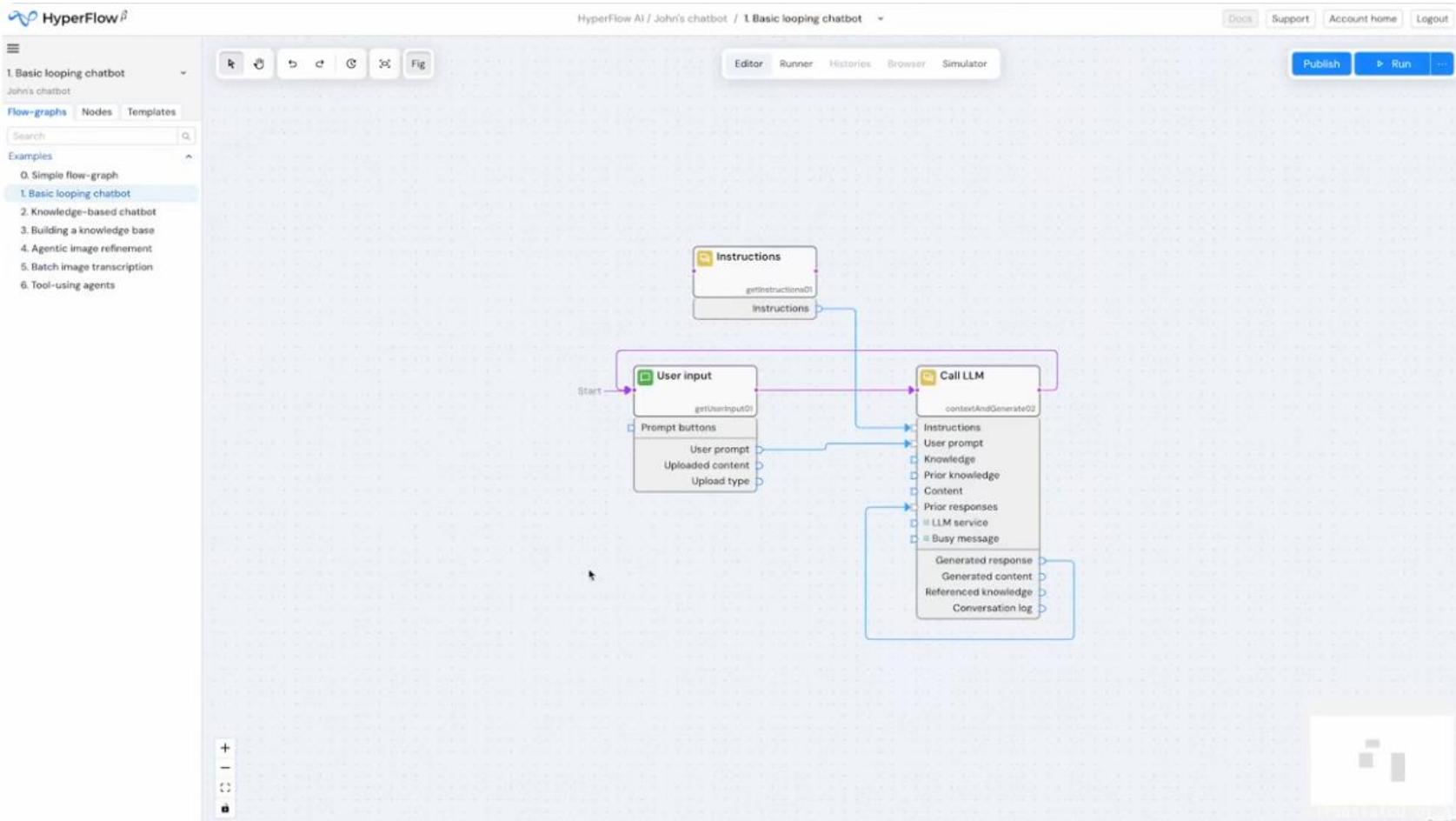
- 프로젝트 생성

HyperFlow AI 시작



- 프로젝트 샘플 예제

HyperFlow AI 시작



- 프로젝트 샘플 예제

HyperFlow AI 시작



HyperFlow Beta Documentation

Made with

HyperFlow Beta Documentation

▽ 4 more properties

- Beta program welcome
- Release notes
- [hyperflow-ai.com/hyperflow - the HyperFlow IDE](https://hyperflow-ai.com/hyperflow)
- [HyperFlow-AI community support on Discord](#)
- [HyperFlow-AI community support on KakaoTalk](#)
- [HyperFlow-AI tutorial video channel on Youtube](#)

- Nodes reference
- Services reference
- Data types
- Chatbot API reference

- 하이퍼플로우 기본 문서 생성



Introduction to Flow-graphs

- No-code development environment
- Draw work-flows ... which HyperFlow runs for you
- Executable work-flows: “Flow-graphs”

Flow-graph uses

- Building & deploying chatbots
- Importing & building knowledge bases
- Realtime agent interaction with external APIs
- Multi-model agentic work-flows
- Orchestration hub for multiple AI systems

플로우그래프(Flow-graphs) 용도



Flow-graph uses

- ▶ Building & deploying chatbots
- ▶ Importing & building knowledge bases
- ▶ Real-time agent interaction with external APIs
- ▶ Multi-model agentic work-flows
- ▶ Orchestration hub for multiple AI systems

- 챗봇 구축 및 배포
- 지식 기반 가져오기 및 구축
- 외부 API와의 실시간 에이전트 상호작용
- 다중 모델 에이전트 워크플로우
- 여러 AI 시스템을 위한 오케스트레이션 허브

플로우그래프(Flow-graphs) 예제



플로우그래프(Flow-graphs) 예제



HyperFlow β

HyperFlow AI / John's chatbot / --

Docs Support Account home Logout

Editor Runner Histories Browser Simulator

John's chatbot

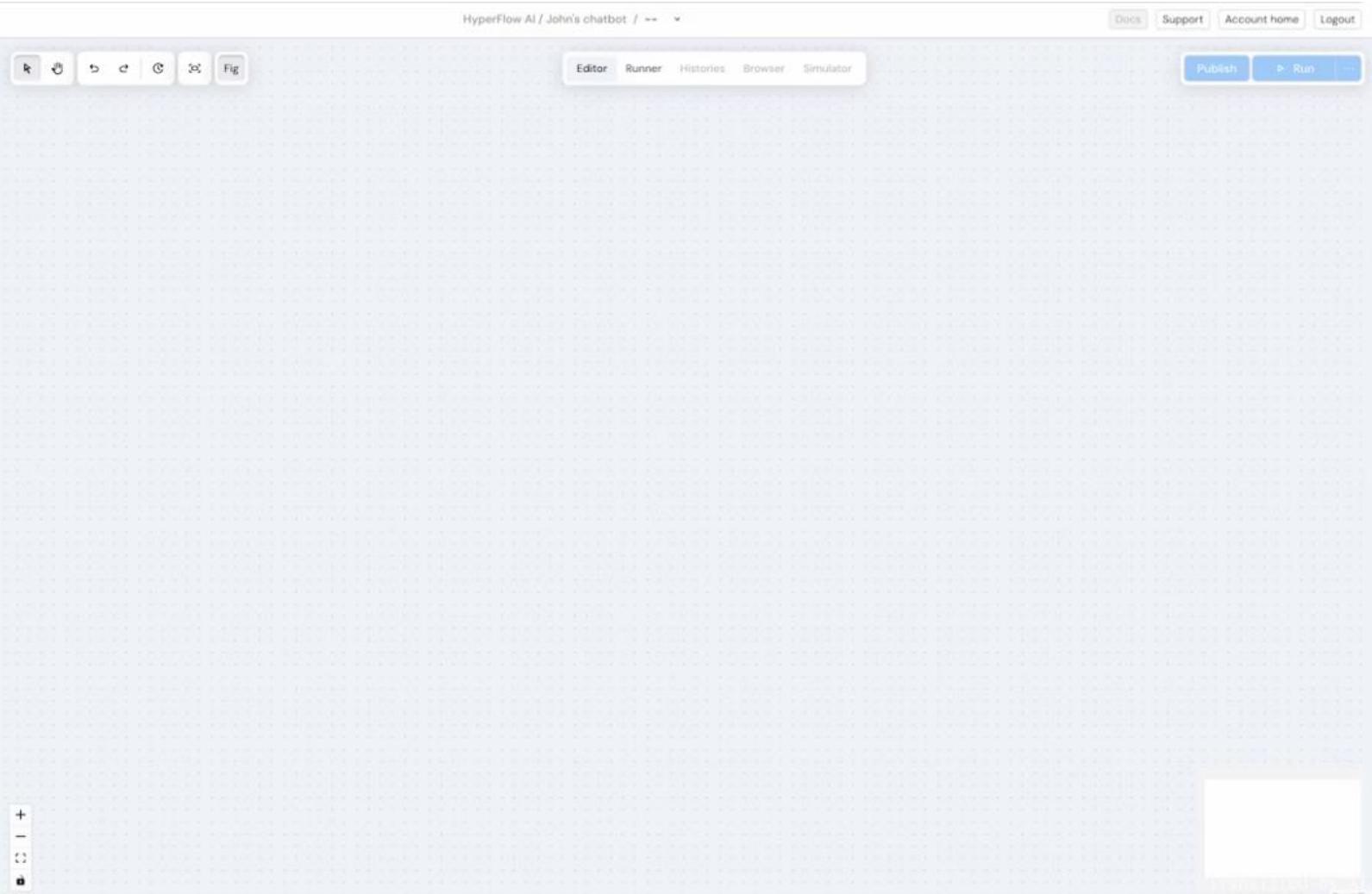
Flow-graphs Nodes Templates

Search

Examples

- 0. Simple flow-graph
- 1. Basic looping chatbot
- 2. Knowledge-based chatbot
- 3. Building a knowledge base
- 4. Agentic image refinement
- 5. Batch image transcription
- 6. Tool-using agents

React File





Ex0. Simple flow-graph

HyperFlow β

HyperFlow AI / John's chatbot / 0. Simple flow-graph

Docs Support Account home Logout

Editor Runner Histories Browser Simulator

Flow-graphs Nodes Templates

Search Examples

0. Simple flow-graph

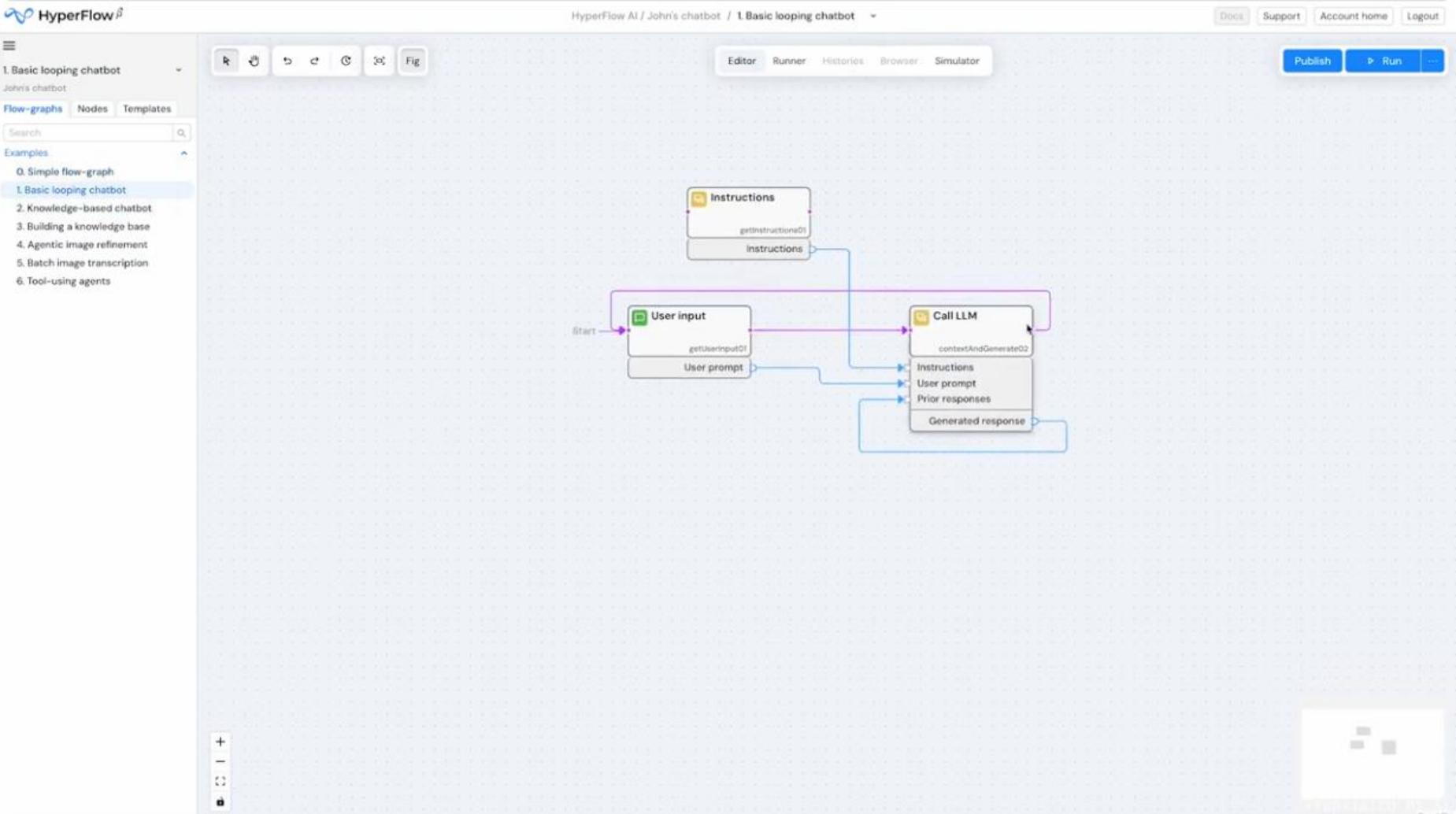
1. Basic looping chatbot
2. Knowledge-based chatbot
3. Building a knowledge base
4. Agentic image refinement
5. Batch image transcription
6. Tool-using agents

```
graph LR; Start((Start)) --> UI[User input<br/>getUserInput01]; UI --> CallLLM[Call LLM<br/>contextAndGenerate02]; CallLLM --> UI; UI -- "User prompt" --> UI;
```

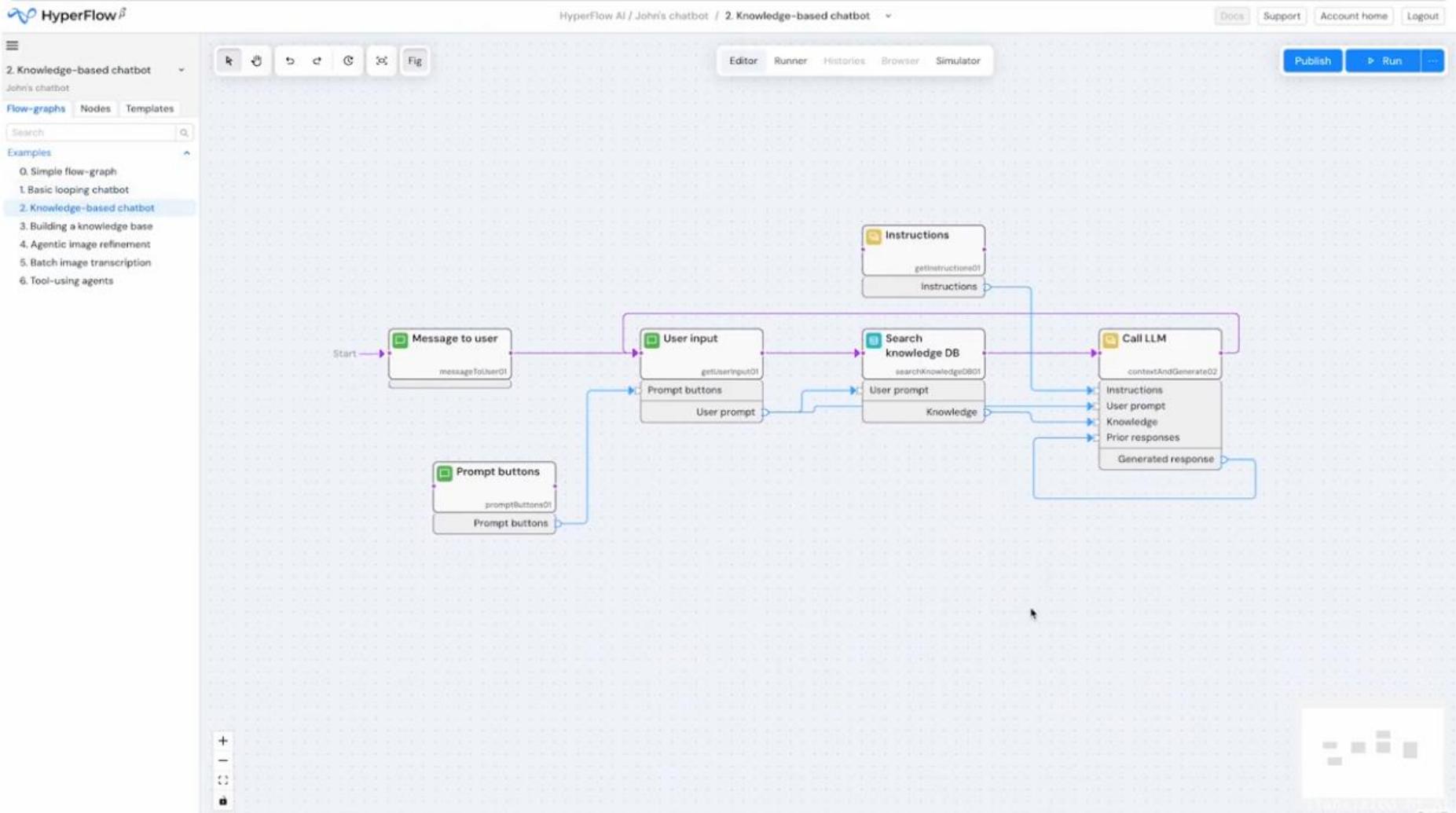
React File



Ex1. Basic looping chatbot

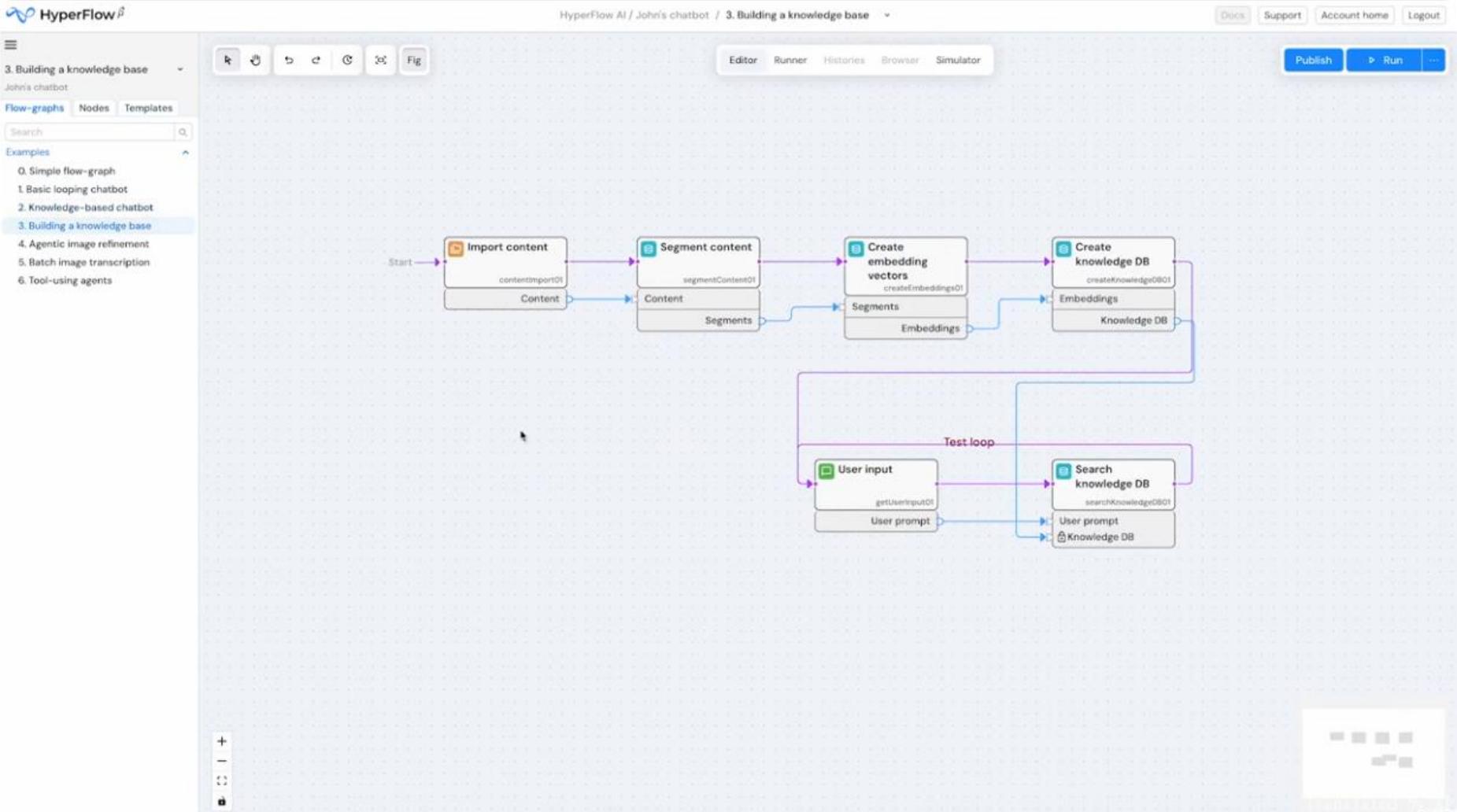


Ex2. Knowledge-based chatbot



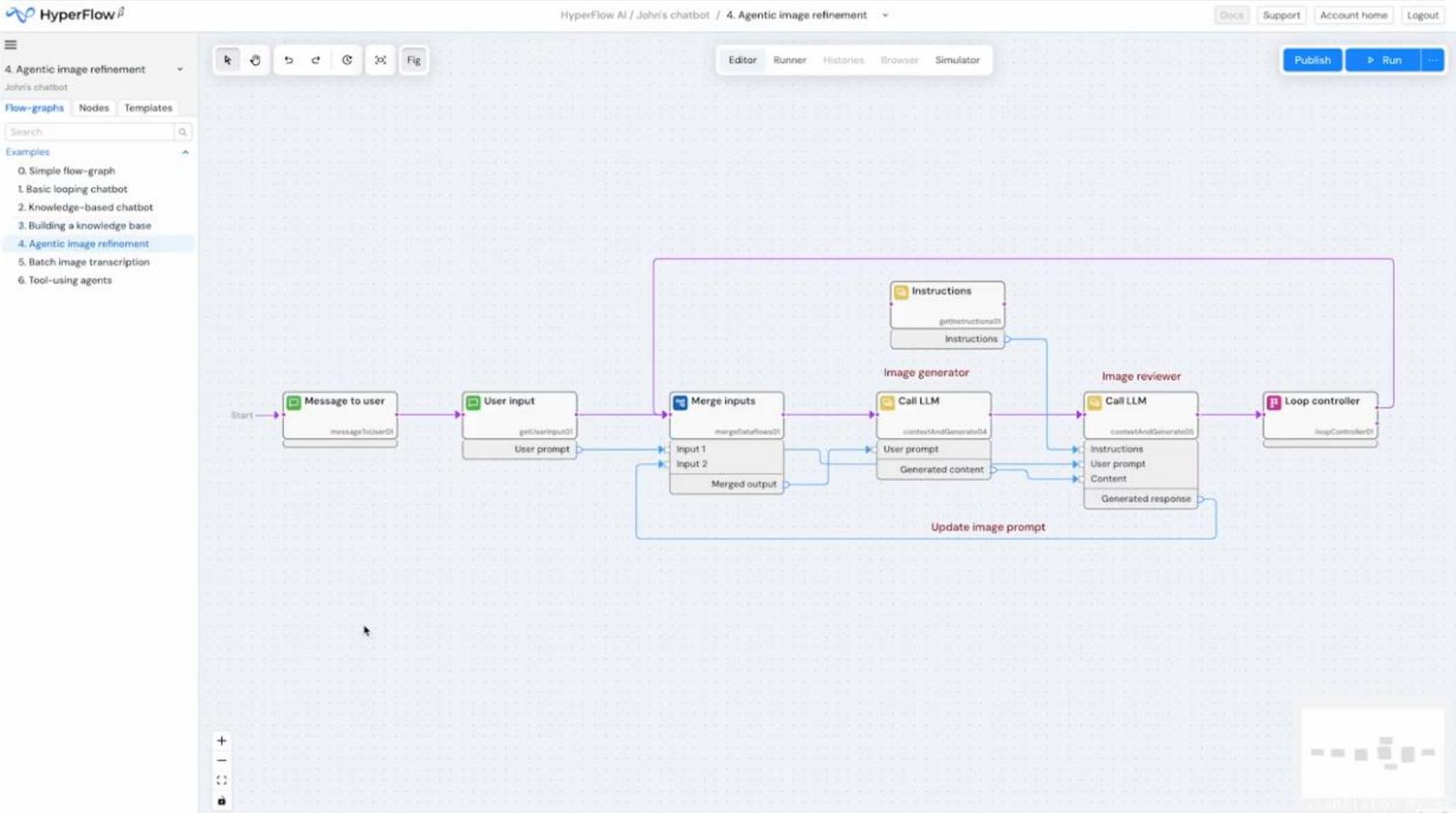


Ex3. Building a knowledge base



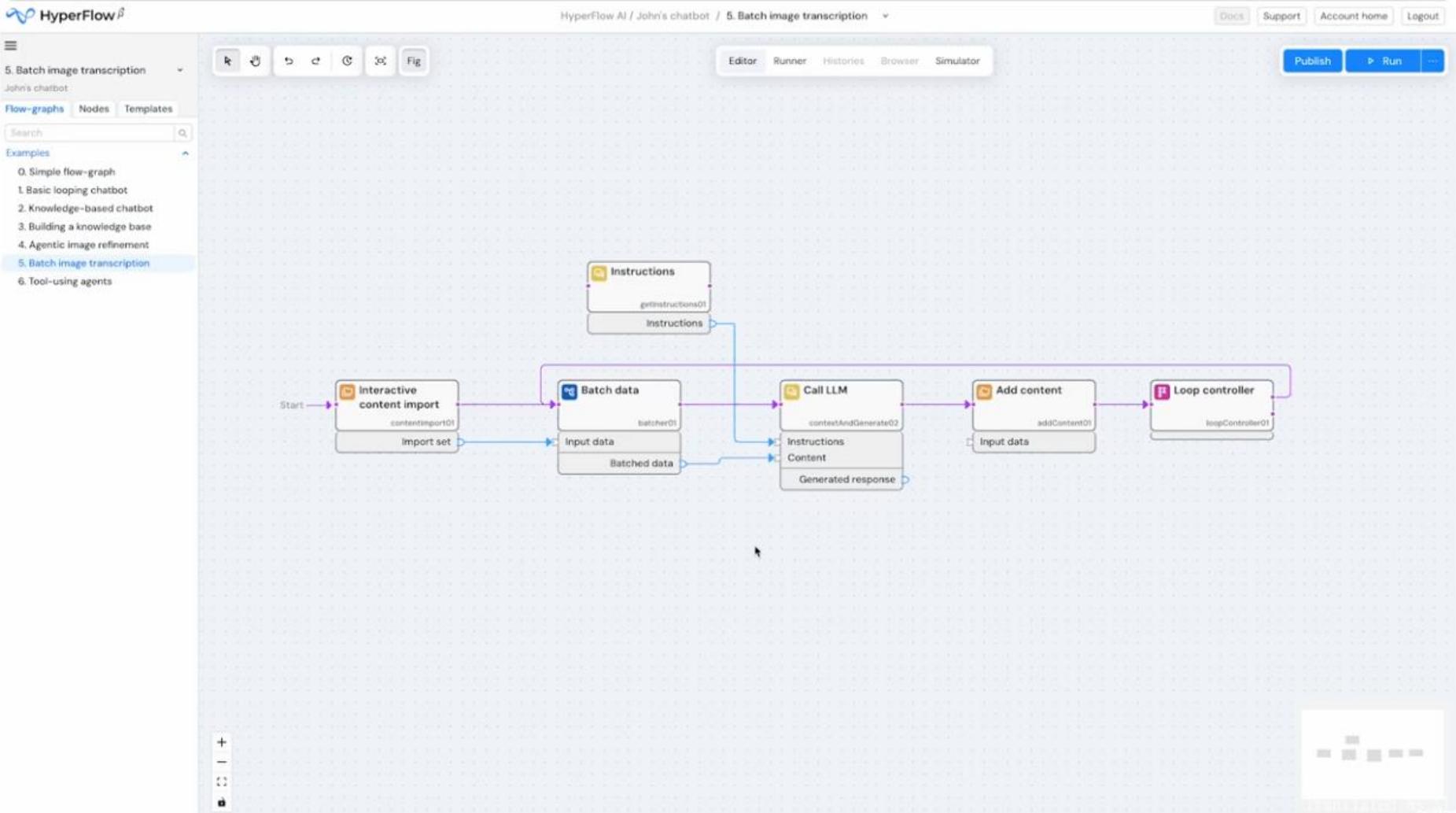


Ex4. Agentic image refinement

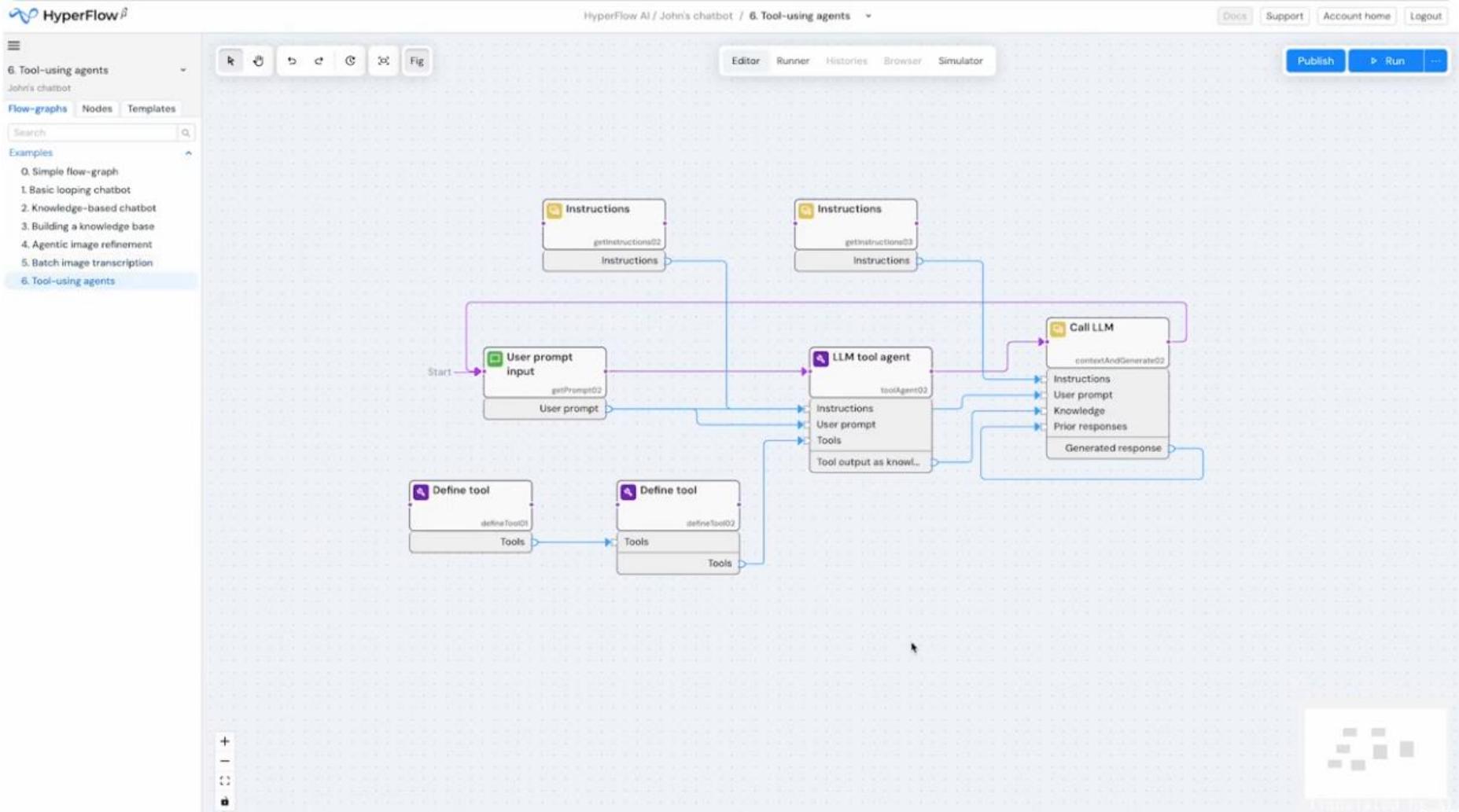




Ex5. Batch image transcription



Ex6. Tool-using agents



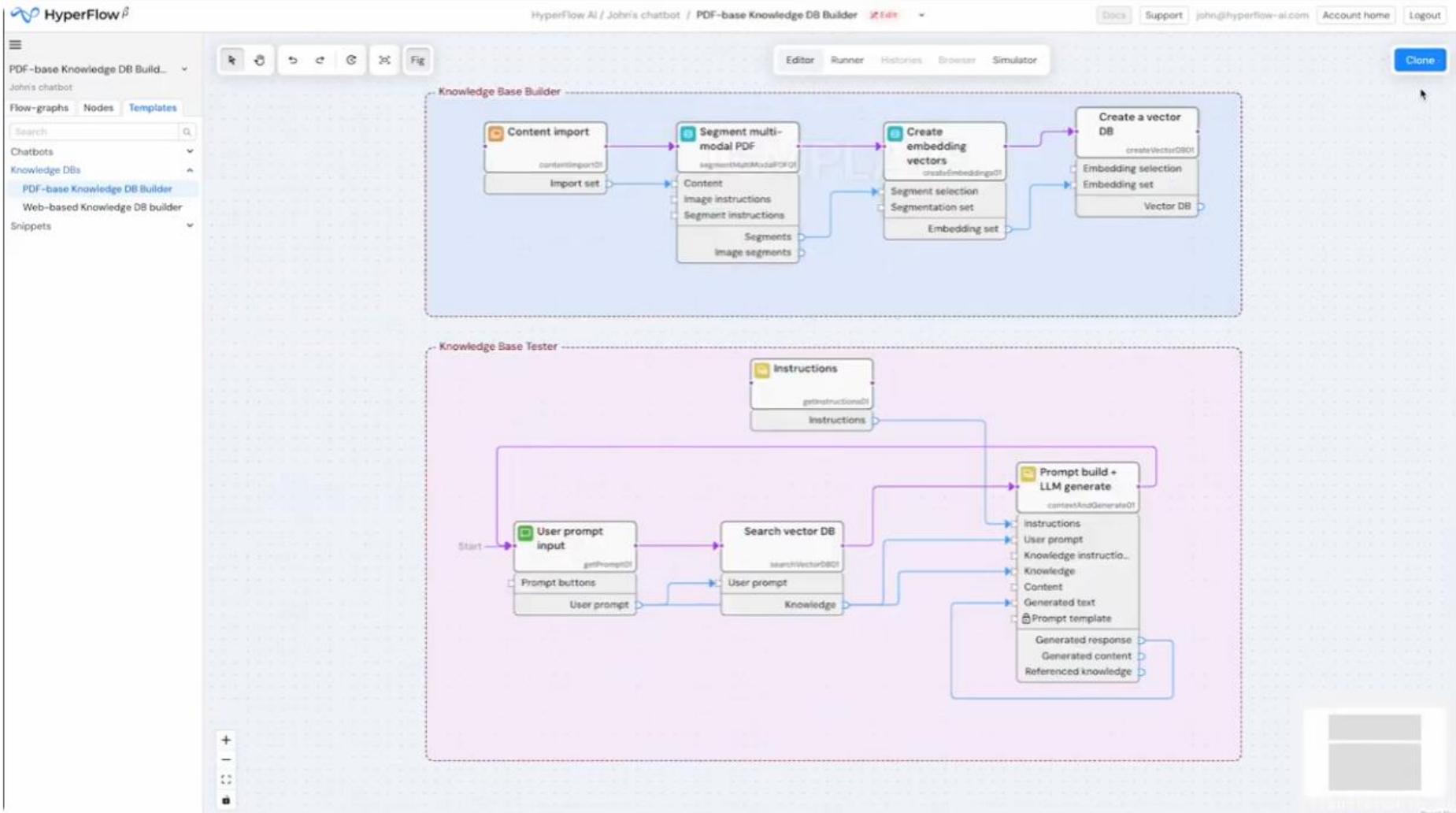
Building and running flow-graphs

플로우그래프(Flow-graphs) 구축 및 실행



The screenshot shows the HyperFlow AI web application interface. At the top, there's a navigation bar with links for 'Docs', 'Support', 'john@hyperflow-ai.com', 'Account home', and 'Logout'. Below the navigation is a toolbar with icons for file operations like 'New', 'Open', 'Save', 'Print', and 'Fig'. The main workspace is titled 'John's chatbot' and contains tabs for 'Flow-graphs', 'Nodes', and 'Templates'. A search bar labeled 'Search' is present. On the left, there's a sidebar with a 'Examples' dropdown menu. The right side of the screen is a large, blank canvas area for building flow graphs.

플로우그래프(Flow-graphs) 구축 및 실행



플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow beta

HyperFlow AI / John's chatbot / Edit

Editor Runner History Browser Simulator

Flow-graphs Nodes Templates

Search

John's chatbot

Chatbots

Knowledge DBs

PDF-base Knowledge DB Builder

Web-based Knowledge DB builder

Snippets

+

-

□

■

Flow-graph details

Name:

Description:

Type: Flow-graph Template

Tags: Chat
Tags define how flow-graphs appear in the main directory. Use dots (.) within a tag to create nested folders in the directory.

Notes:

Private:

Cancel Save

Translated by AI

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow beta

HyperFlow AI / John's chatbot / Edit

Editor Runner Histories FlowView Simulator Publish Run

John's chatbot

Flow-graphs Nodes Templates

Search

Charbots

Knowledge DBs

PDF-base Knowledge DB Builder

Web-based Knowledge DB builder

Snippets

Flow-graph details

Name: John's new bot

Description:

Type: Flow-graph Template

Tags: Chat

Notes:

Private:

Cancel Save

Translated by AI

The screenshot shows the HyperFlow AI platform's user interface. A central modal window titled 'Flow-graph details' is open, prompting for a name ('John's new bot'), description, and tags ('Chat'). The 'Type' is set to 'Flow-graph'. A note about tags specifies they define how flow-graphs appear in the main directory, using dots (.) to create nested folders. The 'Private' toggle is turned on. At the bottom of the modal are 'Cancel' and 'Save' buttons. The background shows a dark workspace with a toolbar at the top and a sidebar on the left containing links to 'Flow-graphs', 'Nodes', 'Templates', and other AI-related tools like 'Charbots' and 'Knowledge DBs'. A watermark 'Translated by AI' is visible in the bottom right corner.

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow β

HyperFlow AI / John's chatbot / Edit

Editor Runner History Browser Simulator

John's chatbot

Flow-graphs Nodes Templates

Search

Chatbots

Knowledge DBs

PDF-base Knowledge DB builder

Web-based Knowledge DB builder

Snippets

Flow-graph details

* Name: John's new bot

Description:

Type: Flow-graph Template

* Tags: My bots

Chat
KB Preparation
RAG

Notes:

Private:

Cancel Save

Translated by AI

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow β

HyperFlow AI / John's chatbot / --- Edit

Editor Runner History Browser Simulator Publish Run

John's chatbot

Flow-graphs Nodes Templates

Search:

Chatbots

Knowledge DBs

PDF-base Knowledge DB builder

Web-based Knowledge DB builder

Snippets

Flow-graph details

Name: John's new bot

Description:

Type: Flow-graph Template

Tags: My bots

Notes:

Private:

Cancel

Translated by AI

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histograms Recover Simulator Publish Run

John's new bot

John's chatbot

Flow-graphs Nodes Templates

Search Examples My bots John's new bot

Flow-graph details

* Name: John's new bot

Description:

Type: Flow-graph Template

* Tags: My bots

Tags define how flow-graphs appear in the main directory. Use dots ('.') within a tag to create nested folders in the directory.

Notes:

Private:

Created by john@hyperflow-ai.com, 2025-01-01 23:43

Cancel Save

Translated by AI

The screenshot displays the HyperFlow AI platform's user interface for managing flow-graphs. A central modal window titled 'Flow-graph details' is open, prompting the user to enter information for a new flow-graph named 'John's new bot'. The modal includes fields for 'Description', 'Type' (set to 'Flow-graph'), 'Tags' (set to 'My bots'), 'Notes', and a 'Private' toggle switch. Below the form is a timestamp indicating it was created by 'john@hyperflow-ai.com' on '2025-01-01 23:43'. At the bottom of the modal are 'Cancel' and 'Save' buttons. The background shows a dark-themed dashboard with various navigation links and sections like 'Editor', 'Runner', 'Histograms', 'Recover', 'Simulator', 'Publish', and 'Run'.

플로우그래프(Flow-graphs) 구축 및 실행



The screenshot shows the HyperFlow AI platform interface. At the top, there's a navigation bar with links for 'Docs', 'Support', 'john@hyperflow-ai.com', 'Account home', and 'Logout'. Below the navigation is a toolbar with icons for 'Editor', 'Runner', 'Histories', 'Browser', and 'Simulator', along with 'Publish' and 'Run' buttons.

The main area is the 'Flow-graphs' editor. On the left, there's a sidebar with a tree view of nodes and templates. The 'Flow-graphs' tab is selected. The tree includes categories like 'Annotation', 'Chat', 'Content', 'Data flow', 'Flow control', 'Knowledge bases', 'LLMs', 'Call LLM', 'Instructions', and 'Tools'. Under 'Content', there are options for 'Add content', 'Import content', 'Select content', and 'Transform content'. A search bar is also present in the sidebar.

The central workspace is currently empty, indicated by a large white area with a faint watermark 'ReactFlow' in the bottom right corner.

플로우그래프(Flow-graphs) 구축 및 실행



The screenshot shows the HyperFlow AI application interface. At the top, there's a navigation bar with the HyperFlow logo, the text "HyperFlow β", and links for "Docs", "Support", "john@hyperflow-ai.com", "Account home", and "Logout". Below the navigation bar is a toolbar with icons for search, filter, and export. The main menu bar includes "Editor", "Runner", "Histories", "Browser", and "Simulator". On the right side, there are "Publish", "Run", and other buttons. The left side features a sidebar with a tree view of available nodes categorized under "Annotation", "Chat", "Content", "Data flow", "Flow control", "Knowledge bases", "LLMs", "Call LLM", "Instructions", and "Tools". The "Nodes" tab is currently selected. The central workspace is empty, indicating no nodes have been placed yet.

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output
Match & message to user
Match & prompt buttons
Message to user
Prompt buttons
User input

Content
Add content
Import content
Select content
Transform content

Data flow
Batch content
Data range
Data set
Data transform
Data value
Key-value store
Launch parameters
Match & select
Merge inputs
Nested loop data

Flow control
Knowledge bases
LLMs
Call LLM
Instructions
Tools

User input

```
graph TD; Start((Start)) --> UserInput[/User input/]; UserInput -- "getUserInput01" --> PromptButtons[Prompt buttons]; UserInput -- "getUserInput01" --> UserPrompt[User prompt]; UserInput -- "getUserInput01" --> UploadedContent[Uploaded content]; UserInput -- "getUserInput01" --> UploadType[Upload type]
```

React Flow

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

User input

Call LLM

Start

```
graph TD; Start((Start)) --> UserInput[/User input/]; UserInput -- "getUserInput()"; UserInput --> PromptButtons[Prompt buttons]; UserInput --> CallLLM[Call LLM]; UserInput --> End((End));
```

User prompt
Uploaded content
Upload type

contentAndGenerate()
Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
Prompt template
LLM service
Generated response
Generated content
Referenced knowledge
Conversation log

John's new bot

Annotation

Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation

Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

Call LLM

Prompt buttons

User prompt

Uploaded content

Upload type

contextAndGenerate()

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

```
graph LR; Start((Start)) --> UserInput[User input]; UserInput --> CallLLM[Call LLM];
```

React Flow

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

Call LLM

Prompt buttons

User prompt

Uploaded content

Upload type

contextAndGenerate()

Generated response

Generated content

Referenced knowledge

Conversation log

```
graph LR; Start((Start)) --> UI[User input]; UI --> CallLLM[Call LLM]; UI --> PromptButtons[Prompt buttons]; UI --> CallLLM; PromptButtons --> CallLLM;
```

React Flow

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

Call LLM

Start

getUserInput()

contextAndGenerate()

User prompt

Uploaded content

Upload type

Instructions

User prompt

Prior knowledge

Content

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

John's new bot

Flow-graphs Nodes Templates

Search

Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

Prompt b

Call LLM

contextAndGenerateOf

Instructions

User prompt

Prior knowledge

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

Start

Set as start node

Toggle breakpoint

Clear all breakpoints

Cut

Copy

Duplicate

Delete

Unlock all parameters

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

User input #1 get userInput01

Mode: text

User input:

Start

```
graph LR; Start((Start)) --> UserInput[User input]; UserInput -- "get(UserInput01)" --> CallLLM[Call LLM]; UserInput -- "Prompt buttons" --> CallLLM; UserInput -- "User prompt" --> CallLLM; UserInput -- "Uploaded content" --> CallLLM; UserInput -- "Upload type" --> CallLLM; CallLLM -- "contextAndGenerate01" --> GeneratedResponse[Generated response]; CallLLM -- "Instructions" --> GeneratedContent[Generated content]; CallLLM -- "User prompt" --> ReferencedKnowledge[Referenced knowledge]; CallLLM -- "Knowledge" --> ConversationLog[Conversation log]; GeneratedResponse --> GeneratedContent; GeneratedContent --> ReferencedKnowledge; ReferencedKnowledge --> ConversationLog;
```

Annotations

Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input #1 get userInput01

Mode

Text prompt input

File upload

Optional upload + text input

Clear selection

Start

User input

getUserInput01

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate01

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

```
graph LR; Start((Start)) --> UI[User input]; UI -- "getUserInput01" --> PB[Prompt buttons]; PB -- "User prompt" --> CallLLM[Call LLM]; PB -- "Uploaded content" --> CallLLM; PB -- "Upload type" --> CallLLM; CallLLM -- "contextAndGenerate01" --> Instructions[Instructions]; CallLLM -- "User prompt" --> Instructions; CallLLM -- "Knowledge" --> Instructions; CallLLM -- "Prior knowledge" --> Instructions; CallLLM -- "Content" --> Instructions; CallLLM -- "Prior responses" --> Instructions; CallLLM -- "LLM service" --> Instructions; CallLLM -- "Busy message" --> Instructions; CallLLM -- "Generated response" --> GeneratedResponse[Generated response]; CallLLM -- "Generated content" --> GeneratedContent[Generated content]; CallLLM -- "Referenced knowledge" --> ReferencedKnowledge[Referenced knowledge]; CallLLM -- "Conversation log" --> ConversationLog[Conversation log]
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

- Add content
- Import content
- Select content
- Transform content

Data flow

- Batch content
- Data range
- Data set
- Data transform
- Data value
- Key-value store
- Launch parameters
- Match & select
- Merge inputs
- Nested loop data

Flow control

Knowledge bases

LLMs

- Call LLM
- Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

User prompt

Prior knowledge

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

- Add content
- Import content
- Select content
- Transform content

Data flow

- Batch content
- Data range
- Data set
- Data transform
- Data value
- Key-value store
- Launch parameters
- Match & select
- Merge inputs
- Nested loop data

Flow control

Knowledge bases

LLMs

- Call LLM
- Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

- User prompt
- Prior knowledge
- Prior responses
- LLM service
- Busy message

Generated response

- Generated content
- Referenced knowledge
- Conversation log

Call LLM #2 contextAndGenerate()

All locks

Role Chat

Generation options

Prompt template

LLM service Select service...

- OpenAI
- Cohere
- Ollama
- Anthropic
- Groq
- GoogleAI

```
graph LR; Start((Start)) --> UI[User input  
getUserInput()]; UI --> PB[Prompt buttons  
User prompt  
Uploaded content  
Upload type]; PB --> CallL1[Call LLM  
contextAndGenerate()]; CallL1 --> UserPrompt[User prompt]; CallL1 --> Content[Uploaded content]; CallL1 --> Type[Upload type]; CallL1 --> Instructions[Instructions  
User prompt  
Prior knowledge  
Prior responses  
LLM service  
Busy message]; CallL1 --> GeneratedResponse[Generated response  
Generated content  
Referenced knowledge  
Conversation log]; CallL1 --> GeneratedContent[Generated content]; CallL1 --> ReferenceKnowledge[Referenced knowledge]; CallL1 --> ConversationLog[Conversation log];
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

User prompt

Prior knowledge

Content

Prior responses

LLM service

Busy message

LLM model

Generated response

Generated content

Referenced knowledge

Conversation log

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

- Add content
- Import content
- Select content
- Transform content

Data flow

- Batch content
- Data range
- Data set
- Data transform
- Data value
- Key-value store
- Launch parameters
- Match & select
- Merge inputs
- Nested loop data

Flow control

Knowledge bases

LLMs

- Call LLM
- Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Start

Call LLM

contextAndGenerate()

Instructions

- User prompt
- Prior knowledge
- Content
- Prior responses
- LLM service
- Busy message
- LLM model

Generated response

Generated content

Referenced knowledge

Conversation log

LLM service: openai

LLM parameters:

LLM model: Select option...

- gpt-4-audio-preview-2024-10-01
- gpt-40-realtime-preview
- gpt-40-realtime-preview-2024-10-01
- gpt-4-turbo
- gpt-4-mini-2024-07-18
- gpt-4-106-preview
- gpt-4-mini
- gpt-3.5-turbo
- gpt-3.5-turbo-0125
- gpt-3.5-turbo-instruct
- gpt-3.5-turbo-16k
- gpt-4-0125-preview
- gpt-4-turbo-preview
- gpt-40-2024-05-09
- chatgpt-40-latest
- gpt-4
- gpt-4-0613
- gpt-3.5-turbo-1106
- gpt-40-audio-preview
- gpt-3.5-turbo-instruct-0914
- gpt-40-realtime-preview-2024-12-17
- gpt-40-mini-realtime-preview
- gpt-40-mini-realtime-preview-2024-12-17
- gpt-40-2024-08-06
- gpt-4
- gpt-4-2024-11-20
- gpt-40-audio-preview-2024-12-17
- gpt-40-mini-audio-preview
- gpt-40-mini-audio-preview-2024-12-17
- gpt-4-turbo-2024-04-09

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

Call LLM

Start

gotUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

contextAndGenerate()

Instructions

User prompt

Prior knowledge

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

User prompt

Prior knowledge

Prior responses

Content

LLM service

Busy message

LLM model

Generated response

Generated content

Referenced knowledge

Conversation log

Role Chat

Generation options

Prompt template

LLM service OpenAI

Temperature 1.23

Max tokens maximum

Top P 1

Frequency penalty 0

Presence penalty 0

Stop sequences

LLM model Select option...

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation

Chat

Chat output

Match & message to user

Match & prompt buttons

Message to user

Prompt buttons

User input

Content

Add content

Import content

Select content

Transform content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

User prompt

Prior knowledge

Prior responses

LLM service

Busy message

LLM model

Generated response

Generated content

Referenced knowledge

Conversation log

```
graph LR; Start((Start)) --> UI[User input  
getUserInput()]; UI --> PB[Prompt buttons  
User prompt  
Uploaded content  
Upload type]; PB --> CallLLM[Call LLM  
contextAndGenerate()]; CallLLM --> GeneratedContent[Generated content  
Generated response  
Referenced knowledge  
Conversation log]; CallLLM --> CallLLM
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

User input #1 get userInput() Step 1

All locks

Mode: Text prompt input

User input: Why are there 7 colors in the rainbow?

Run session tags:

User input #1 get user input() Step 1

Waiting on settings OK...

OK

Start Flow

Trained by

```
graph LR; Start((Start)) --> UI1[User input: get userInput()]; UI1 --> UI2[Prompt buttons: User prompt, Uploaded content, Upload type]; UI1 --> CallLLM[Call LLM: contextual, Instructions, User prompt, Knowledge, Prior knowledge, Content, Prior response, Generated response, Generated reference, Reference, Conversation];
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow β

HyperFlow AI / John's chatbot / John's new bot Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

User input #1 gettextInput() Step 1

Parameters

Call LLM #2 contextAndGenerate() Step 2

All locks 0

Role Chat

Generation options

Prompt template

LLM service OpenAI

LLM parameters

LLM model Select option...

OK

Waiting on settings OK...

React Flow

Transferred by

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

User input #1 getuserInput() Step 1

Parameters

Call LLM #2 contextAndGenerate()

Step 2

All locks ⊕ ⊖

Role Chat

Generation options

Prompt template

LLM service OpenAI

LLM parameters

LLM model Select option...

gpt-4-audio-preview-2024-10-01
gpt-4-realtime-preview
gpt-4-realtime-preview-2024-10-01
gpt-4-turbo
gpt-4-mini-2024-07-18
gpt-4-1106-preview
gpt-4-mini
gpt-3.5-turbo
gpt-3.5-turbo-0125
gpt-3.5-turbo-instruct
gpt-3.5-turbo-16k
gpt-4-0125-preview
gpt-4-turbo-preview
gpt-4-2024-05-13
chatgpt-4-latest
gpt-4
gpt-4-0613
gpt-3.5-turbo-1106
gpt-4-audio-preview
gpt-3.5-turbo-instruct-0914

User input

getuserInput()

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate()

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Start

```
graph LR; Start((Start)) --> UserInput[User input]; UserInput --> CallLLM[Call LLM]; CallLLM --> GeneratedResponse[Generated response];
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

User input #1 getGenerating()

Call LLM #2 contextAndGenerate()

Step 2

All locks

Role Chat

Generation options

Prompt template

LLM service OpenAI

LLM parameters

LLM model gpt-4o-mini

OK

Run session tags

User input #1 getGenerating()

Step 1

Parameters

Call LLM #2 contextAndGenerate()

Step 2

Parameters

Composite prompt

Type Content

L user **User Query** Why are there 7 colors in the rainbow

Approx. tokens 9

Source step 1

Call LLM: ...

Start flow

Transferred by

```
graph LR; Start((Start)) --> UserInput[User input]; UserInput --> CallLLM[Call LLM]; CallLLM --> CompositePrompt[Composite prompt];
```

The screenshot displays the HyperFlow AI platform's interface for building and executing flow graphs. On the left, a flow graph is shown with nodes: 'User input' (with methods 'getGenerating()' and 'Prompt buttons' containing 'User prompt', 'Uploaded content', and 'Upload type'), 'Call LLM' (with methods 'contextAndGenerate()' and a dropdown menu for 'Instructions', 'User prompt', 'Knowledge', 'Prior knowledge', 'Content', and 'Prior responses' which lists 'Generated response', 'Generated content', 'Referenced knowledge', and 'Conversation log'), and 'Composite prompt' (containing a user query: "Why are there 7 colors in the rainbow"). The right side of the screen shows detailed configurations for the 'Call LLM' and 'Composite prompt' steps, including role selection, generation options, prompt templates, LLM services, parameters, and models. The top navigation bar includes links for 'Docs', 'Support', 'john@hyperflow-ai.com', 'Account home', and 'Logout'.

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow

HyperFlow AI / John's chatbot / John's new bot

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

User input #1 getUserInput01 Step 3

All locks

Mode: Text prompt input

User input:

Call LLM #2 contextAndGenerate01 Step 2

Parameters

Composite prompt

Type: Content

1 user **User Query** Why are there 7 colors in the rainbow

Approx. tokens: 9 Source step: 1

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

- Light Dispersion: When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
- Newton's Contribution: Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.
- Additional Science: Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

Start Flow

User input #1 getUserInput01 Step 3

Waiting on settings OK...

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Editor Runner Histories Browser Simulator

Run session tags

User input #1 get userInput01 Step 3

Mode: Text prompt input

User input:

Call LLM #2 contextAndGenerate01 Step 2

Role: Chat

Generation options:

Prompt template:

LLM service: OpenAI

LLM parameters:

LLM model: gpt-4o-mini

Composite prompt

Type	Content	Approx. tokens	Source step
1. user	**User Query:** Why are there 7 colors in the rainbow	9	1

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

1. **Light Dispersion:** When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
2. **Newton's Contribution:** Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.

Start Flow

```
graph LR; Start((Start)) --> UI1[User input: get userInput01]; UI1 --> UI2[User input: User prompt, Uploaded content, Upload type]; UI2 --> LLM1[Call LLM: contextAndGenerate01]; LLM1 --> LLM2[Call LLM: Instructions, User prompt, Knowledge, Prior knowledge, Content, Prior response, Generated response, Generated reference, Generated knowledge, Generated conversation]; LLM2 --> Text[Generated text: The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.  
1. Light Dispersion: When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.  
2. Newton's Contribution: Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.]
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator

Pause

User input #1 getInput()

Mode: Text prompt input

User input:

Call LLM #2 contextAndGenerate()

Parameters

Composite prompt

Type: Content

Content: user: **User Query** Why are there 7 colors in the rainbow

Approx. tokens: 9

Source step: 1

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

1. **Light Dispersion:** When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
2. **Newton's Contribution:** Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.
3. **Additional Science:** Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

Start Flow

```
graph LR; Start((Start)) --> UserInput1[User input: #1 getInput()]; UserInput1 --> PromptButtons[Prompt buttons: User prompt, Uploaded content, Upload type]; UserInput1 --> CallLLM1[Call LLM: #2 contextAndGenerate()]; PromptButtons --> CallLLM1; CallLLM1 --> GeneratedText1[Generated text: The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.]; GeneratedText1 --> Metadata1[Metadata]; GeneratedText1 --> UserInput2[User input: #1 getInput()]; UserInput2 --> Pending[Waiting on settings OK...]
```

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Step 3

Editor Runner Histories Browser Simulator

Pause

User input #1 getInput01 Step 3

All locks

Mode: Text prompt input

User input: Why is it in the shape of an arc?

Call LLM #2 contextAndGenerate01 Step 2

Parameters

Composite prompt

Type: Content

user: **User Query** Why are there 7 colors in the rainbow

Approx. tokens: 9 Source step: 1

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

- Light Dispersion: When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
- Newton's Contribution: Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.
- Additional Science: Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

Issue: #1-4a-mini-2024-07-19 | Channel: 2025-01-01.2234 | Elapsed: 3.7ms | Prompt Tokens: 21 | Completion Tokens: 254

User input #1 getInput01 Step 3

Waiting on settings OK...

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Editor Runner Histories Brower Simulator

Run session tags

Call LLM #2 contextAndGenerate01 Step 4

Role Chat
Generation options
Prompt template
LLM service OpenAI

OK

User input `get userInput()`

Prompt buttons

User prompt
Uploaded content
Upload type

Call LLM `contextAndGenerate01`

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses

Generated response
Generated content
Referenced knowledge
Conversation log

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

- Light Dispersion:** When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
- Newton's Contribution:** Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.
- Additional Science:** Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

User input `#1 getUserInput()`

Parameters

Call LLM `#2 contextAndGenerate01`

Step 4

Waiting on settings OK...

Start Flow

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot Edit

Editor Runner Histories Browser Simulator

Run session tags

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

1. **Light Dispersion:** When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.
2. **Newton's Contribution:** Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.
3. **Additional Science:** Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

Model: gpt-4o-mini-2024-07-18	Created: 2026-01-01 23:54	Elapsed: 3.78s	Prompt Tokens: 21	Completion Tokens: 254
-------------------------------	---------------------------	----------------	-------------------	------------------------

User input #1 gettextInput()

Call LLM #2 contextAndGenerate()

Generated response

Composite prompt

Type	Content	Approx tokens	Source step
1 - user	**User Query** Why are there 7 colors in the rainbow?	9	1
2 - generator	The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.	250	2
3 - user	**User Query** Why is it in the shape of an arc?	10	3

Call LLM: ...

The screenshot shows the HyperFlow AI interface. On the left, a flow graph starts with a 'User input' node, followed by a 'Call LLM' node, and finally a 'Generated response' node. The 'Call LLM' node has several parameters: Role (Chat), Generation options (empty), Prompt template (empty), LLM service (OpenAI), LLM parameters (empty), and LLM model (gpt-4o-mini). A modal window for 'Call LLM #2 contextAndGenerate()' is open, showing the same configuration. To the right, the generated response is displayed, explaining the formation of a rainbow. Below the response, the 'Composite prompt' table shows the user query and the generated response. At the bottom, a partial LLM call is shown.

플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow ^β

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator

Run session tags

User input #1 getInput() Step 5

All locks: 0

Mode: Text prompt input

User input:

Call LLM #2 contextAndGenerate() Step 4

Parameters

Composite prompt

Type	Content	Approx. tokens	Source step
1. user	**User Query** Why are there 7 colors in the rainbow	9	1
2. generator	The rainbow is typically described as having seven distinct colors – red, orange, yellow, green, blue, indigo, and violet – largely due to the way light is refracted and dispersed in raindrops.	250	2
3. user	**User Query** Why is it in the shape of an arc?	10	3

Generated text

The characteristic arc shape of a rainbow is a result of the way light is refracted, reflected, and dispersed in raindrops. Here's how this process creates the arc:

1. **Refraction:** When sunlight enters a raindrop, it is refracted (bent) at the surface of the drop. This bending of light separates the different colors because each color has a different wavelength and is refracted by a different amount, with red light bending the least and violet bending the most.
2. **Internal Reflection:** After refraction, the light travels through the drop and reflects off the inner surface of the raindrop. This reflection contributes to the dispersion of colors through a process called total internal reflection.
3. **Exit and Second Refraction:** As the light exits the raindrop, it undergoes refraction again, further spreading the colors.
4. **Circular Shape:** The refraction and reflection processes inside the raindrop create a circle of light. However, from a typical ground perspective, we only see the upper half of this circle. The rest of the circle is beneath the horizon or blocked by the ground, giving us the familiar arc shape.
5. **Observer's Point of View:** The rainbow forms a circle centered on the observer's shadow. The water droplets that are scattering and refracting the light are located in a certain region of the sky opposite the sun, which creates the arc shape that is observed.

In summary, the combination of light bending and reflecting in raindrops, along with our point of view, results in the observed shape of a rainbow as an arc.

Metadata

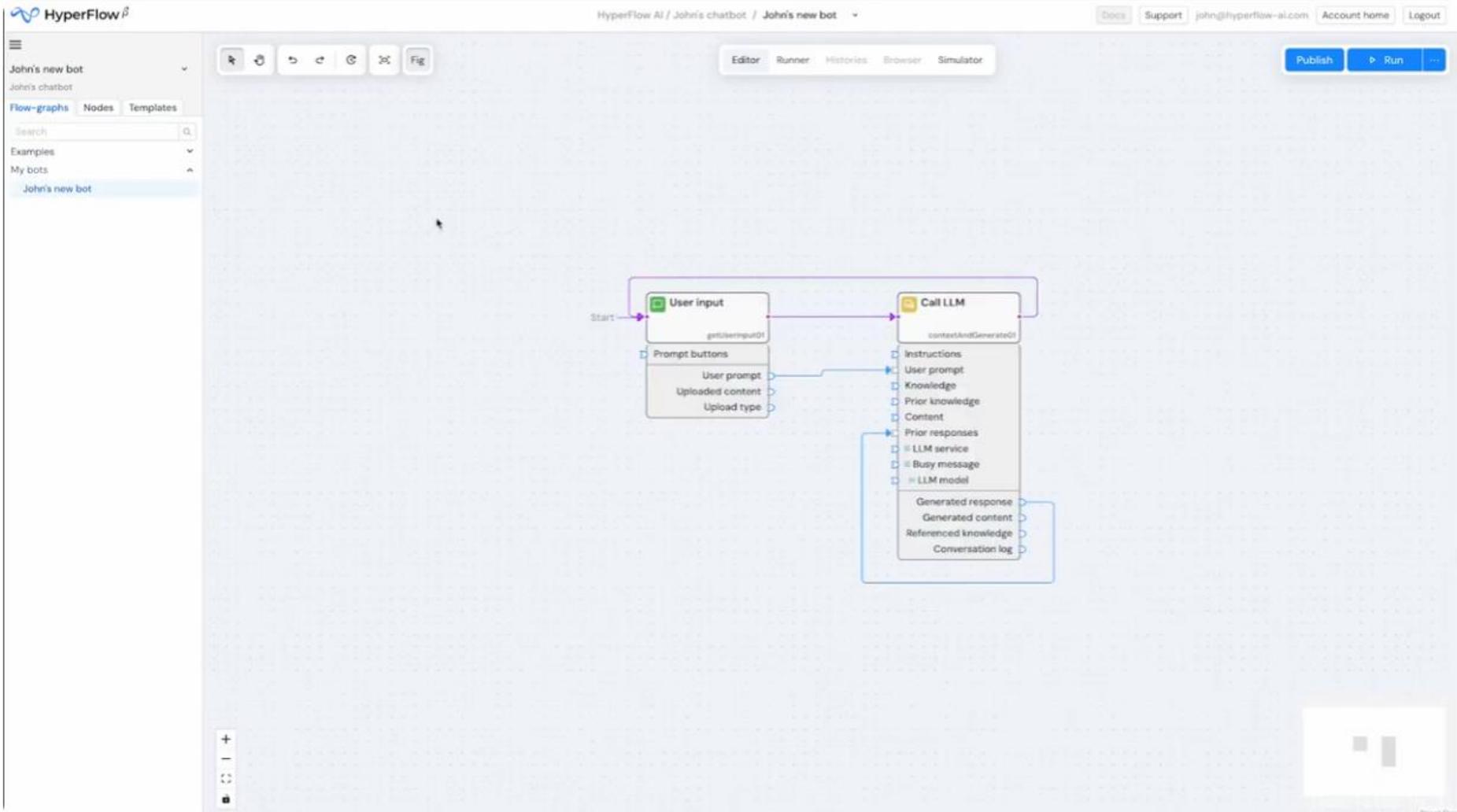
Model: gpt-4e-mini-2024-07-18	Created: 2025-01-01 23:58	Elapsed: 7.93s	Prompt Tokens: 298	Completion Tokens: 330
-------------------------------	---------------------------	----------------	--------------------	------------------------

User input #1 getInput() Step 5

Waiting on settings OK...

The screenshot shows the HyperFlow AI platform interface. At the top, there are tabs for Editor, Runner, Histories, Browser, and Simulator. The Runner tab is active. In the center, there is a detailed configuration window for a 'User input' step (Step 5). This window includes fields for Mode (Text prompt input), User input (a text area), and a composite prompt table. The table lists three entries: a user query about rainbow colors, a generator response describing the physical process, and another user query about the arc shape. To the right of this window is a large panel displaying 'Generated text' explaining the physics of rainbows. Below these are sections for 'Metadata' (model information) and another 'User input' step (Step 5) showing a status message. On the left side of the main workspace, there is a flow graph diagram showing the connection between 'User input' and 'Call LLM' nodes. The flow graph has nodes like 'User input', 'Prompt buttons', 'Call LLM', and various data fields like 'User prompt', 'Uploaded content', and 'Upload type'. Arrows indicate the flow of data between these nodes.

플로우그래프(Flow-graphs) 구축 및 실행



플로우그래프(Flow-graphs) 구축 및 실행



HyperFlow

HyperFlow AI / John's chatbot / John's new bot

Editor Runner Histories Browser Simulator Publish Run

John's new bot

Flow-graphs Nodes Templates

Search Examples My bots John's new bot

Flow-graph run history

1. user ***User Query*** Why are there 7 colors in the rainbow

Generated text

The rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops.

1. **Light Dispersion:** When sunlight passes through raindrops, it gets bent and splits into its constituent colors because different wavelengths of light are refracted by different amounts. Each color corresponds to a different wavelength.

2. **Newton's Contribution:** Sir Isaac Newton was instrumental in defining the seven colors of the spectrum in the 17th century. He associated these colors with the musical scale, possibly influenced by the idea of perfection found in the number 7, which has significance in various cultures and philosophies.

3. **Additional Scenario:** Although the visible spectrum is actually a continuum, humans perceive a range of colors, and naming them in groups like seven makes it easier to communicate and understand.

In essence, the seven colors of the rainbow are an outcome of the physics of light along with historical classifications. It's worth noting that some people actually classify rainbows differently, often just recognizing the primary colors or seeing additional shades based on personal perception.

Metadata

Model: gpt-4a-mini-2024-07-18 | Created: 2024-07-01 23:54 | Elapsed: 3.78s | Prompt tokens: 21 | Completion tokens: 254

User input #1 getUserInput()

Call LLM #2 contextAndGenerate()

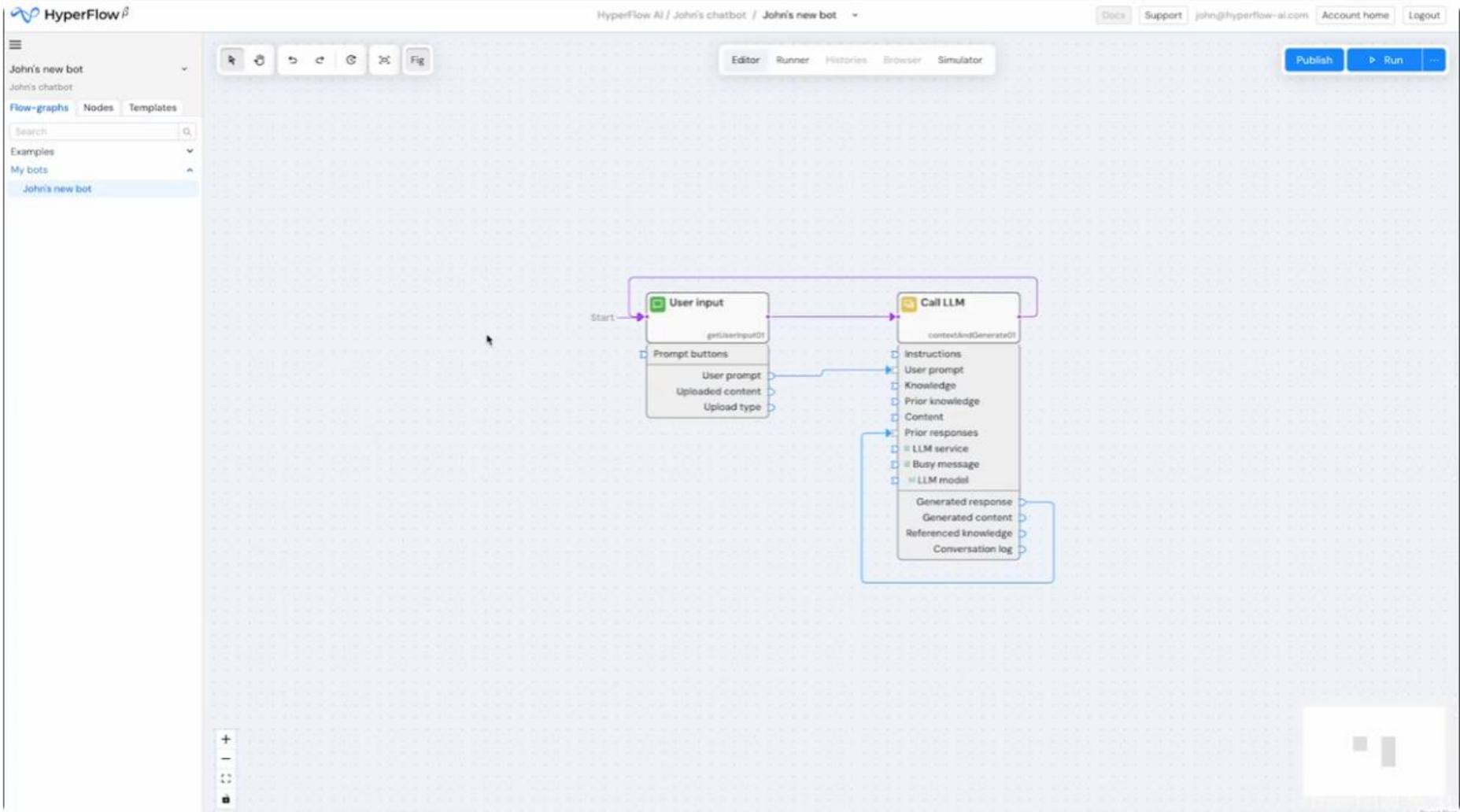
Composite prompt

Type	Content	Approx. tokens	Source step
1. user	***User Query*** Why are there 7 colors in the rainbow	9	1
2. generator	This rainbow is typically described as having seven distinct colors — red, orange, yellow, green, blue, indigo, and violet — largely due to the way light is refracted and dispersed in a prism, which can also be applied to natural rainbows formed by sunlight interacting with raindrops. 1. "Light Dispersion"	250	2
3. user	***User Query*** Why is it in the shape of an arc?	10	3

Generated text

The characteristic arc shape of a rainbow is a result of the way light is refracted, reflected, and dispersed in

플로우그래프(Flow-graphs) 구축 및 실행



Building and running flow-graphs, Part 2

Building and running flow-graphs, Part 2

Constructing an agentic AI application

Agentic AI Applications

- Multiple LLMs in different roles
- Acting as agents - “agentic”
- Two-agent, content-generation app
 - Agent One generating images
 - Agent Two refining the image prompt
 - Looping analysis & re-prompting



Agentic AI Applications

- ▶ Multiple LLMs in different roles
- ▶ Acting as agents - "agentic"
- ▶ Two-agent, content-generation app
 - Agent One generating images
 - Agent Two refining the image prompt
 - Looping analysis & re-prompting

- ▶ 다양한 역할을 하는 여러 LLM
- ▶ 에이전트 역할 - "에이전트"
- ▶ 2가지 에이전트, 콘텐츠 생성 앱
 - 에이전트 1, 생성 이미지
 - 에이전트 2, 이미지 프롬프트 정제하기
 - 루프 분석 및 반복프롬프팅

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Flow-graphs Nodes Templates

Search Examples My bots Agentic image creation Saved

Fig

+

-

✖

✖

React File

The screenshot shows the HyperFlow AI application interface. At the top, there's a navigation bar with 'HyperFlow β' on the left, 'Docs', 'Support', 'john@hyperflow-ai.com', 'Account home', and 'Logout' on the right. Below the navigation is a toolbar with icons for back, forward, search, and a 'Fig' button. A tab bar above the main area shows 'Editor', 'Runner', 'Histories', 'Browser', and 'Simulator'. On the left, a sidebar has sections for 'Flow-graphs', 'Nodes', 'Templates', 'Search', 'Examples', 'My bots', and 'Saved'. Under 'Saved', 'Agentic image creation' is selected. At the bottom left, there are zoom and orientation controls. A large, empty white area is the main workspace.

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Publish Run

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control Knowledge bases LLMs Tools

+

React File

The screenshot shows the HyperFlow AI application interface. At the top, there's a navigation bar with the title "HyperFlow β", the current project "HyperFlow AI / John's chatbot / Agentic image creation", and links for "Docs", "Support", "john@hyperflow-ai.com", "Account home", and "Logout". Below the navigation is a toolbar with icons for "Editor", "Runner", "Histories", "Browser", and "Simulator", along with "Publish" and "Run" buttons. On the left, a sidebar displays the project name "Agentic image creation" and "John's chatbot". It includes tabs for "Flow-graphs", "Nodes" (which is selected), and "Templates". A search bar is also present. A large central area is labeled "Agentic image creation" and "John's chatbot". On the far left of this area, there's a vertical toolbar with icons for "Annotation", "Chat", "Content", "Data flow", "Flow control", "Knowledge bases", "LLMs", and "Tools". At the bottom left, there are zoom controls (+, -, ×, ÷). At the bottom right, there's a "React File" button.

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control Knowledge bases LLMs Call LLM Instructions Tools

Start → Call LLM

contextAndGenerate()

Instructions User prompt Knowledge Prior knowledge Content Prior responses Prompt template LLM service

Generated response Generated content Referenced knowledge Conversation log

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control Knowledge bases LLMs Call LLM Instructions Tools

Start → Call LLM contextAndGenerate01

- Instructions
- User prompt
- Knowledge
- Prior knowledge
- Content
- Prior responses
- Prompt template
- LLM service

Generated response Generated content Referenced knowledge Conversation log

Call LLM contextAndGenerate02

- Instructions
- User prompt
- Knowledge
- Prior knowledge
- Content
- Prior responses
- Prompt template
- LLM service

Generated response Generated content Referenced knowledge Conversation log

+

React File

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control Knowledge bases LLMs

Call LLM Instructions Tools

Add node > Annotation > Annotation

Insert flow-graph snippet > Chat > Call LLM

Cut

Copy

Paste

Delete

Zoom to fit

Toggle show all ports

Toggle IDs

Annotation

Call LLM contextAndGenerate01

Instructions User prompt Knowledge Prior knowledge Content Prior responses Prompt template LLM service

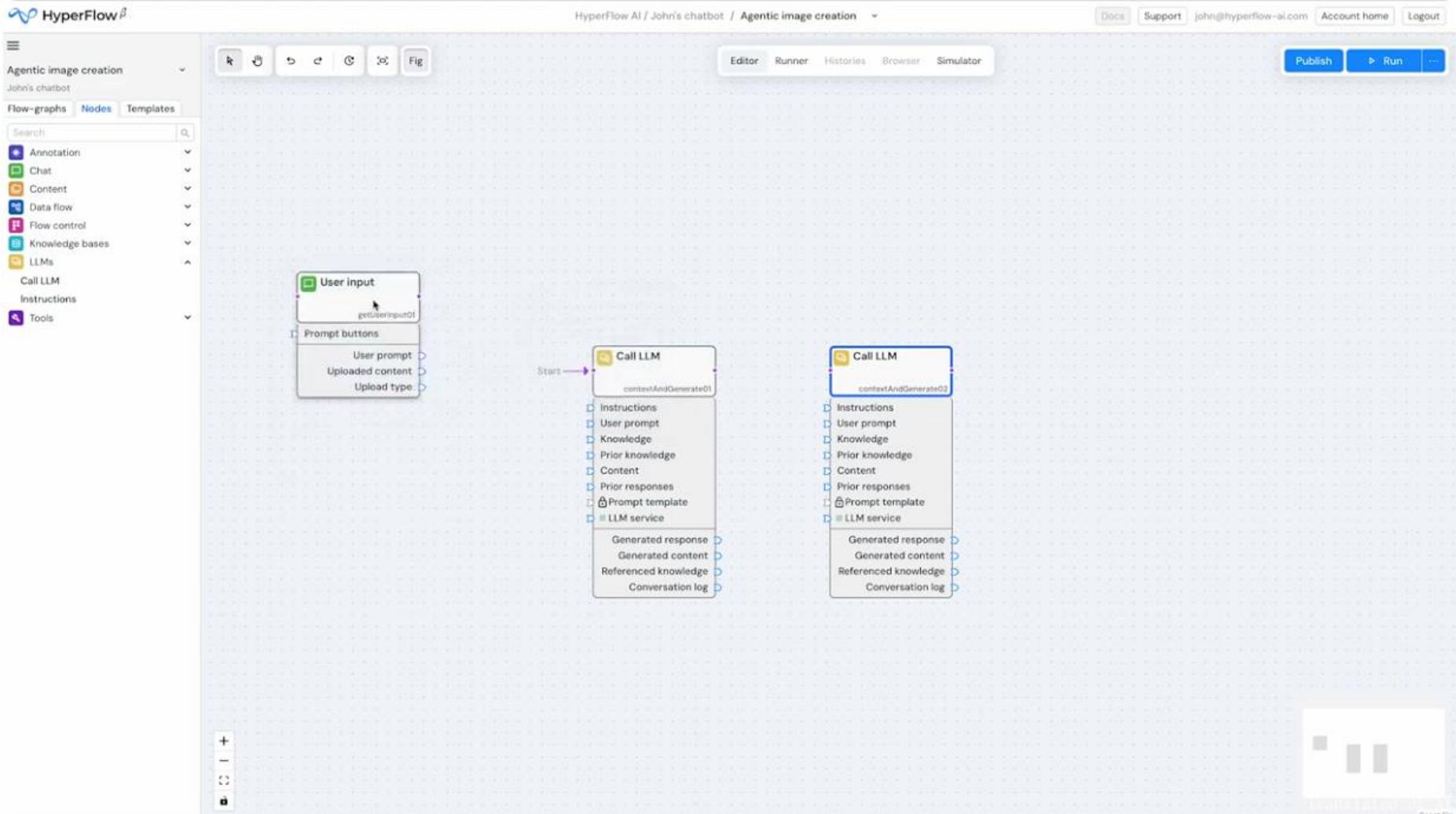
Generated response Generated content Referenced knowledge Conversation log

Call LLM contextAndGenerate02

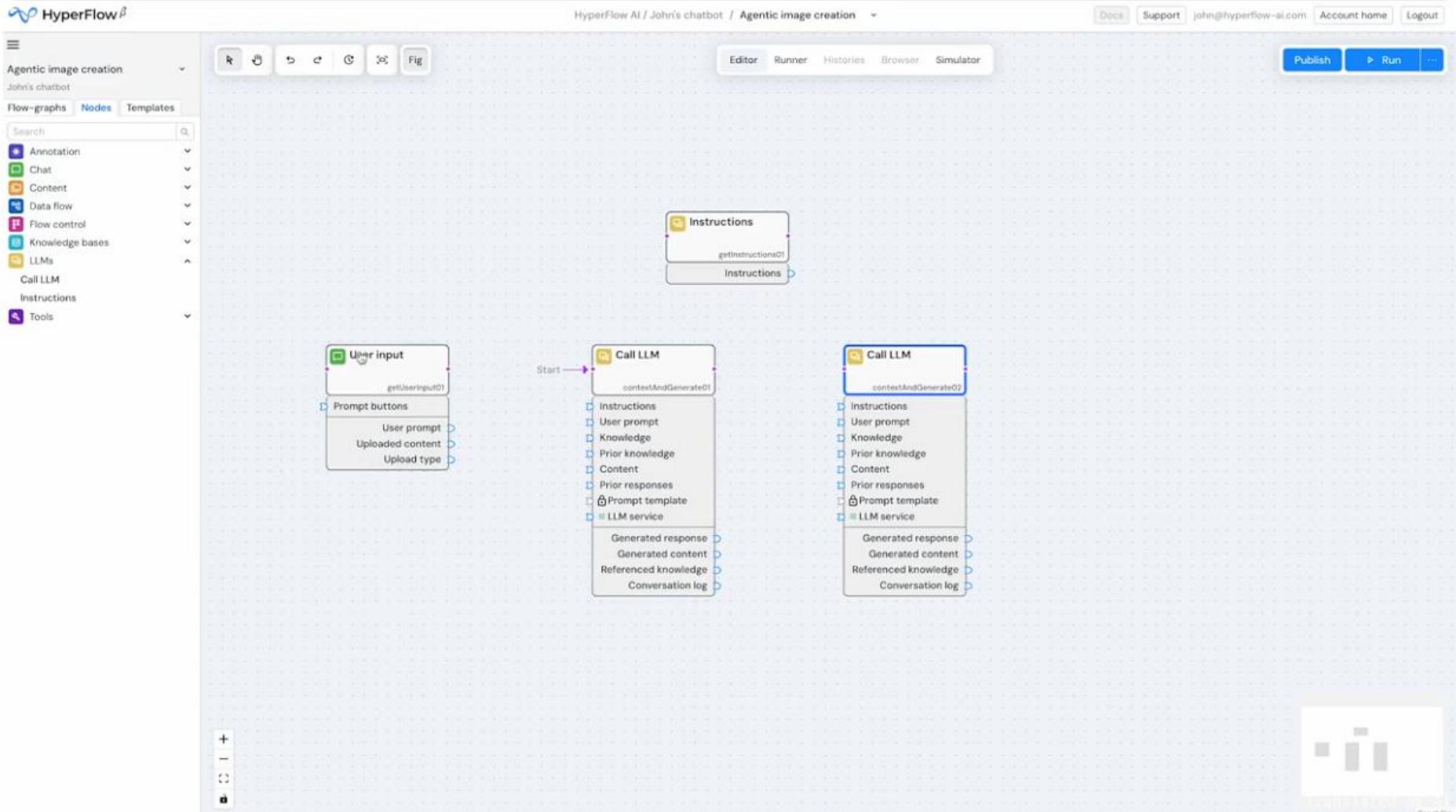
Instructions User prompt Knowledge Prior knowledge Content Prior responses Prompt template LLM service

Generated response Generated content Referenced knowledge Conversation log

에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control Knowledge bases Call LLM Instructions Tools

```
graph LR; Start((Start)) --> UserInput[User input]; UserInput --> CallLLM1[Call LLM]; CallLLM1 --> CallLLM2[Call LLM]; CallLLM2 --> End((End));
```

User input

Prompt buttons

User prompt

Uploaded content

Upload type

Instructions

getInstructions01

Instructions

contextAndGenerate01

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Prompt template

LLM service

Generated response

Generated content

Referenced knowledge

Conversation log

Generated response

Generated content

Referenced knowledge

Conversation log

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow Ai / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run ...

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

- Annotation
- Chat
- Content
- Data flow
- Flow control
- Knowledge bases
- LLMs
- Call LLM
- Instructions
- Tools

Start → User input → Call LLM → Call LLM

User input node details:

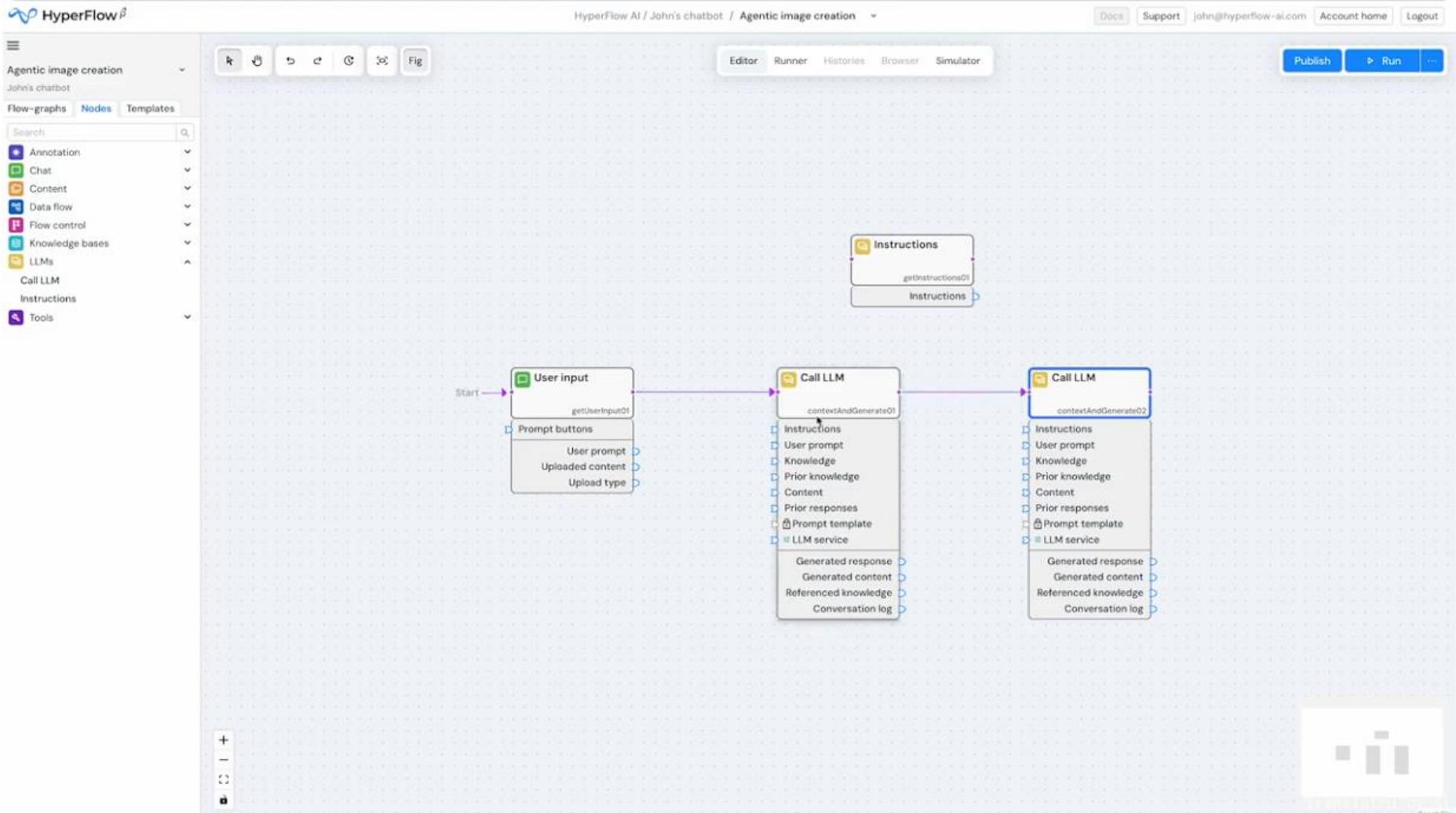
- getInput01
- Prompt buttons
 - User prompt
 - Uploaded content
 - Upload type

Call LLM node details:

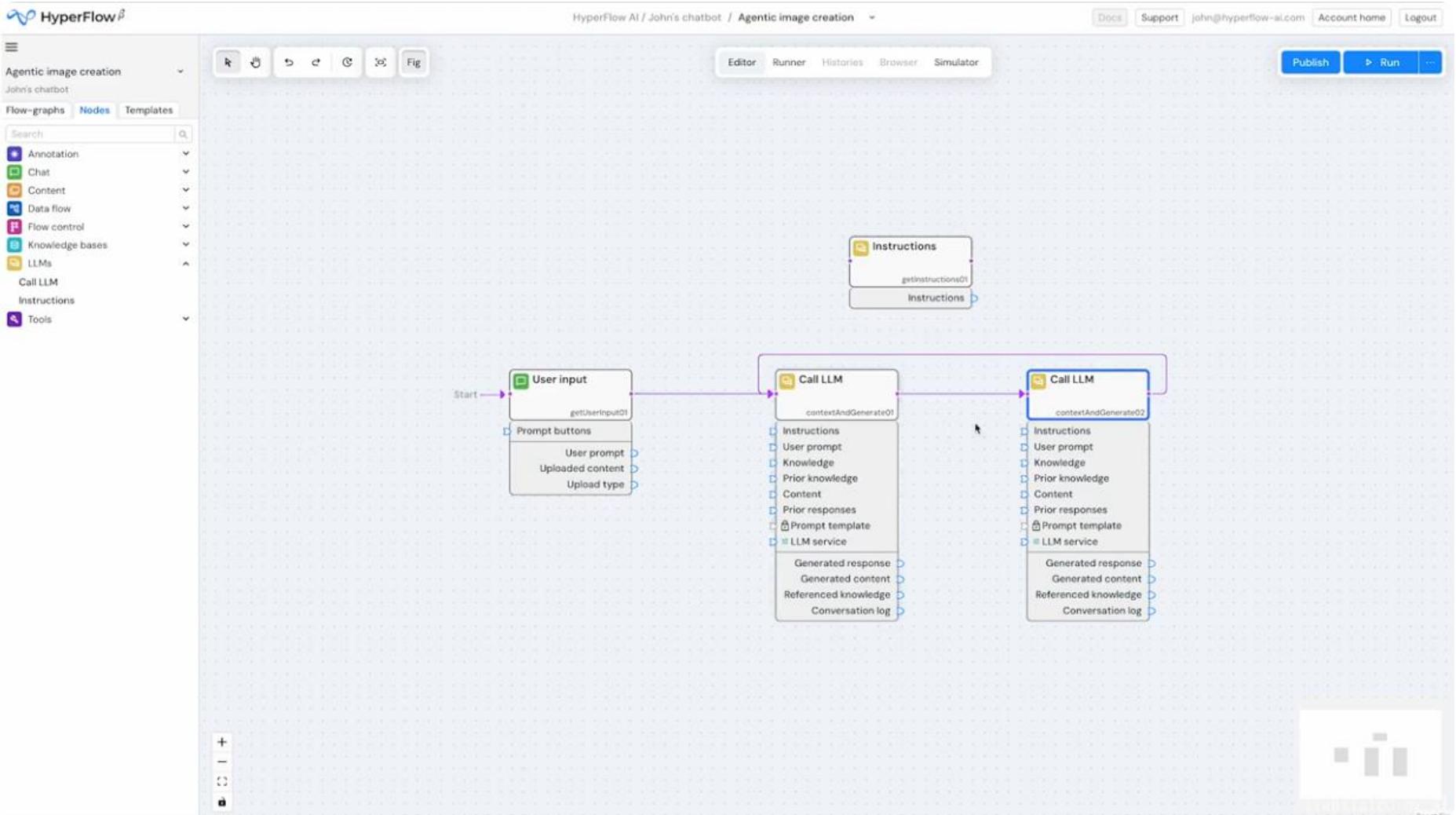
- contextAndGenerate01
 - Instructions
 - User prompt
 - Knowledge
 - Prior knowledge
 - Content
 - Prior responses
 - Prompt template
 - LLM service
- Generated response
- Generated content
- Referenced knowledge
- Conversation log

- contextAndGenerate02
 - Instructions
 - User prompt
 - Knowledge
 - Prior knowledge
 - Content
 - Prior responses
 - Prompt template
 - LLM service
- Generated response
- Generated content
- Referenced knowledge
- Conversation log

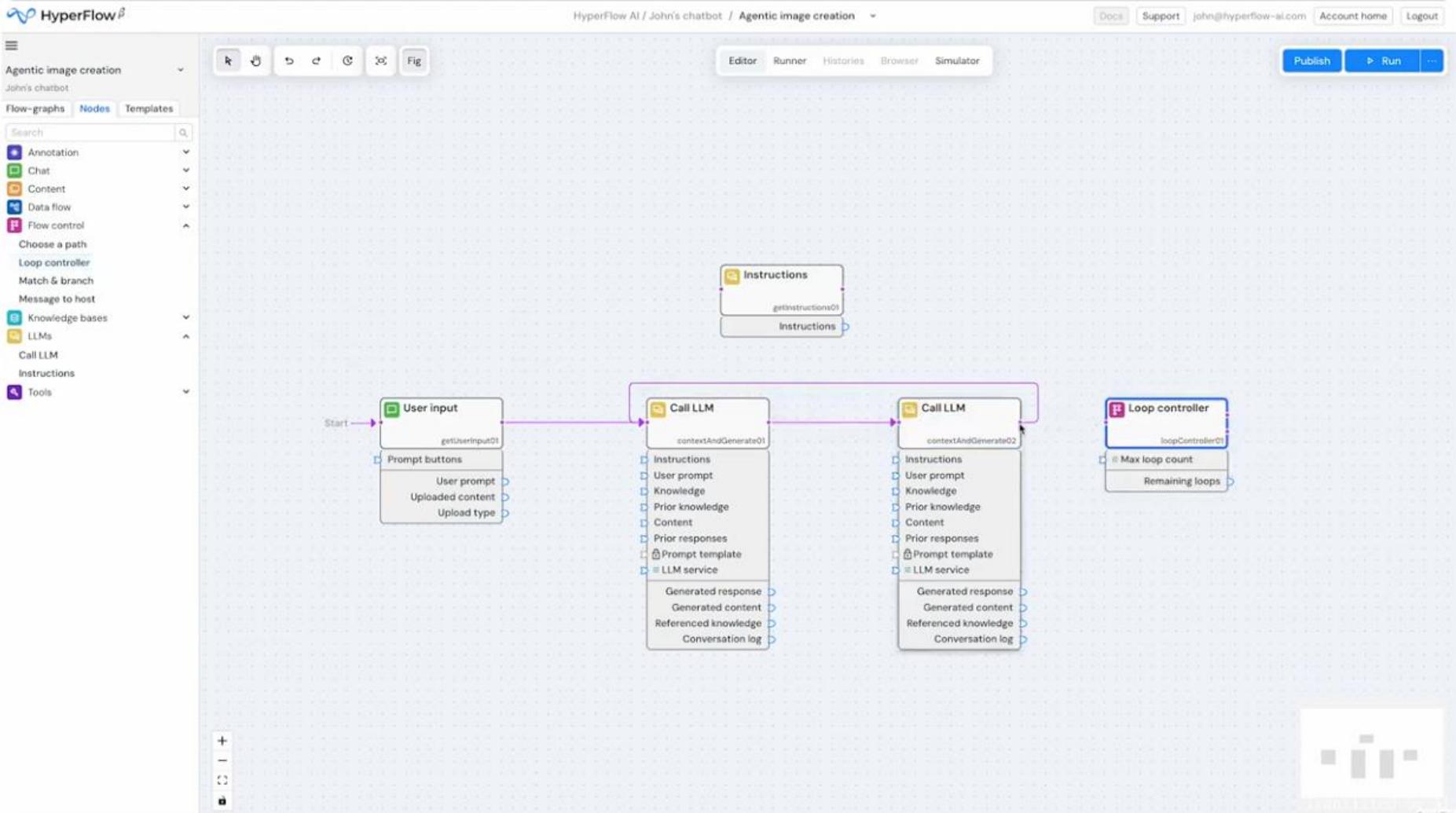
에이전트 AI 애플리케이션 구축



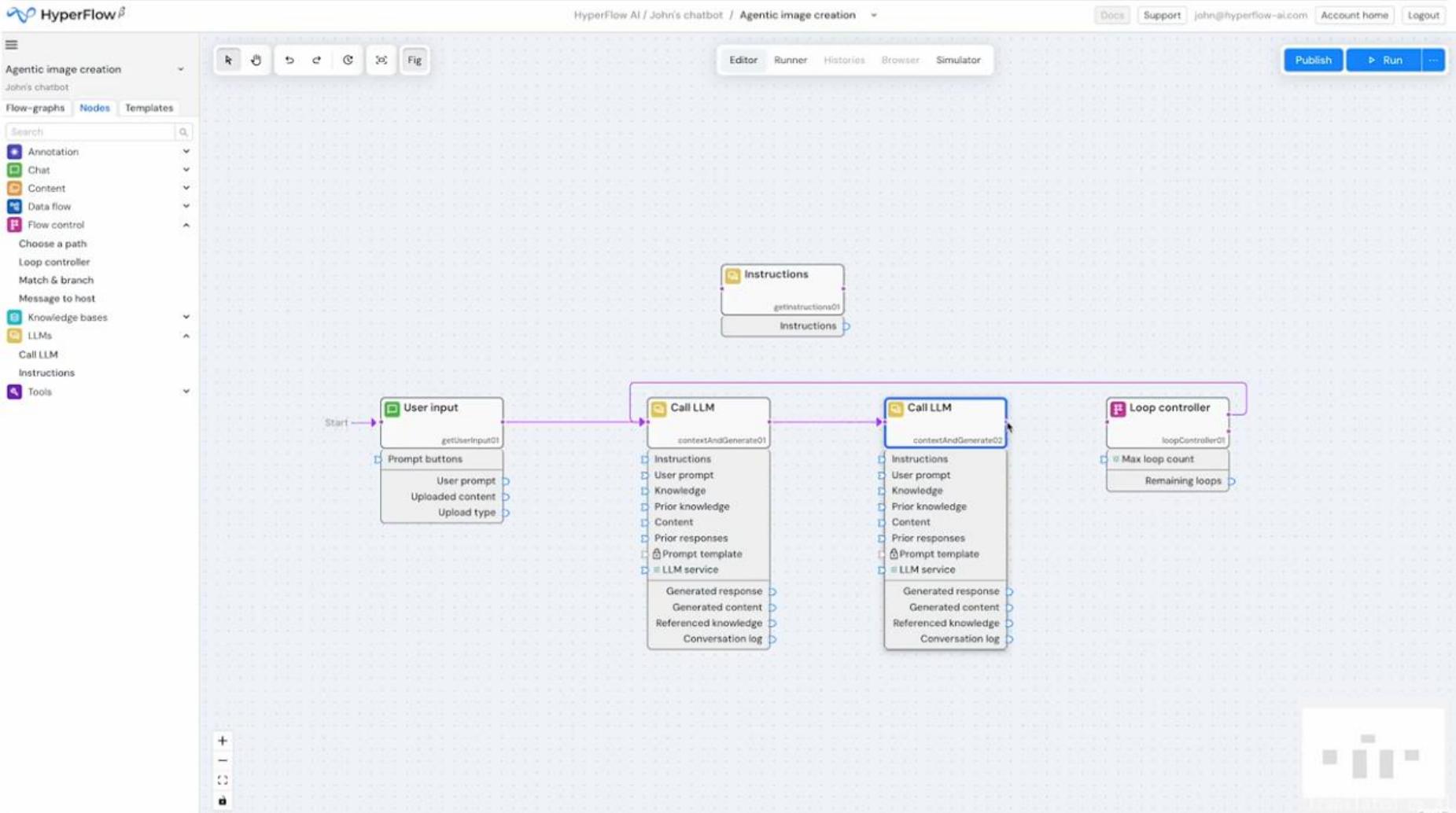
에이전트 AI 애플리케이션 구축



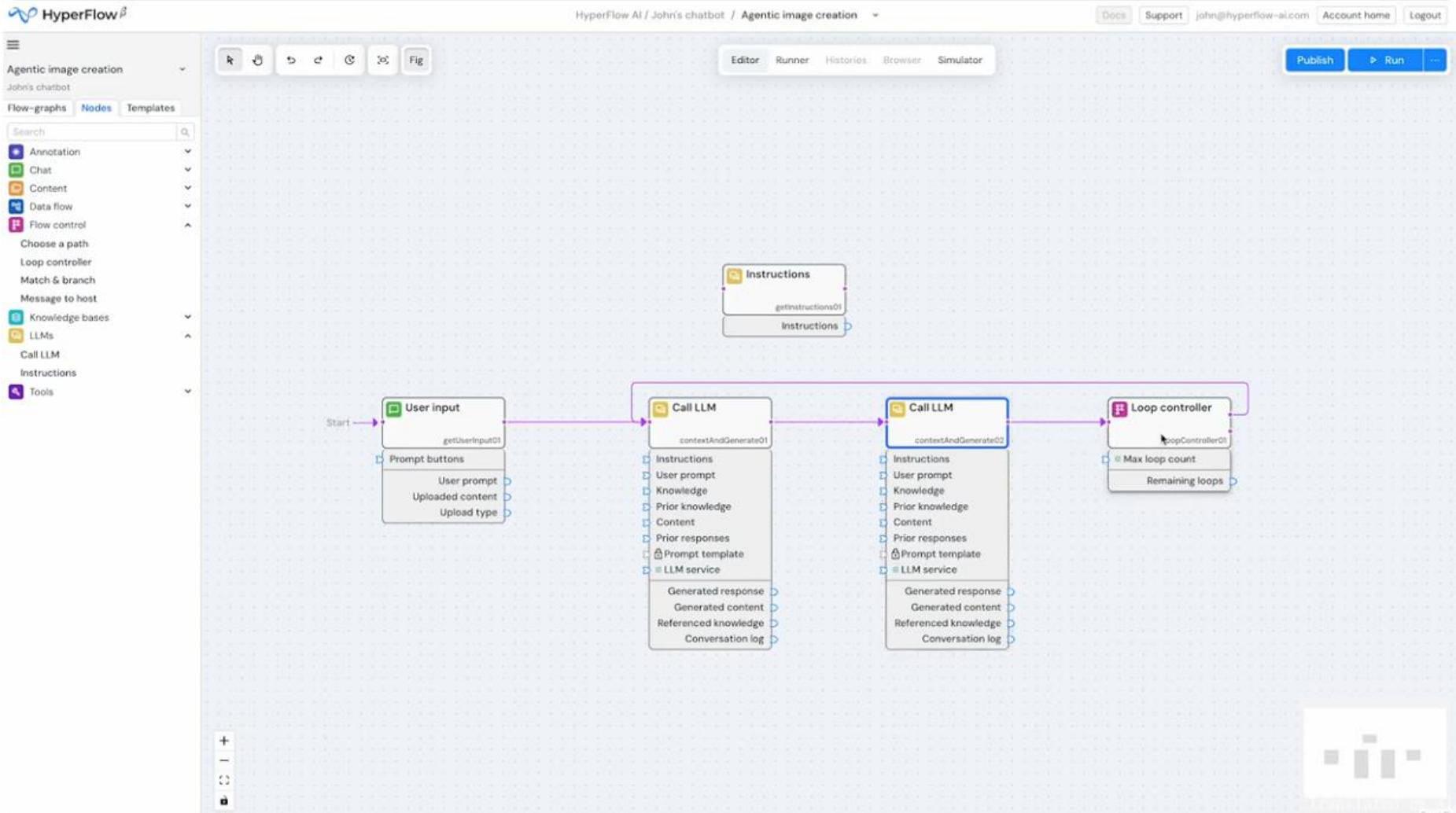
에이전트 AI 애플리케이션 구축



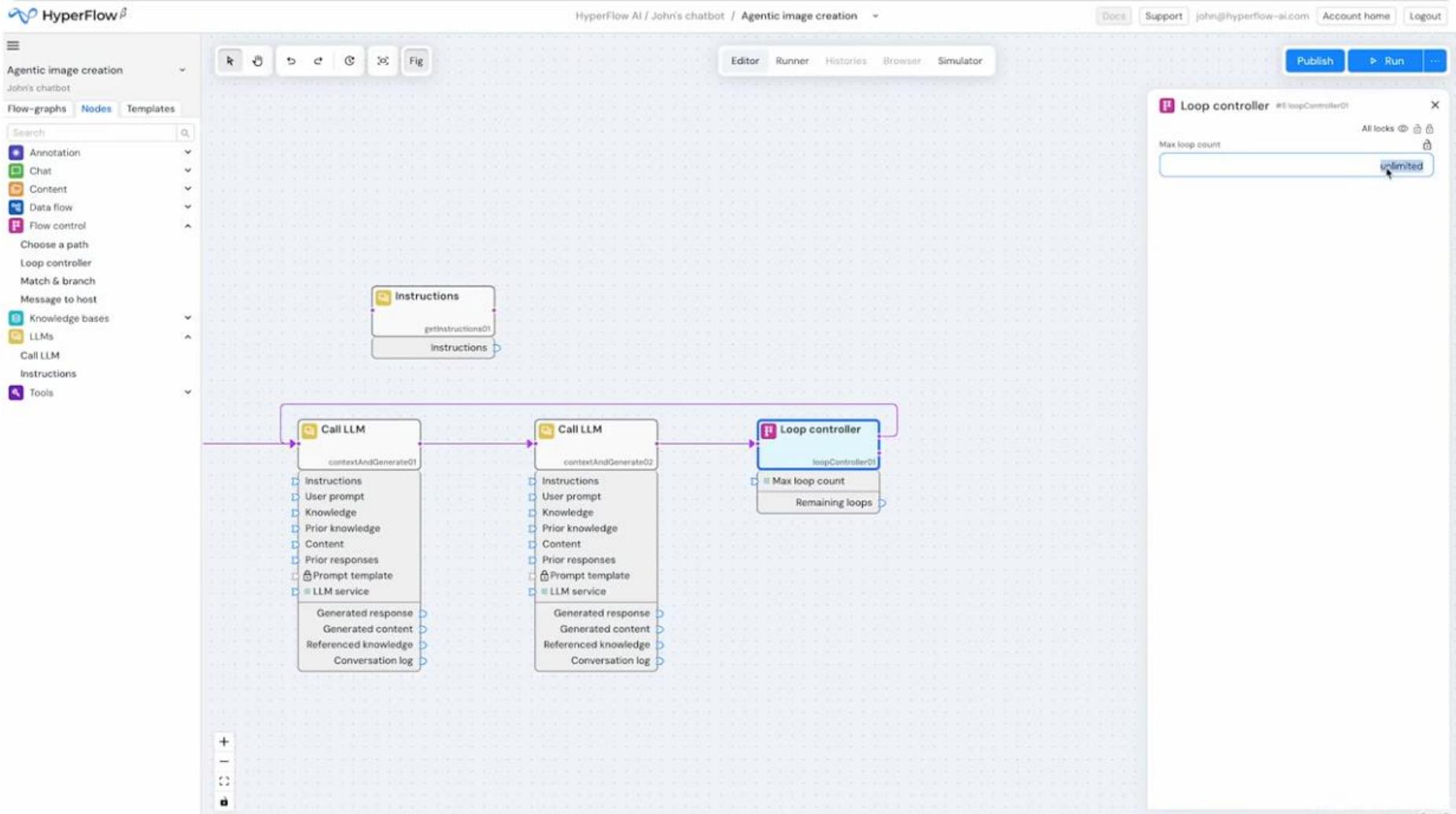
에이전트 AI 애플리케이션 구축



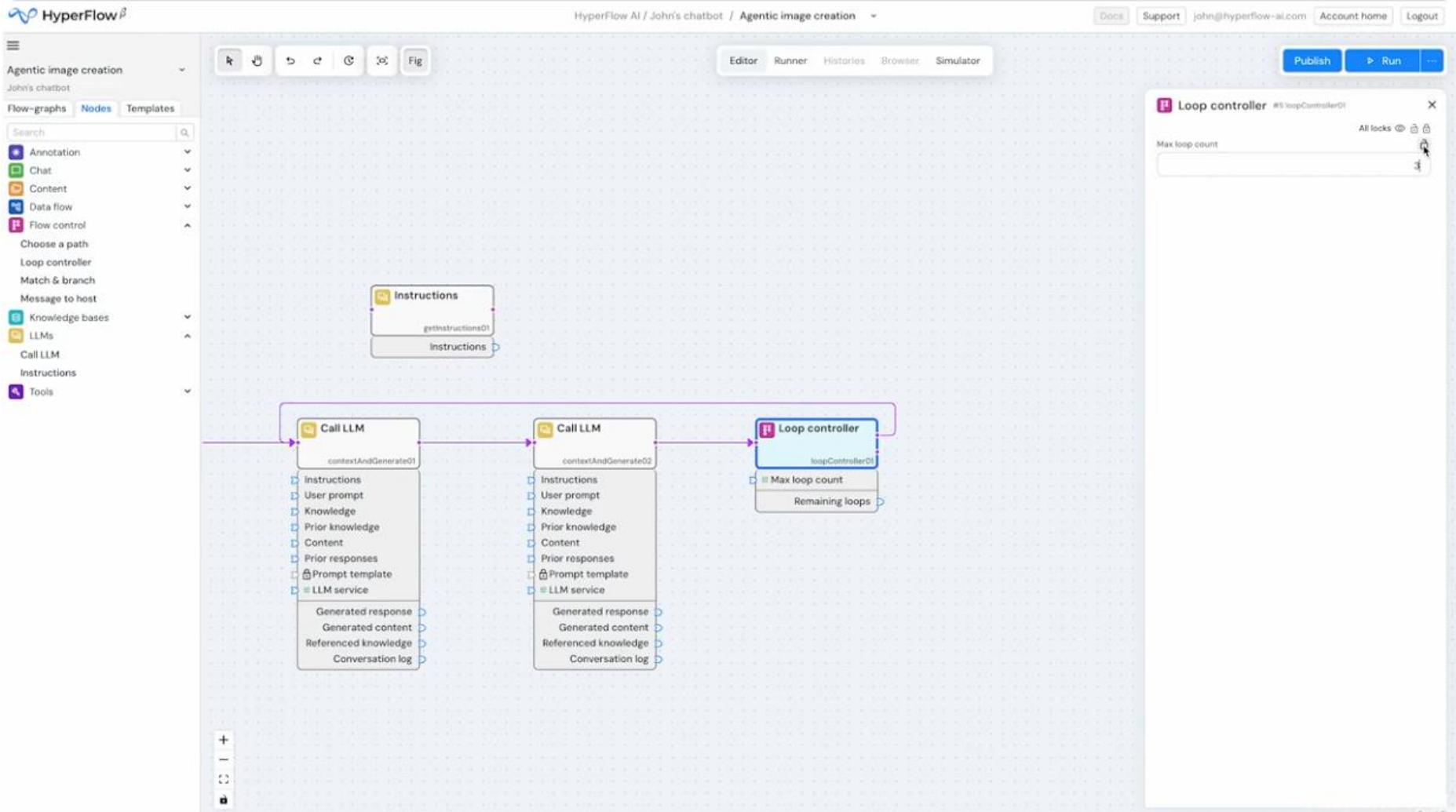
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation

John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control

Choose a path

Loop controller Match & branch Message to host

Knowledge bases LLMs

Call LLM Instructions Tools

User input

Start → get userInput01

Prompt buttons

User prompt
Uploaded content
Upload type

Call LLM contextAndGenerate01

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
LLM service
Busy message

Generated response
Generated content
Referenced knowledge
Conversation log

Call LLM contextAndGenerate02

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
Prompt template
LLM service

Generated response
Generated content
Referenced knowledge
Conversation log

Loop controller loopController0

Max loop count
Remaining loops

Call LLM #1 contextAndGenerate01

All locks: 0 0 0

Role: chat

Generation options

Prompt template

LLM service: Select service...

```
graph LR; Start((Start)) --> UI[User input: get userInput01]; UI --> C1[Call LLM: contextAndGenerate01]; C1 --> C2[Call LLM: contextAndGenerate02]; C2 --> LC[Loop controller: loopController0]; LC --> End((End));
```

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation

John's chatbot

Flow-graphs Nodes Templates

Search

- Annotation
- Chat
- Content
- Data flow
- Flow control
- Choose a path
- Loop controller
- Match & branch
- Message to host
- Knowledge bases
- LLMs
- Call LLM
- Instructions
- Tools

User input

Start

getUserId01

Prompt buttons

User prompt

Uploaded content

Upload type

Call LLM

contextAndGenerate01

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

Call LLM

contextAndGenerate02

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Prompt template

LLM service

Generated response

Generated content

Referenced knowledge

Conversation log

Loop controller

loopController0

Max loop count

Remaining loops

Call LLM #1 contextAndGenerate01

All locks

Role Chat

G Chat

Image generation

Pr Video generation

Audio generation

Clear selection

Select service...

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Flow control

Choose a path

Loop controller Match & branch Message to host

Knowledge bases LLMs Call LLM Instructions

Tools

User input

Start → Call LLM (contextAndGenerate01)

Call LLM (contextAndGenerate01) → Call LLM (contextAndGenerate02)

Call LLM (contextAndGenerate02) → Loop controller

Loop controller → End

Call LLM (contextAndGenerate01) inputs:

- Instructions
- User prompt
- Knowledge
- Prior knowledge
- Content
- Prior responses
- LLM service
- Busy message

Call LLM (contextAndGenerate01) outputs:

- Generated response
- Generated content
- Referenced knowledge
- Conversation log

Call LLM (contextAndGenerate02) inputs:

- Instructions
- User prompt
- Knowledge
- Prior knowledge
- Content
- Prior responses
- Prompt template
- LLM service

Call LLM (contextAndGenerate02) outputs:

- Generated response
- Generated content
- Referenced knowledge
- Conversation log

Loop controller inputs:

- Max loop count
- Remaining loops

Call LLM #1 contextAndGenerate01

Role: Image generation

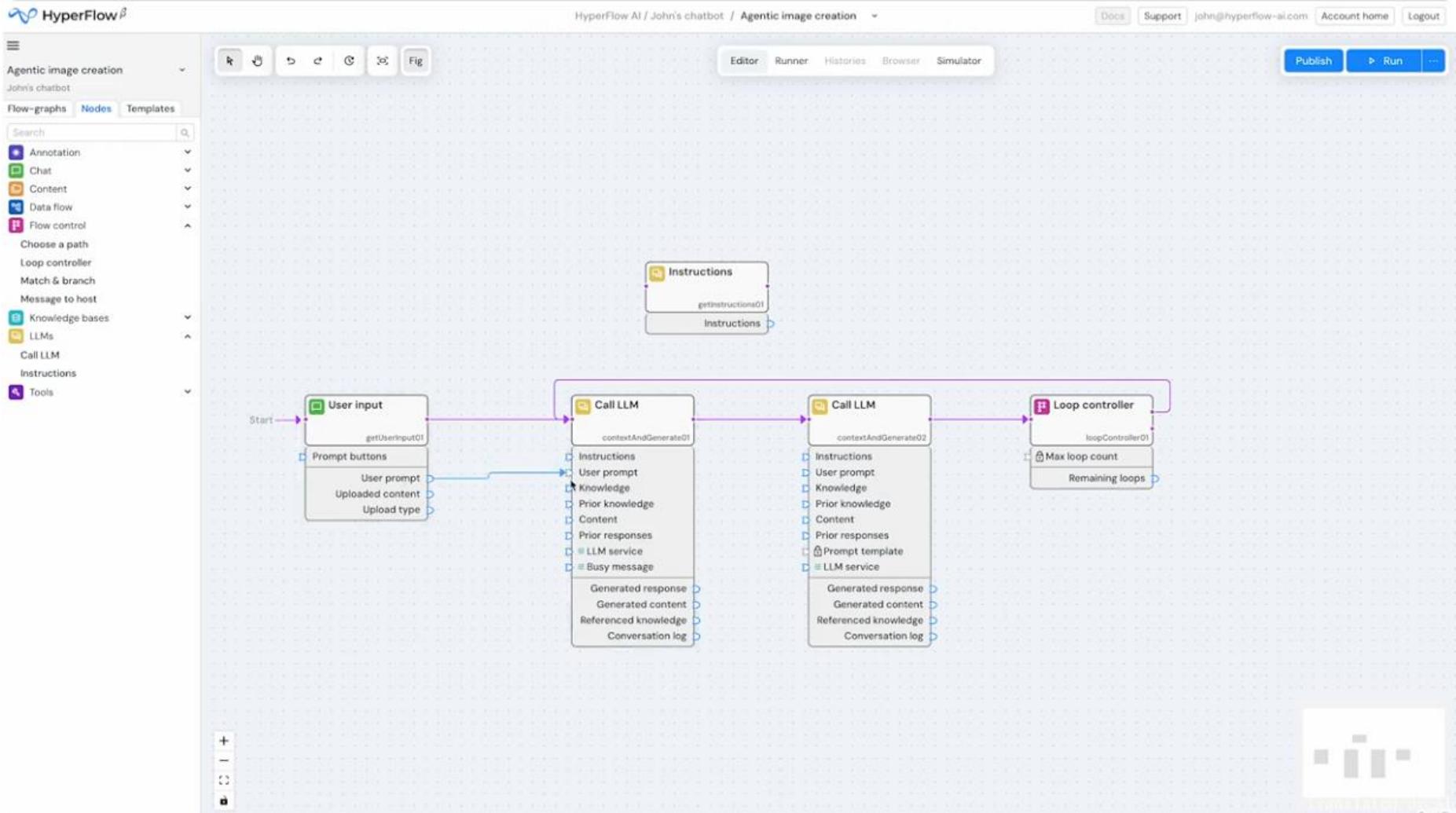
Generation options

Prompt template

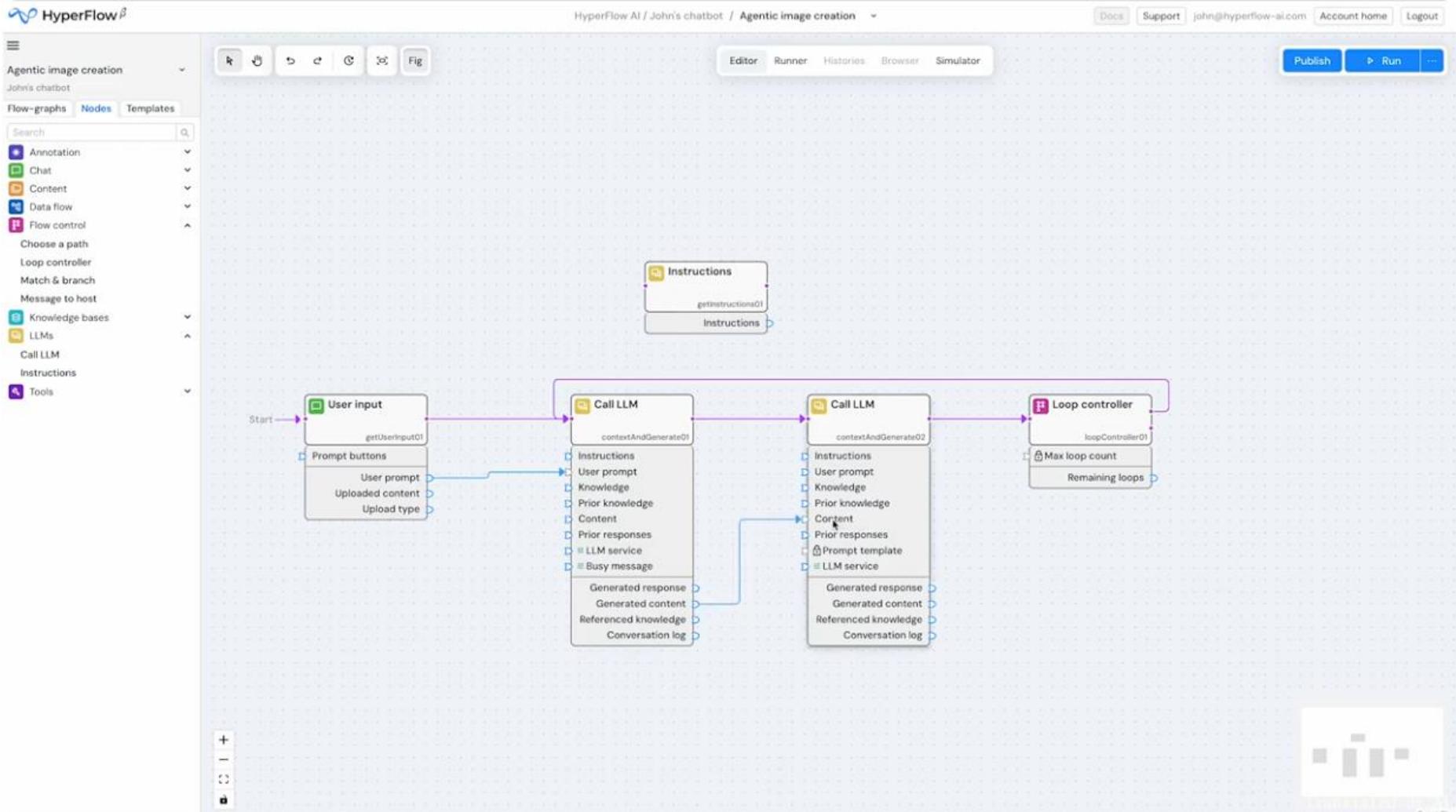
Image LLM: Select service...

OpenAI StabilityAI RecruitAI ComfyUI

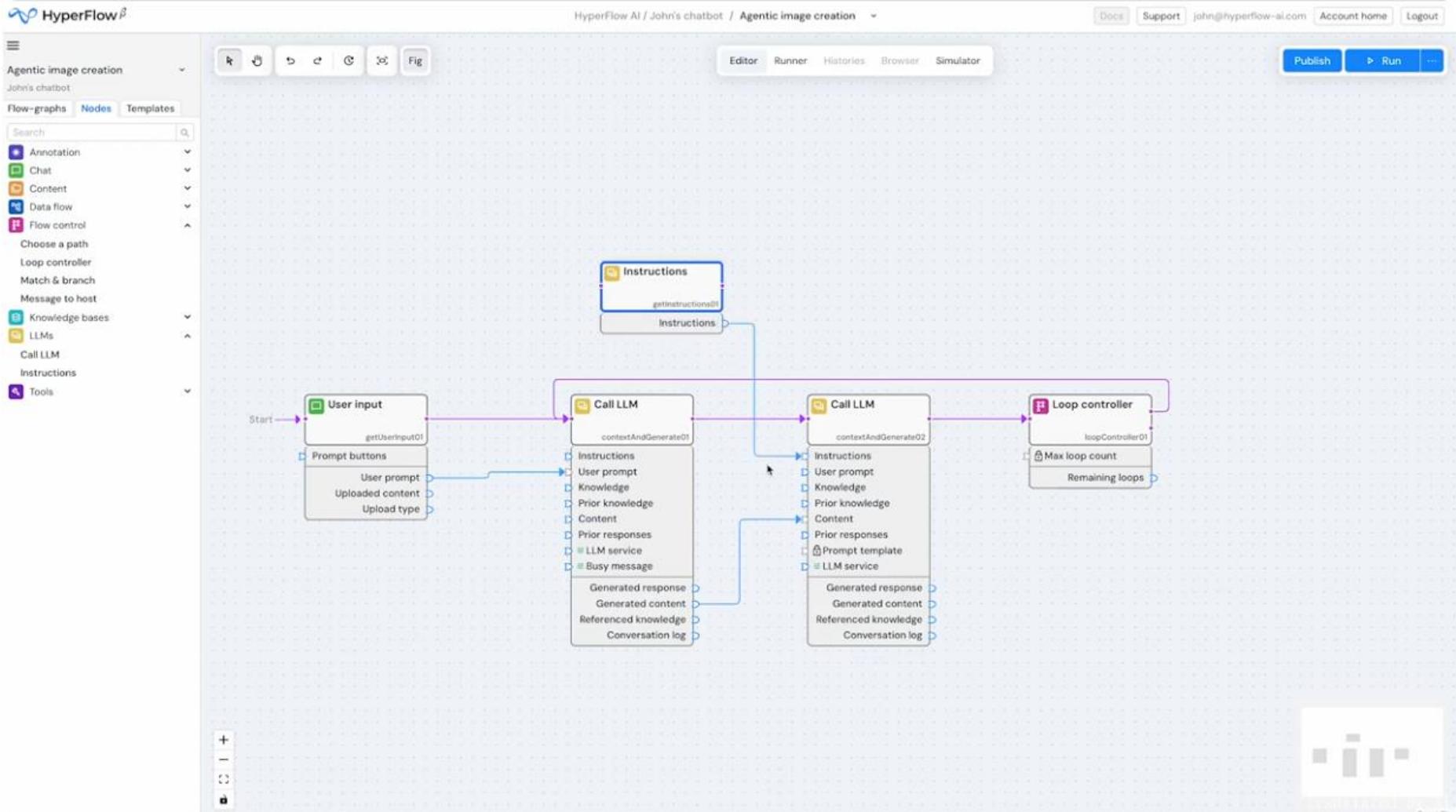
에이전트 AI 애플리케이션 구축



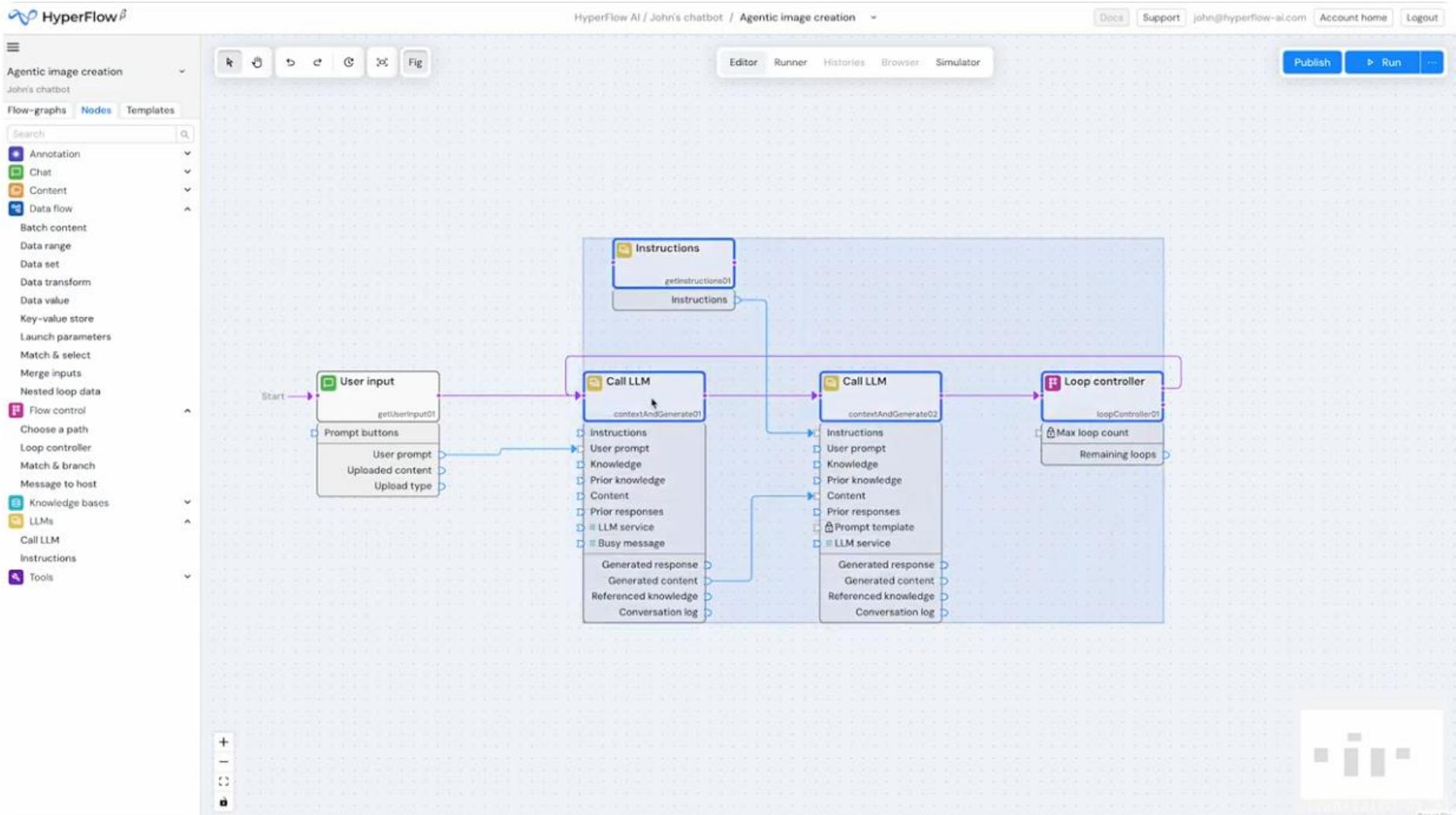
에이전트 AI 애플리케이션 구축



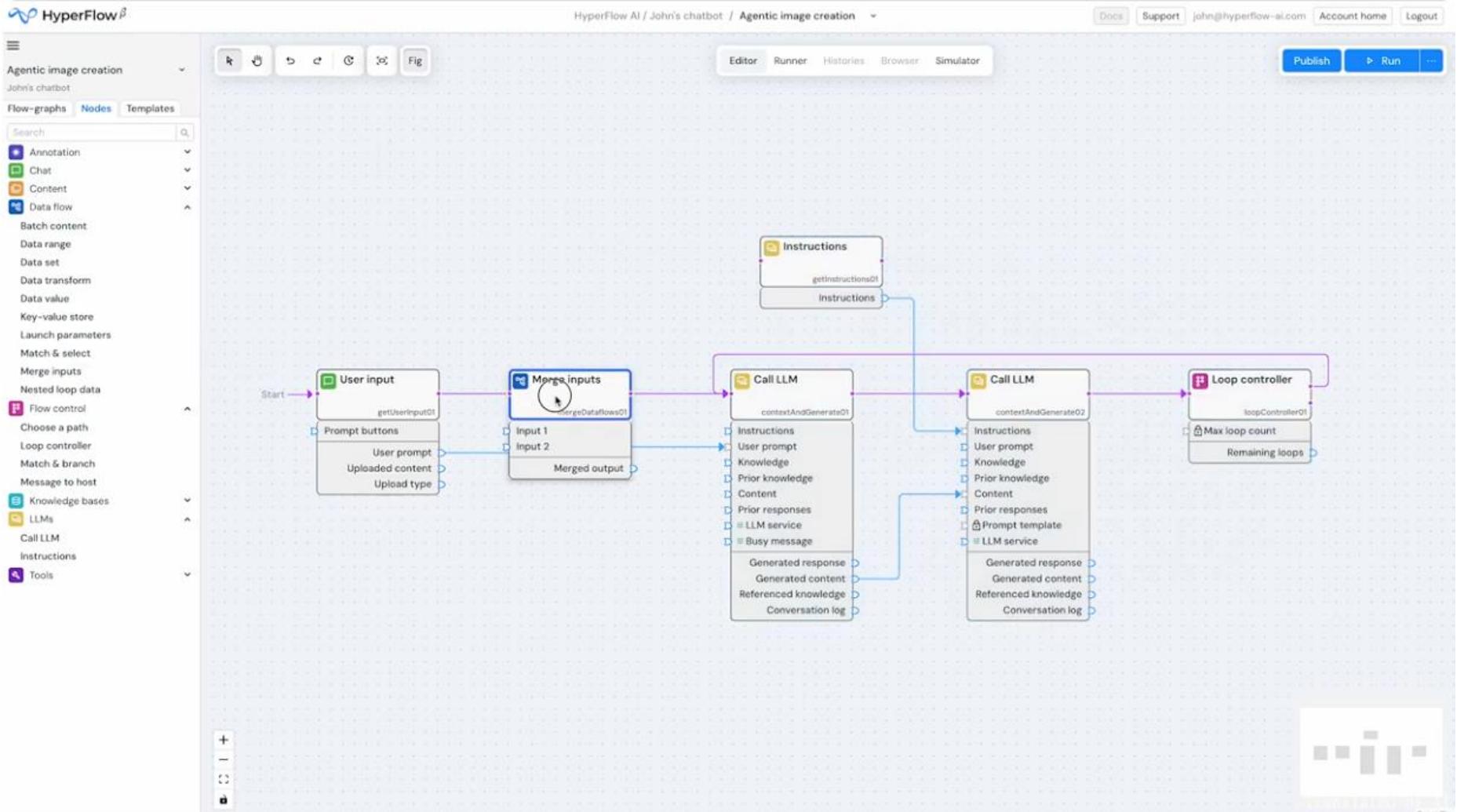
에이전트 AI 애플리케이션 구축



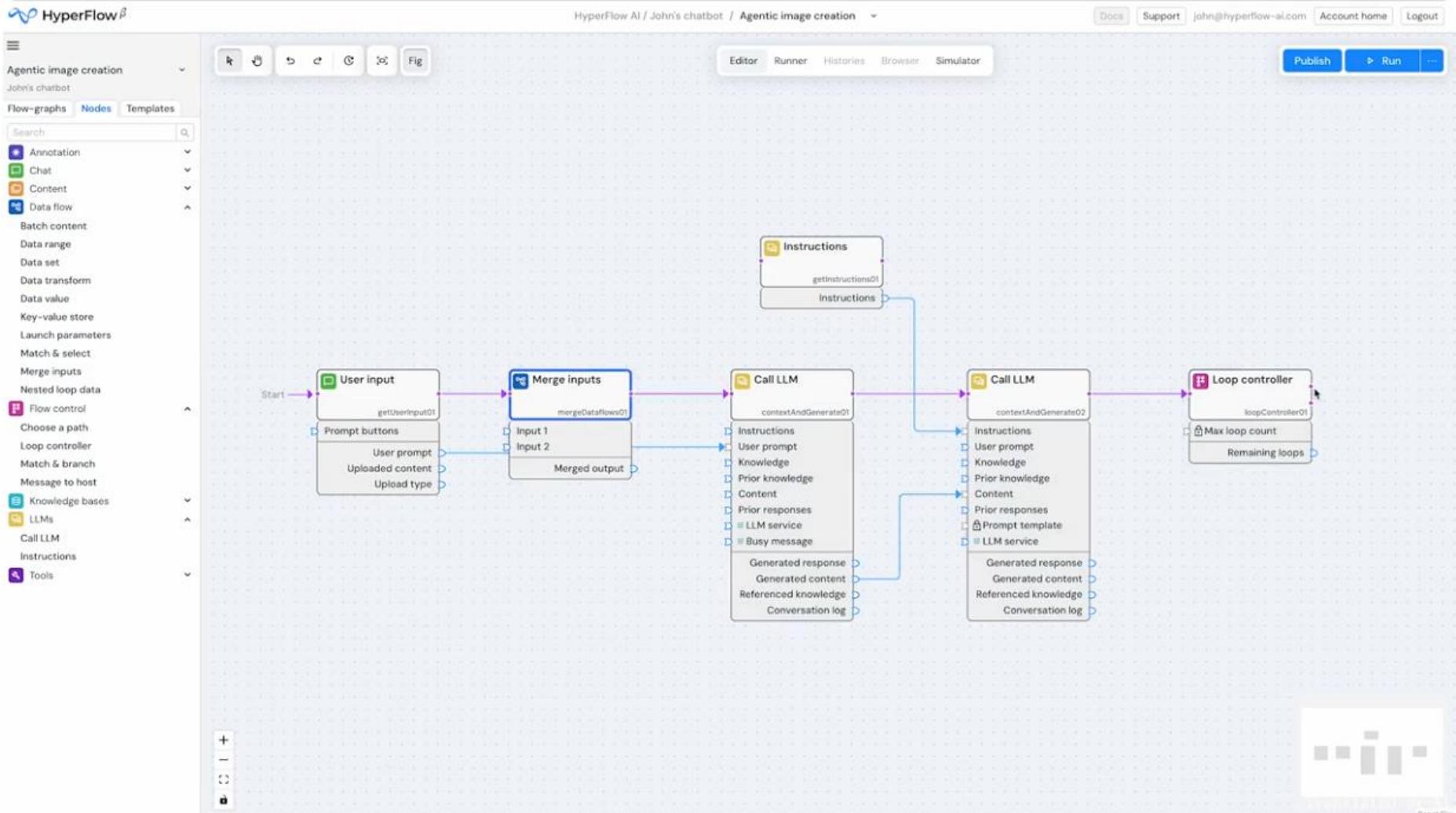
에이전트 AI 애플리케이션 구축



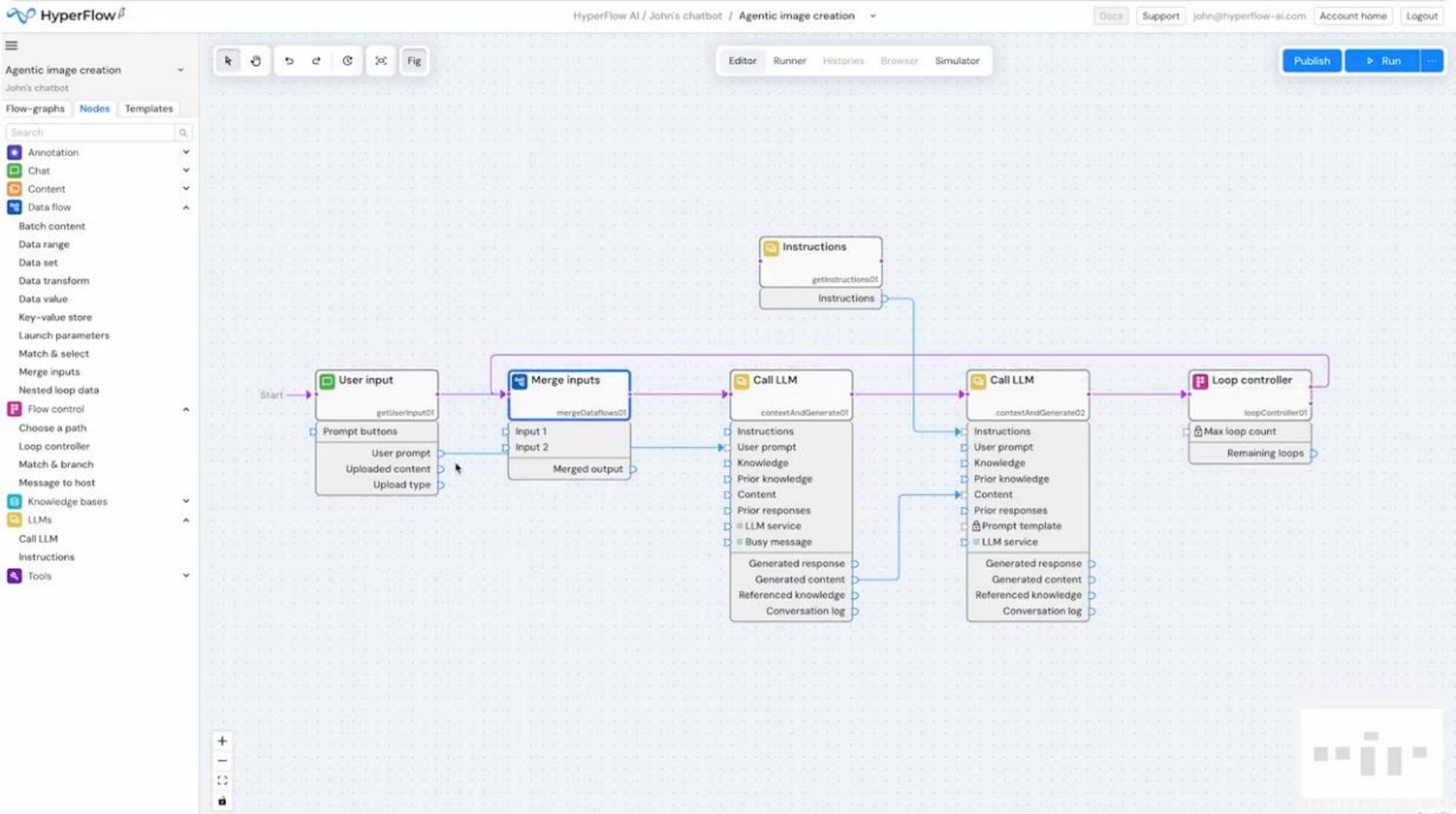
에이전트 AI 애플리케이션 구축



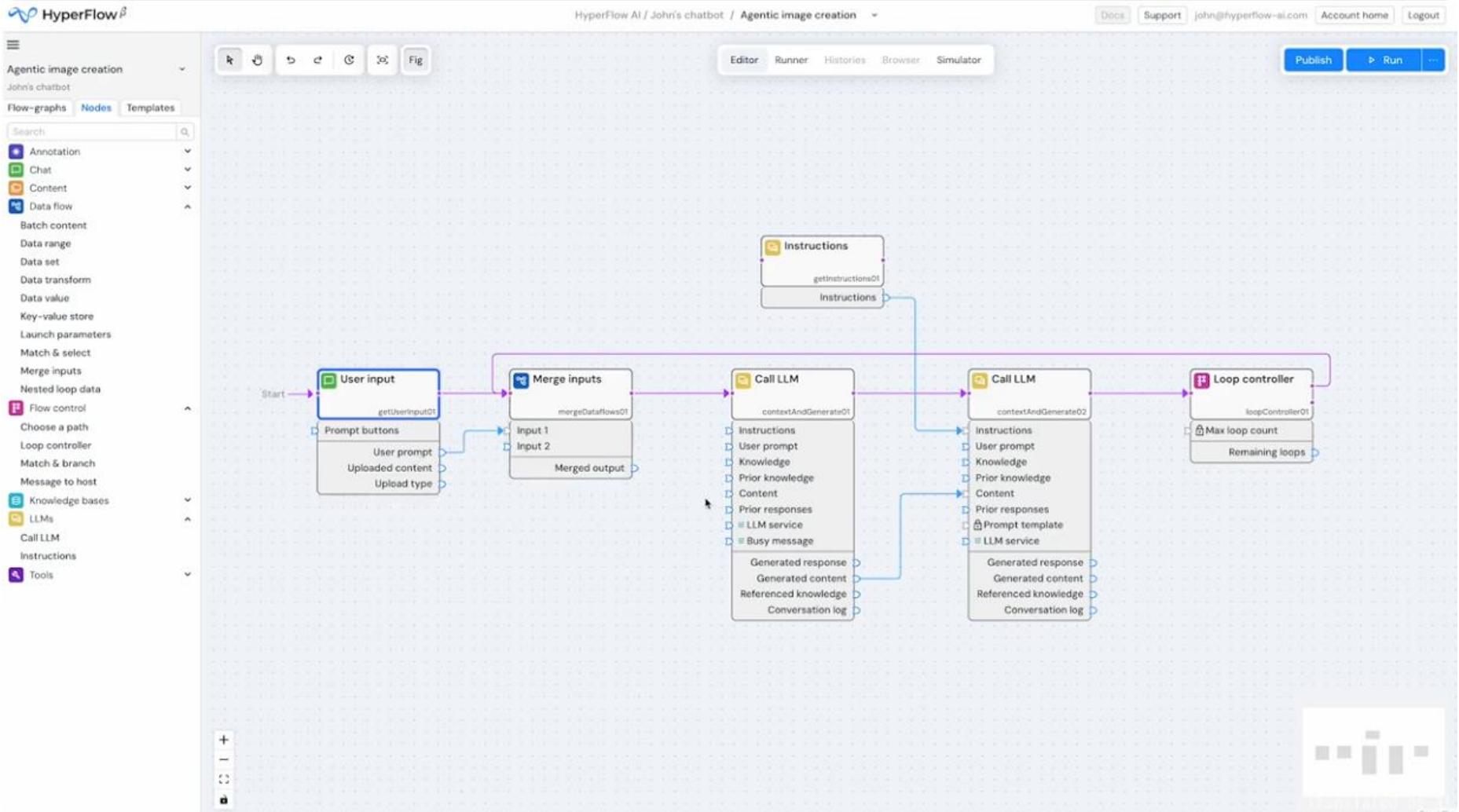
에이전트 AI 애플리케이션 구축



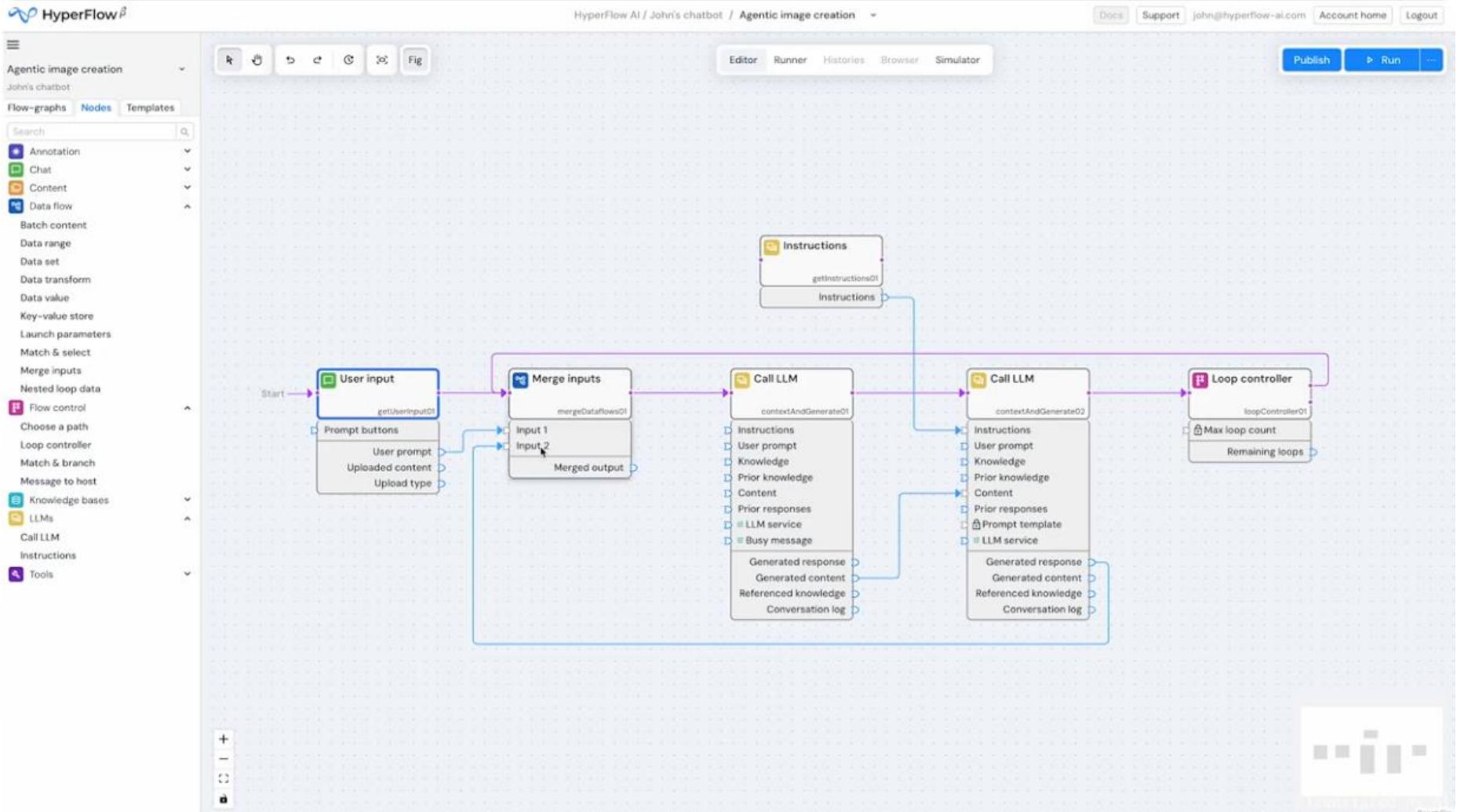
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Choose a path

Loop controller

Match & branch

Message to host

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserIdentity01

Prompt buttons

User prompt
Uploaded content
Upload type

Merge inputs

mergeDataflows01

Input 1
Input 2
Merge mode
Merged output

Instructions

getInstructions01

Instructions

Call LLM

contextAndGenerate01

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
LLM service
Busy message

Generated response
Generated content
Referenced knowledge
Conversation log

Call LLM

contextAndGenerate02

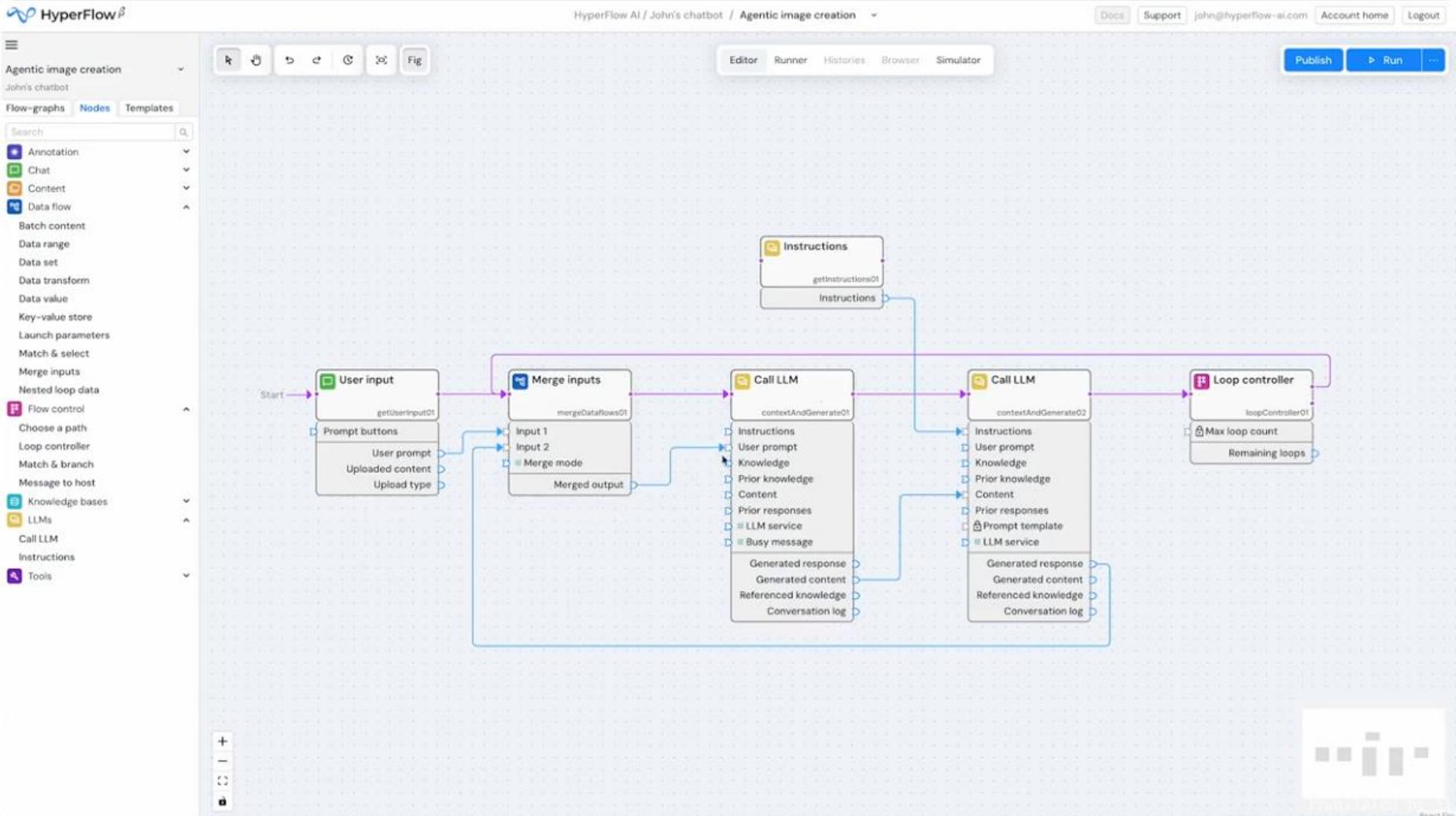
Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
Prompt template
LLM service

Generated response
Generated content
Referenced knowledge
Conversation log

Read File

```
graph LR; Start((Start)) --> UI[User input  
getUserIdentity01]; UI --> Merge1[Merge inputs  
mergeDataflows01]; UI --> Merge2[Merge inputs  
#6 mergeDataflows01]; Merge1 --> Call1[Call LLM  
contextAndGenerate01]; Merge1 --> Call2[Call LLM  
contextAndGenerate02]; Call1 --> End1[Generated response  
Generated content  
Referenced knowledge  
Conversation log]; Call2 --> End2[Generated response  
Generated content  
Referenced knowledge  
Conversation log]; Merge2 --> End3[Generated response  
Generated content  
Referenced knowledge  
Conversation log];
```

에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation

John's chatbot

Flow-graphs Nodes Templates

Search

Annotation

Chat

Content

Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Choose a path

Loop controller

Match & branch

Message to host

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserInput01

Prompt buttons

User prompt
Uploaded content
Upload type

Merge inputs

mergeDataflows01

Input 1
Input 2
Merge mode
Merged output

Call LLM

contextAndGenerate01

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
LLM service
Busy message

Generated response
Generated content
Referenced knowledge
Conversation log

Call LLM

contextAndGenerate02

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses
Prompt template
LLM service

Generated response
Generated content
Referenced knowledge
Conversation log

Instructions #4 getInstructions01

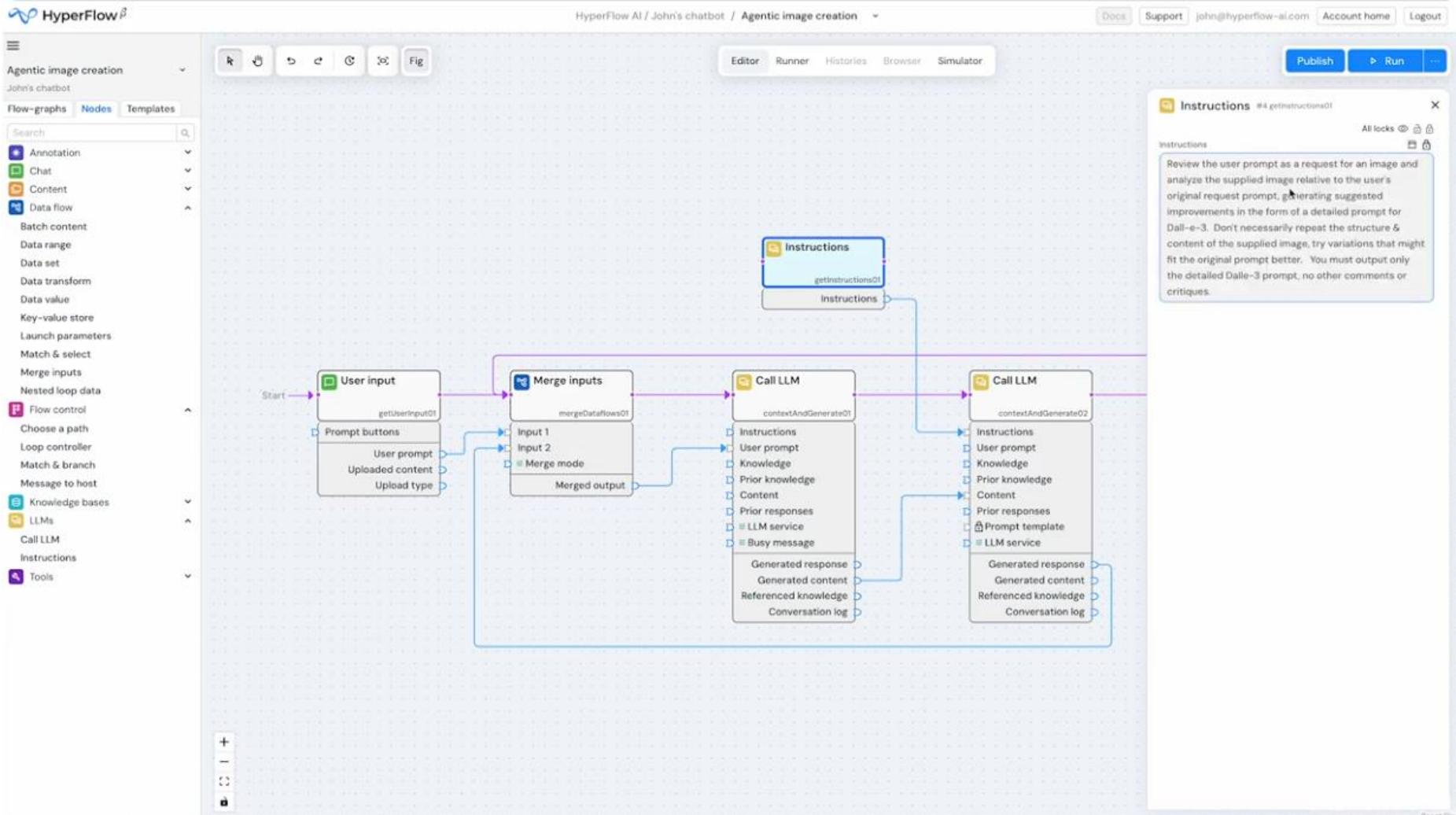
All locks

Instructions

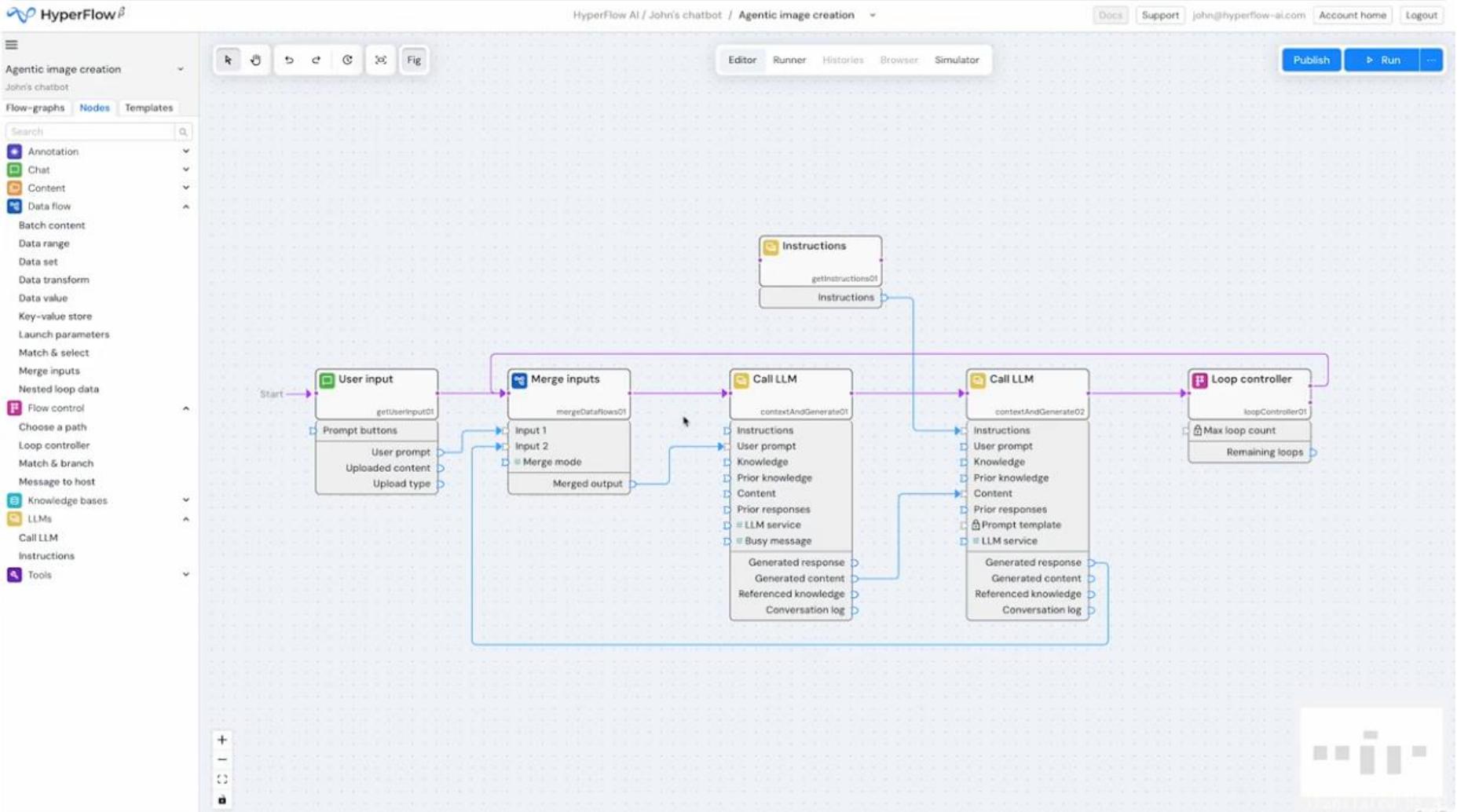
Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dalle-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

```
graph LR; Start((Start)) --> UI[User input]; UI -- getUserInput01 --> Merge[Merge inputs]; Merge -- mergeDataflows01 --> Call1[Call LLM]; Call1 -- contextAndGenerate01 --> Call2[Call LLM]; Call2 -- contextAndGenerate02 --> End((End));
```

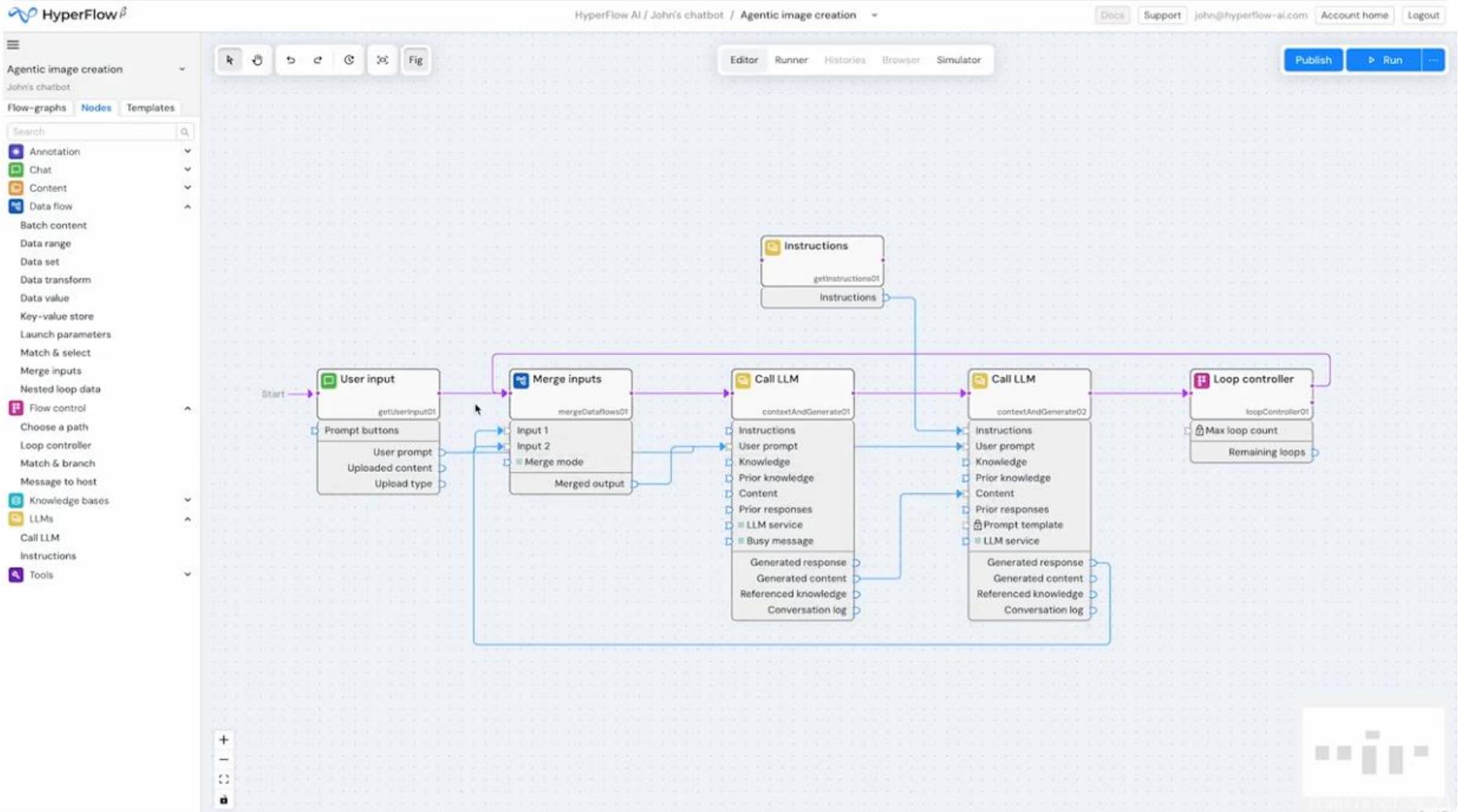
에이전트 AI 애플리케이션 구축



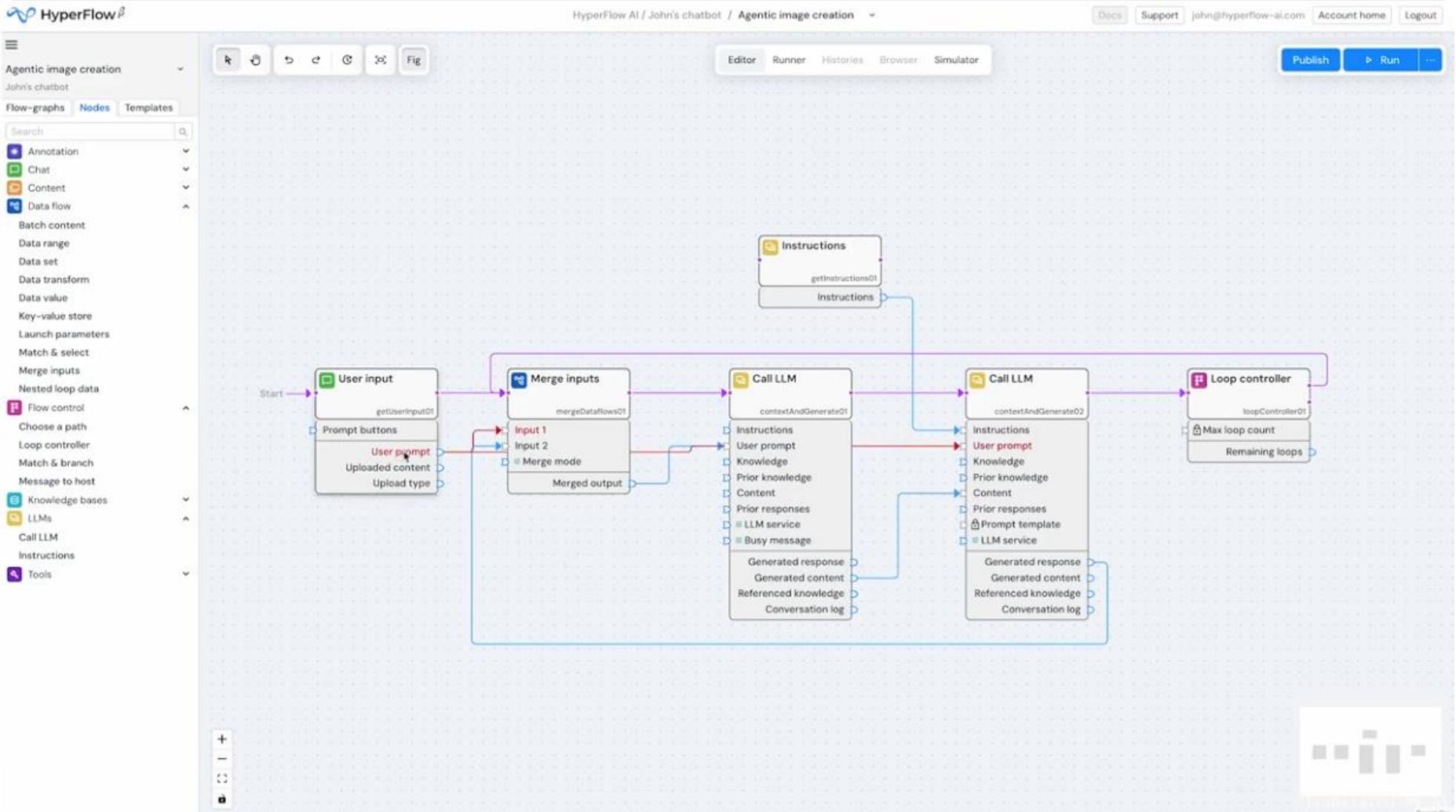
에이전트 AI 애플리케이션 구축



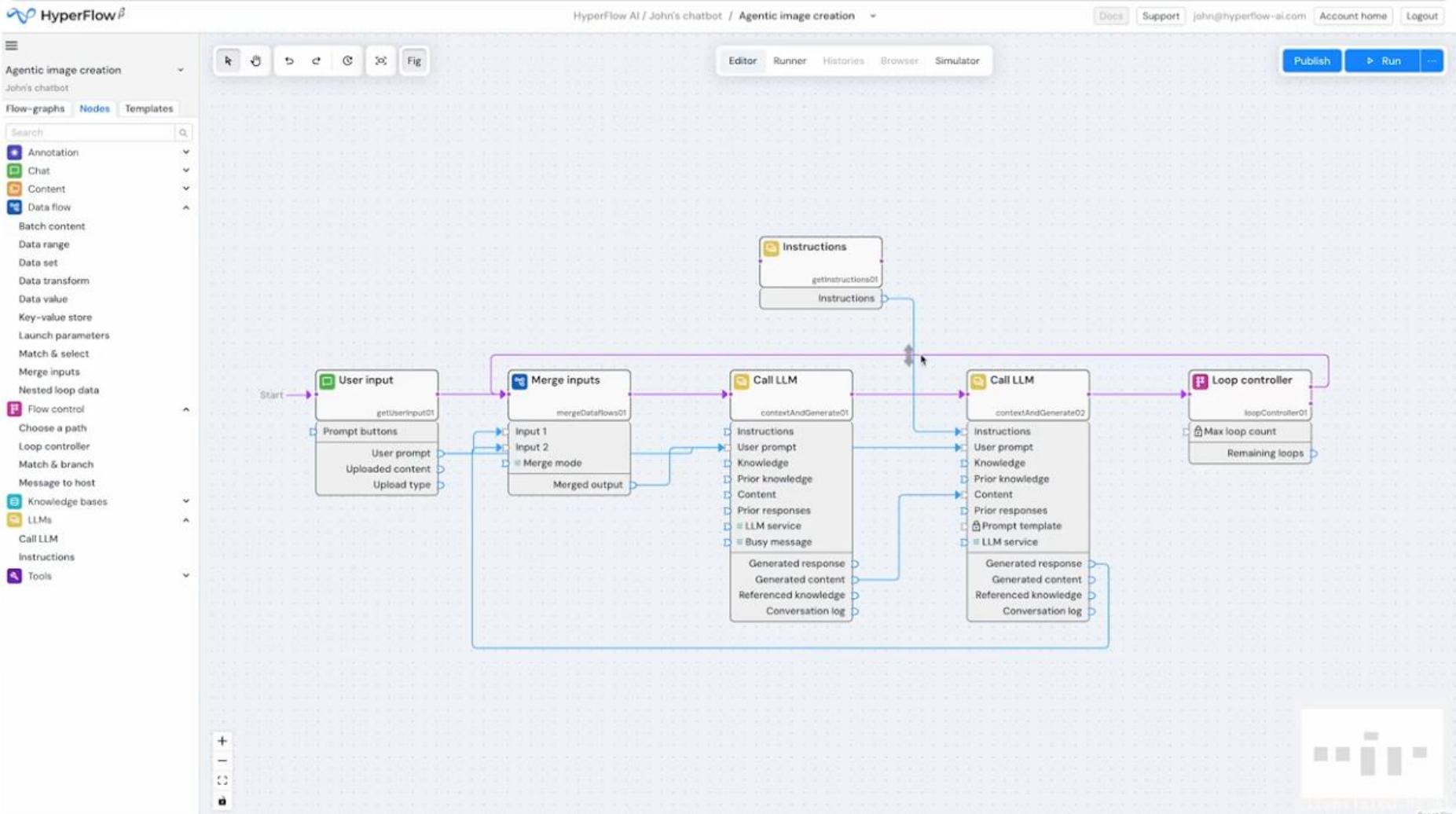
에이전트 AI 애플리케이션 구축



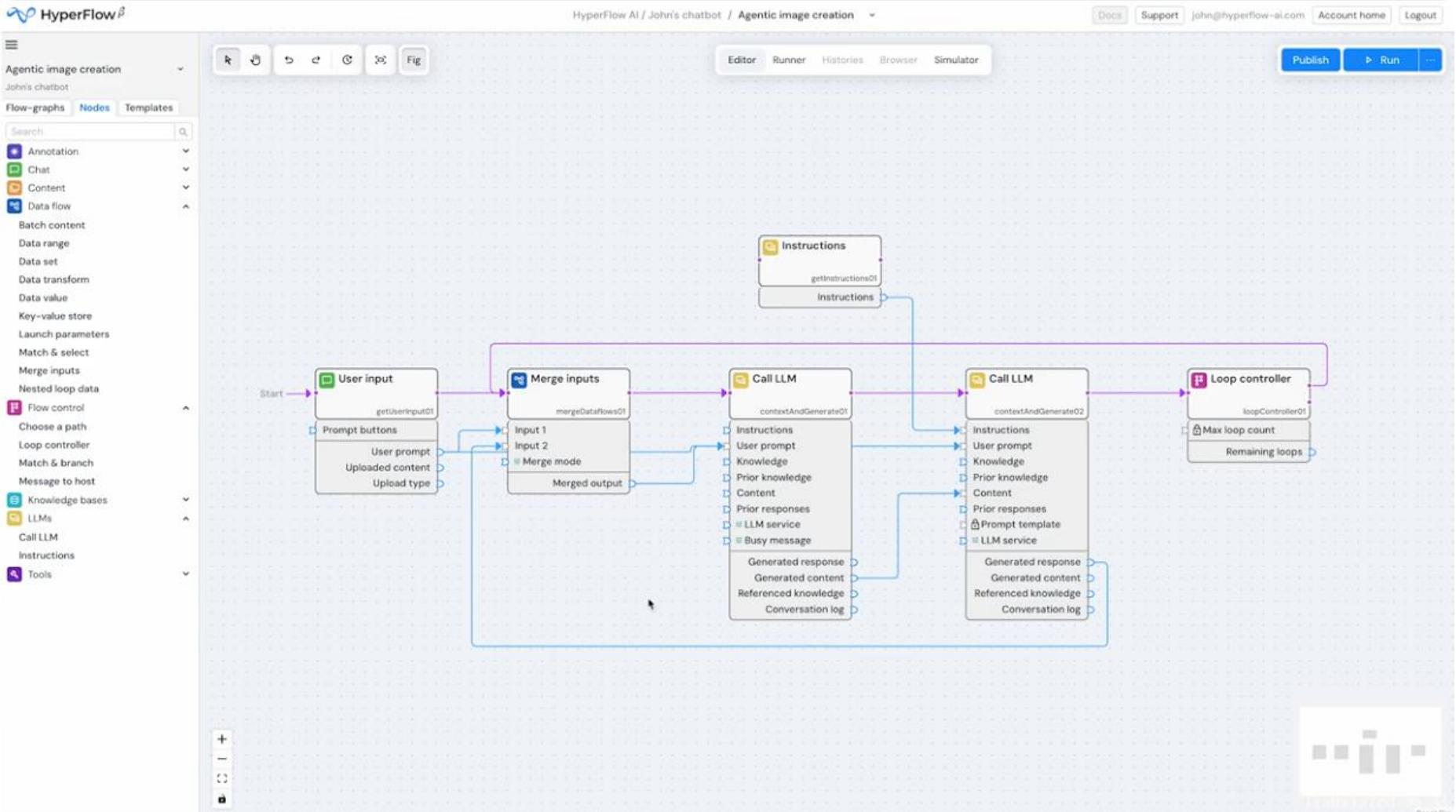
에이전트 AI 애플리케이션 구축



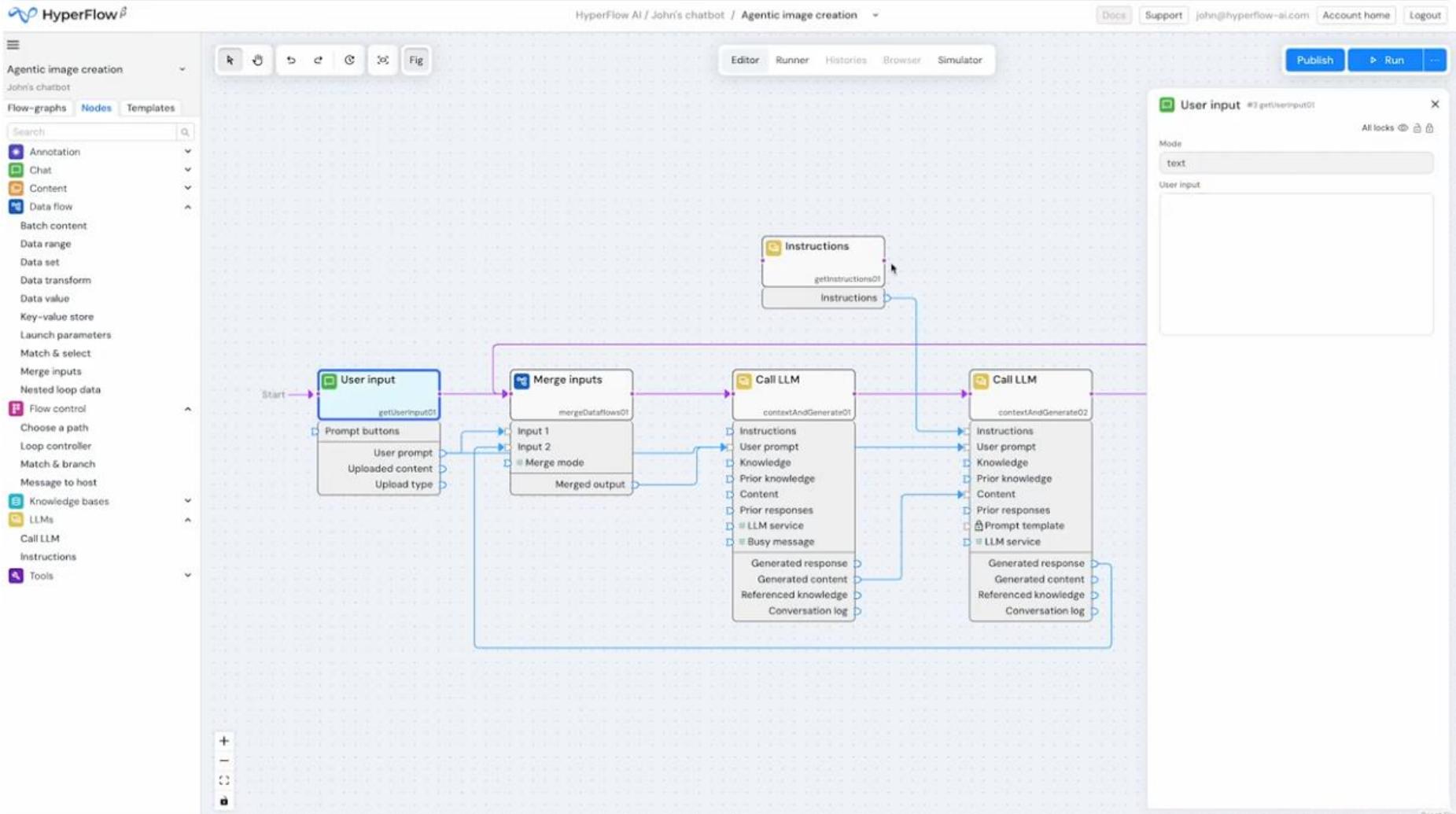
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run ...

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow

Batch content

Data range

Data set

Data transform

Data value

Key-value store

Launch parameters

Match & select

Merge inputs

Nested loop data

Flow control

Choose a path

Loop controller

Match & branch

Message to host

Knowledge bases

LLMs

Call LLM

Instructions

Tools

User input

getUserIdentity01

Prompt buttons

User prompt

Uploaded content

Upload type

Merge inputs

mergeDataflows01

Input 1

Input 2

Merge mode

Merged output

Call LLM

contextAndGenerate01

Instructions

User prompt

Prior knowledge

Content

Prior responses

LLM service

Busy message

Generated response

Generated content

Referenced knowledge

Conversation log

Call LLM

contextAndGenerate02

Instructions

User prompt

Prior knowledge

Content

Prior responses

Prompt template

LLM service

Generated response

Generated content

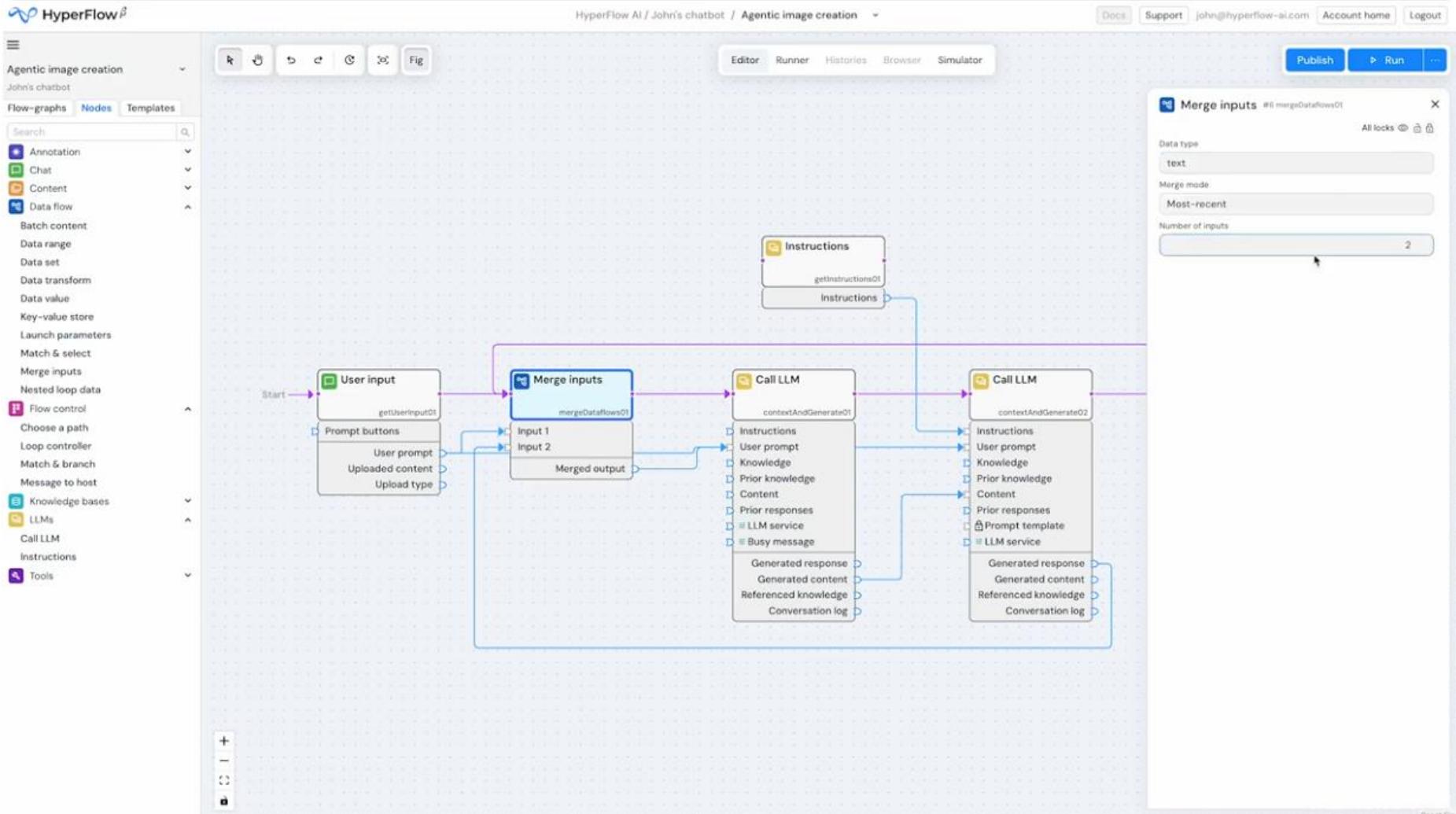
Referenced knowledge

Conversation log

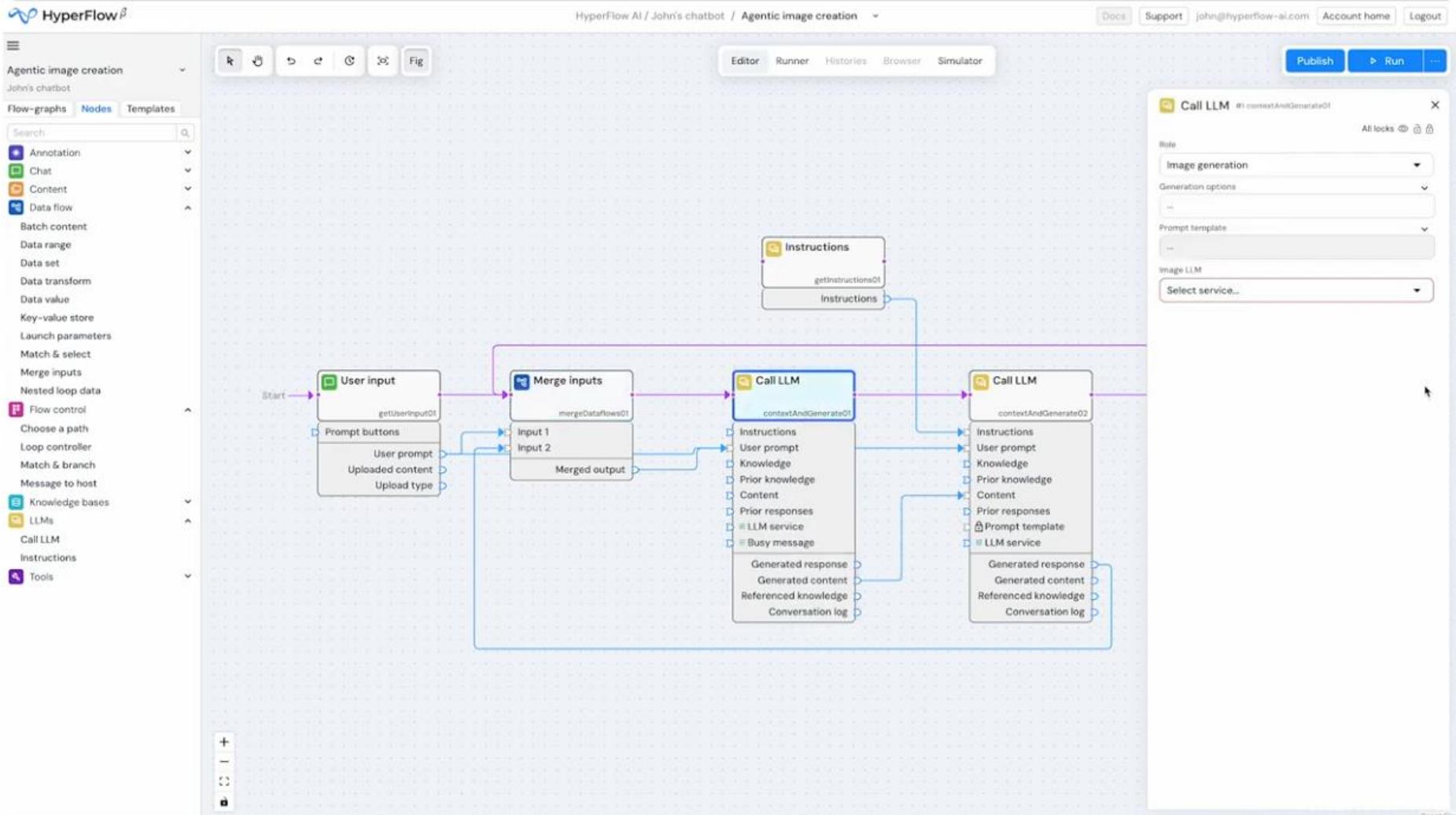
Start

React Flow

에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow

Batch content Data range Data set Data transform Data value Key-value store Launch parameters Match & select Merge inputs Nested loop data

Flow control Choose a path Loop controller Match & branch Message to host

Knowledge bases LLMs Call LLM Instructions Tools

Call LLM #1 contextAndGenerate01

Role: Image generation

Generation options: ...

Prompt template: ...

Image LLM: openai

LLM parameters: ...

LLM model: dall-e-3

Tags: ...

Notes: ...

Call LLM #2 contextAndGenerate02

Instructions: ...

User prompt: ...

Knowledge: ...

Prior knowledge: ...

Content: ...

Prior responses: ...

LLM service: ...

Busy message: ...

LLM model: ...

Tags: ...

Notes: ...

Generated response: ...

Generated content: ...

Referenced knowledge: ...

Conversation log: ...

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation: John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow

Batch content Data range Data set Data transform Data value Key-value store Launch parameters Match & select Merge inputs Nested loop data

Flow control Choose a path Loop controller Match & branch Message to host

Knowledge bases LLMs Call LLM Instructions Tools

```
graph LR; Start((Start)) --> UI[User input]; UI --> MI[Merge inputs]; MI --> C1[Call LLM]; MI --> C2[Call LLM]; C1 --> I1[Instructions]; C1 --> G1[Get instructions]; C1 --> MO[Merged output]; C2 --> I2[Instructions]; C2 --> G2[Get instructions]; C2 --> MO;
```

Call LLM #1 contextAndGenerate01

Role: Image generation

Generation options

Prompt template

Image LLM: openai

LLM parameters

LLM model: dall-e-3

Tags: John, cool images

Notes

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow

Batch content Data range Data set Data transform Data value Key-value store Launch parameters Match & select Merge inputs Nested loop data

Flow control Choose a path Loop controller Match & branch Message to host

Knowledge bases LLMs Call LLM Instructions Tools

Start

User input
get userInput01

Merge inputs
mergeDataflows01

Call LLM
contextAndGenerate01

Call LLM
contextAndGenerate02

Instructions
getInstructions01

Instructions
User prompt

Knowledge

Prior knowledge

Content

Prior responses

LLM service

Busy message

Merged output

Generated response

Generated content

Referenced knowledge

Conversation log

All locks

Role: chat

Generation options: ...

Prompt template: -

LLM service: openai

에이전트 AI 애플리케이션 구축



HyperFlow  HyperFlow AI / John's chatbot / Agentic image creation

Editor Runner Histories Browser Simulator Publish Run

Agentic image creation
John's chatbot

Flow-graphs Nodes Templates

Search

Annotation Chat Content Data flow Batch content Data range Data set Data transform Data value Key-value store Launch parameters Match & select Merge inputs Nested loop data Flow control Choose a path Loop controller Match & branch Message to host Knowledge bases LLMs Call LLM Instructions Tools

Start

User input

Merge inputs

Call LLM

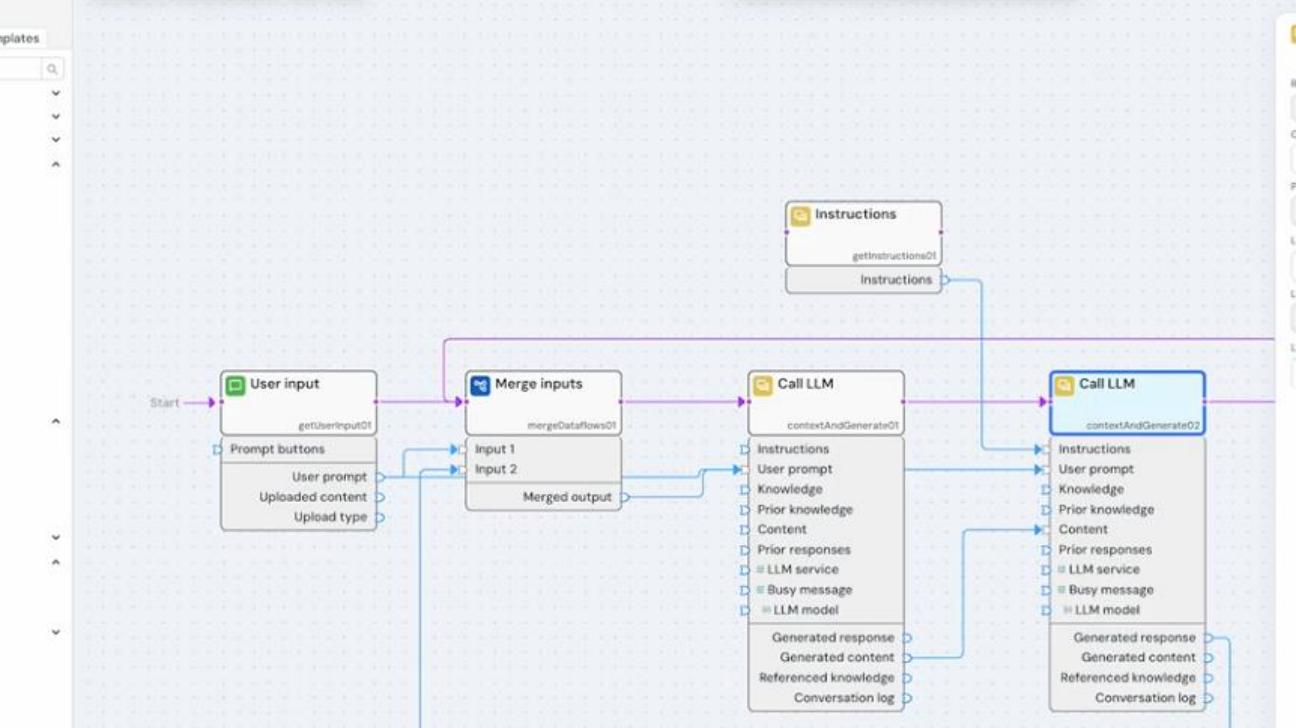
Call LLM

Instructions

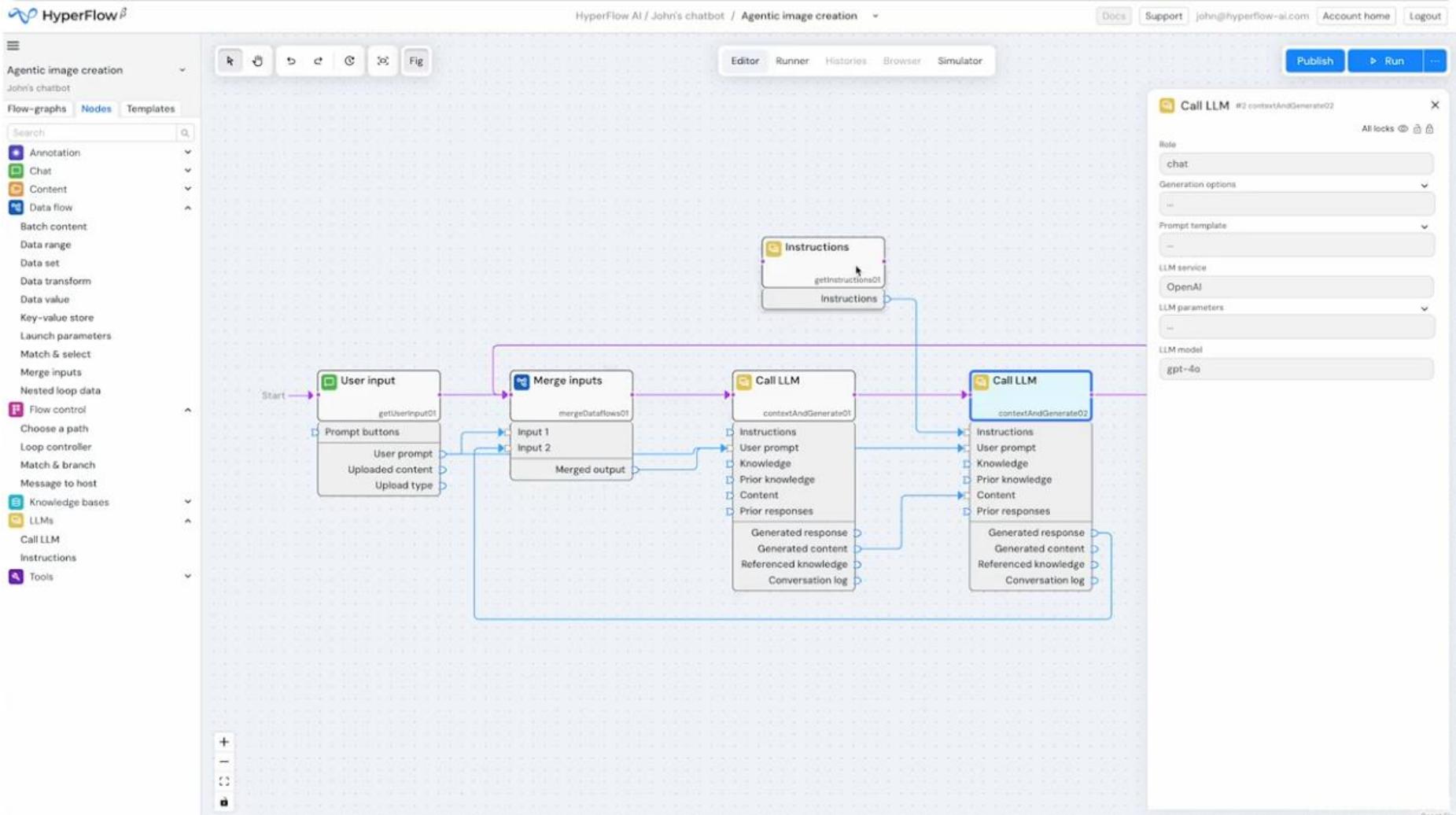
Generated response Generated content Referenced knowledge Conversation log

Generated response Generated content Referenced knowledge Conversation log

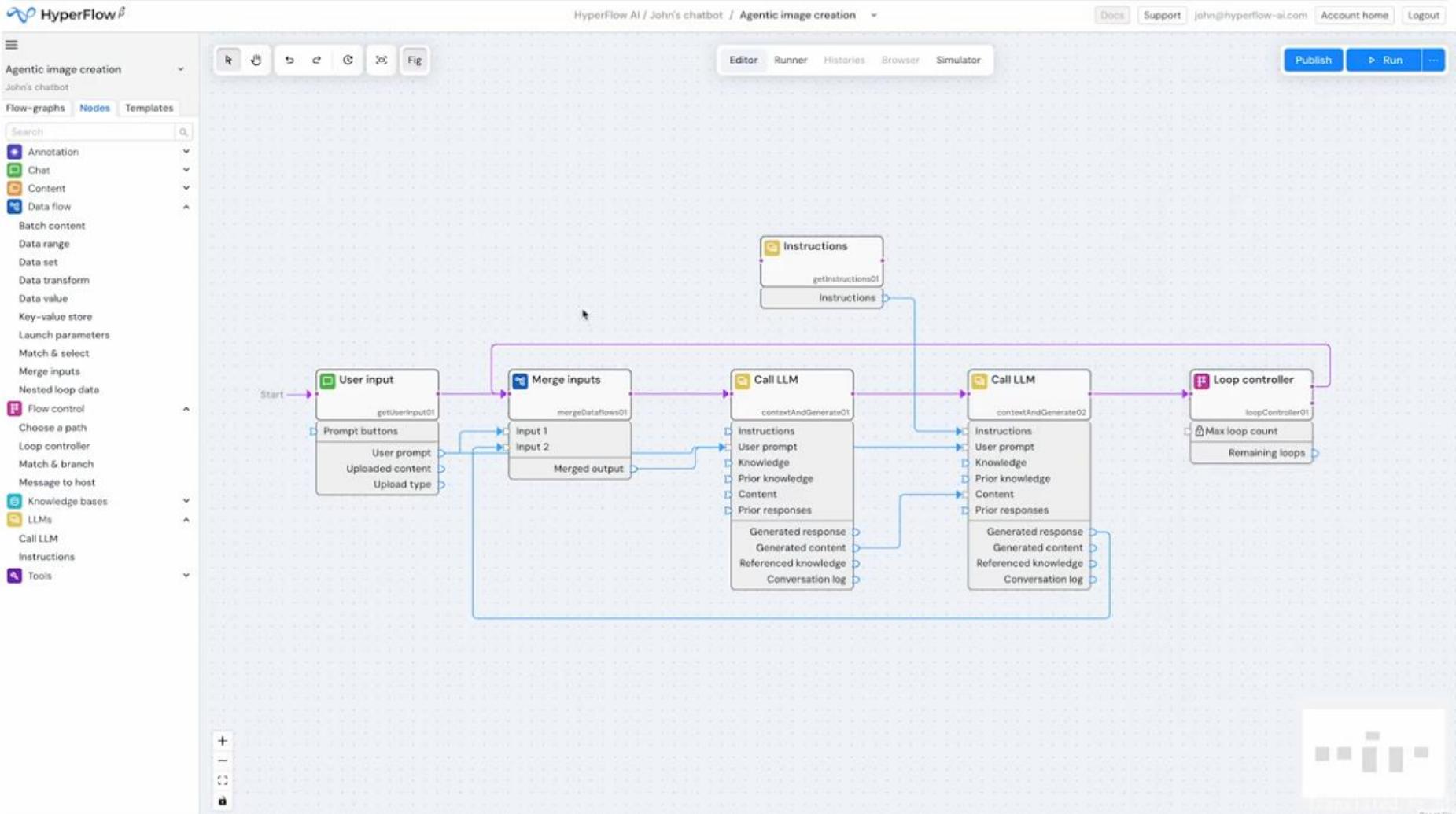
Role: chat
Generation options: ...
Prompt template: ...
LLM service: openai
LLM parameters: ...
LLM model: gpt-4o



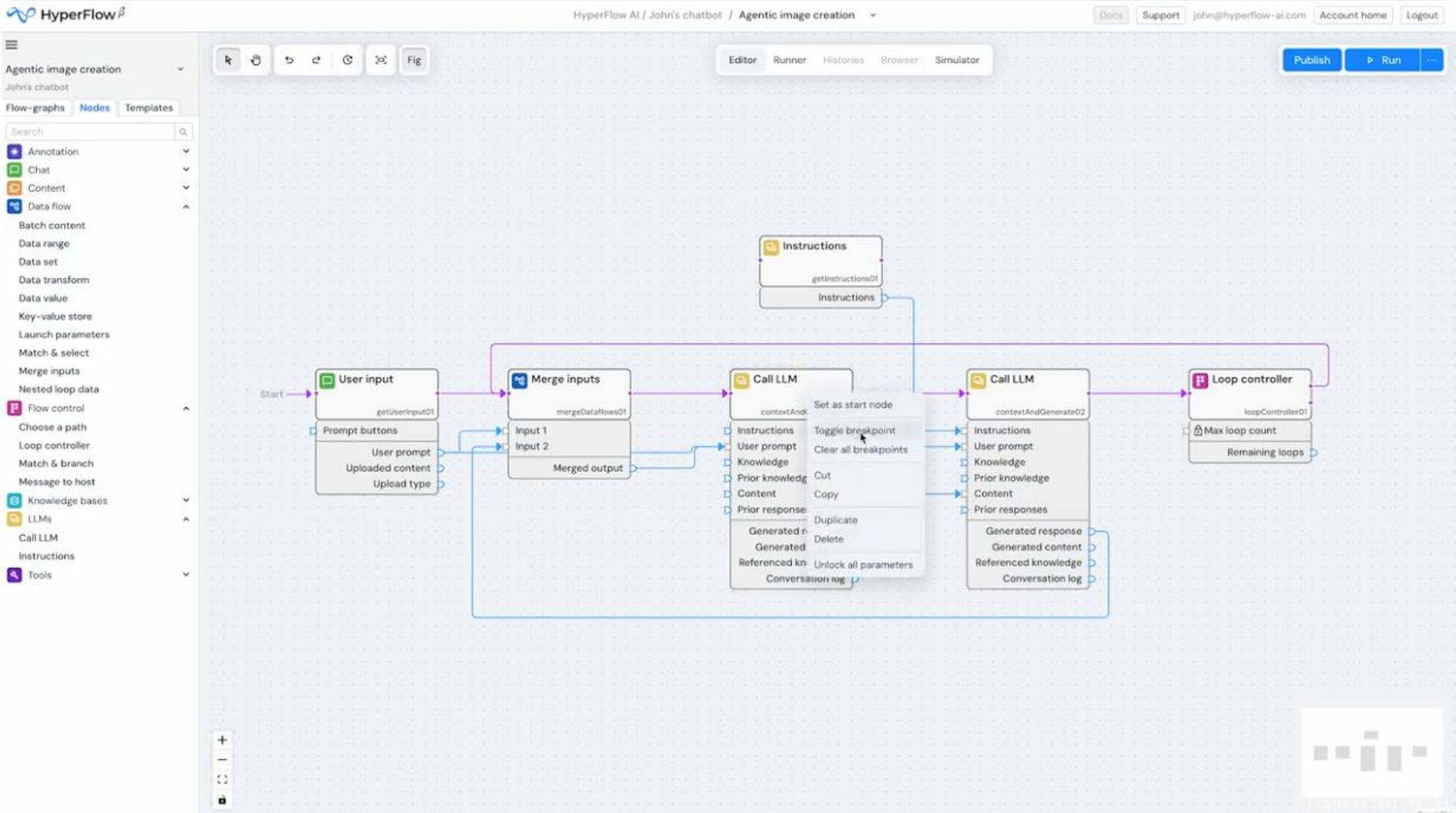
에이전트 AI 애플리케이션 구축



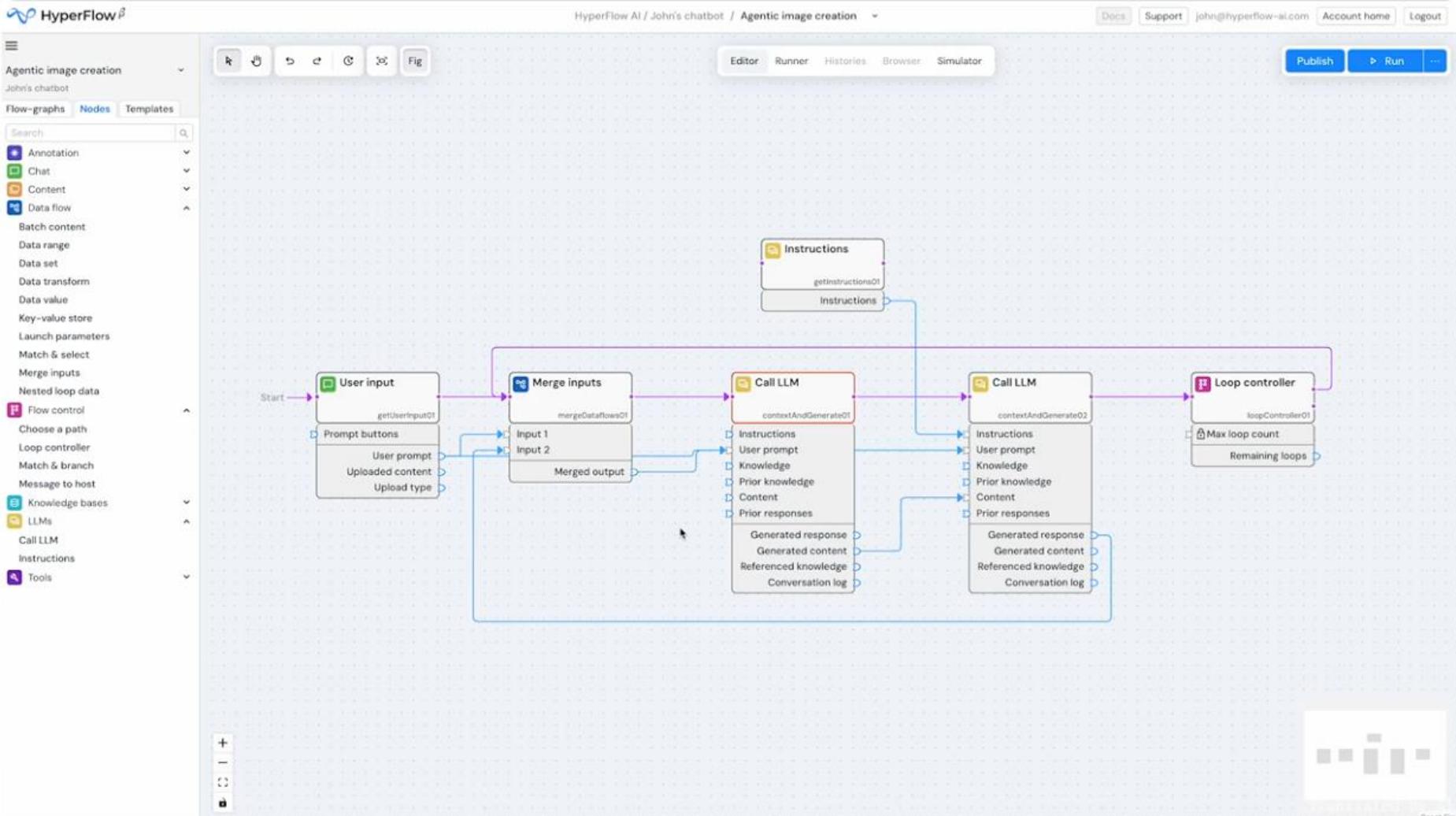
에이전트 AI 애플리케이션 구축



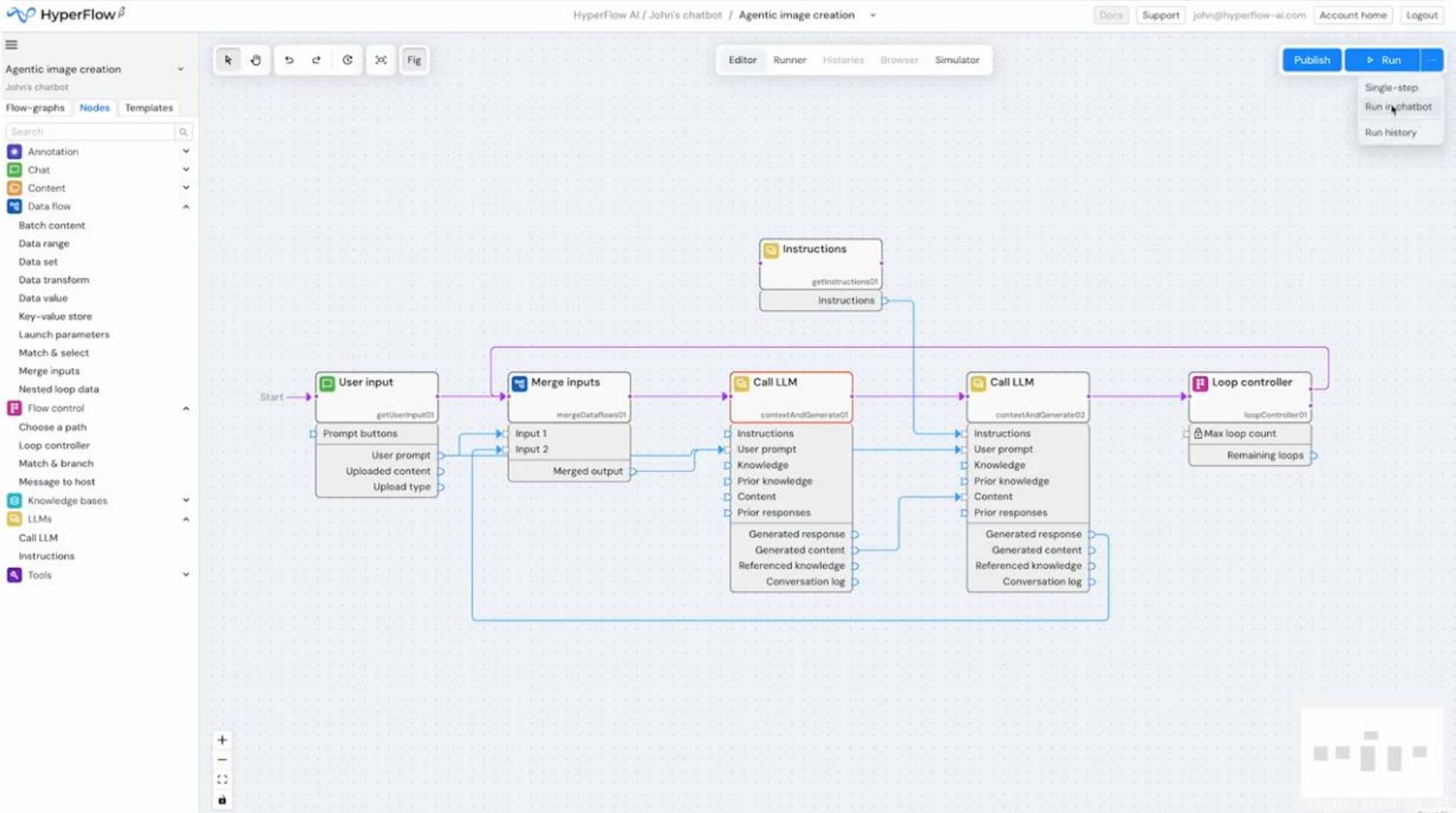
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

HyperFlow AI Chat Bot

Greetings from Agentic image creation!

User input #3 getUserId01 Step 1

All locks ⊕ ⊖ ⊕

Mode text

Prompt input in test chatbot

OK

User input #3 getUserId01 Step 3

Waiting on settings OK...

Run session tags:

React Flow

Translated by AI

에이전트 AI 애플리케이션 구축



The screenshot displays the HyperFlow AI platform interface. At the top, the title bar shows "HyperFlow AI / John's chatbot / Agentic image creation" and includes "Edit", "Docs", "Support", "john@hyperflow-ai.com", "Account home", and "Logout" buttons. The main toolbar has icons for back, forward, search, and file operations, with "Fig" selected.

The left panel features a "HyperFlow AI Chat Bot" window with the message "Greetings from Agentic image creation! A penguin wearing a red coat skating happily on ice." Below this is a "Instructions" block with a "getInstruction01" action and an output "Instructions".

The central workspace contains two "Call LLM" blocks. The first "Call LLM" block (Step 1) has an input "Instructions" from the previous step and outputs "Generated response", "Generated content", "Referenced knowledge", and "Conversation log". The second "Call LLM" block (Step 2) receives these outputs as inputs and also outputs "Generated response", "Generated content", "Referenced knowledge", and "Conversation log".

The right panel shows the execution steps:

- Step 1:** "User input #3 gerUserInput01" (Step 1)
- Step 2:** "Merge inputs #6 mergeDataflows01" (Step 2)
- Step 3:** "Call LLM #1 contextAndGenerate01" (Step 3). Status: "Waiting on next step..."

The bottom right corner of the interface includes a "Feedback" link.

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Stop Restart Stop Continue

Run session tags

Greetings from Agentic image creation!

A penguin wearing a red coat skating happily on ice.

HyperFlow AI Chat Bot

Instructions

getinstruction01

Instructions

Call LLM #1 contextAndGenerate01 Step 3

All locks

Role: image

Generation options

Prompt template

Image LLM: OpenAI

User input #3 getUserInput01 Step 1

Parameters

Merge inputs #6 mergeDataflows01 Step 2

Parameters

Step output: Most Recent Value: A penguin wearing a red coat skating happily on ice.

Call LLM #1 contextAndGenerate01 Step 3

Waiting on settings OK...

OK

Instructions

contextAndGenerate01

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Call LLM contextAndGenerate01

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

React Flow

Translated by AI

The screenshot shows the HyperFlow AI application interface. At the top, there's a navigation bar with 'HyperFlow ^β', 'HyperFlow AI / John's chatbot / Agentic image creation', 'Edit', 'Docs', 'Support', 'john@hyperflow-ai.com', 'Account home', and 'Logout'. Below the navigation is a toolbar with icons for back, forward, search, and a 'Fig' button. The main area has tabs for 'Editor', 'Runner', 'Histories', 'Browser', and 'Simulator'. On the right, there are buttons for 'Stop', 'Restart', 'Stop', and 'Continue'. A 'Run session tags' section is present. On the left, a 'HyperFlow AI Chat Bot' window displays a greeting and a sample image description: 'A penguin wearing a red coat skating happily on ice.' Below this is a 'Instructions' block with a 'getinstruction01' step. In the center, there's a 'Call LLM #1 contextAndGenerate01 Step 3' block. This block has a 'Role' field set to 'image', 'Generation options', and 'Image LLM: OpenAI'. It also includes 'Prompt template', 'User input #3 getUserInput01 Step 1', 'Merge inputs #6 mergeDataflows01 Step 2', and a 'Step output' section showing the generated image description. Another 'Call LLM #1 contextAndGenerate01 Step 3' block is shown below, indicating it's waiting on settings. On the left side of the main canvas, there's a 'React Flow' panel containing a flowchart with nodes like 'Instructions', 'Call LLM', and 'Merge inputs', connected by arrows. The bottom right corner of the interface has a 'Translated by AI' note.

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Step Restart Stop Continue

Run session tags

HyperFlow AI Chat Bot

Greetings from Agentic image creation!

A penguin wearing a red coat skating happily on ice.

Instructions

getInstructions()

Instructions

contextAndGenerate()

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Call LLM #1 contextAndGenerate()

Step 3

All locks

Role image

Generation options

Prompt template

Image LLM OpenAI

LLM parameters

LLM model dall-e-3

Tags John's cool images

Notes

User input #3 gerUserInput()

Merge inputs #6 mergeDataflows()

Step output

Most Recent Value: A penguin wearing a red coat skating happily on ice.

Call LLM #1 contextAndGenerate()

Step 3

Parameters

Composite prompt

Type Content

1 user **User Query** A penguin wearing a red coat skating happily on ice.

Agent tokens 12

Source step 2

Call LLM: ...

React Flow

Translated by AI

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Step Restart Stop Continue

Run session tags

Parameters

Composite prompt

Type Content Approx tokens Source step

1 user **User Query** A penguin wearing a red coat skating happily on ice. 12 2

Generated image

https://hyperflow-ai.com/api/hyperflow/file/get/media/6778cf688ea9c026dea2412

Metadata

Model: dall-e-3 | Created: 2025-01-04 15:04 | Elapsed: 11.81s | Prompt Tokens: 0 | Completion Tokens: 0

Instructions #4 getinstructions01 Step 4

Waiting on next step...

React Flow

The screenshot displays the HyperFlow AI application interface. On the left, there is a 'HyperFlow AI Chat Bot' window showing a penguin wearing a red coat skating on ice. Below it, a flowchart illustrates the process: 'User prompt' leads to 'Input 2' and 'Merged output'. These feed into a 'Call LLM' block labeled 'contextAndGenerate01', which also receives 'Instructions' from another 'Instructions' block. This block outputs 'Generated response', 'Generated content', 'Referenced knowledge', and 'Conversation log'. Simultaneously, the 'Instructions' block sends 'Instructions' to a second 'Call LLM' block labeled 'contextAndGenerate02', which also receives 'User prompt', 'Knowledge', 'Prior knowledge', 'Content', and 'Prior responses'. It outputs 'Generated response', 'Generated content', 'Referenced knowledge', and 'Conversation log'. The right side of the interface shows the generated image, its URL, metadata (including completion and prompt tokens), and session details.

에이전트 AI 애플리케이션 구축



HyperFlow β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags Pause

HyperFlow AI Chat Bot

A penguin wearing a red coat skating happily on ice.

Call LLM #2 contextAndGenerate02 Step 5

Role: chat
Generation options: ...
Prompt template: ...
LLM service: OpenAI ...

OK

Loop controller

Max loop count: 10
Remaining loops: 9

Generated response
Generated content
Referenced knowledge
Conversation log

Metadata

Model: dall-e-3 | Created: 2025-01-04 15:04 | Elapsed: 11.8ms | Prompt Tokens: 0 | Completion Tokens: 0

Instructions #4 getInstructions01

Parameters

Instructions

Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dall-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

Call LLM #2 contextAndGenerate02 Step 5

Waiting on settings OK...

React Flow

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

Pause

HyperFlow AI Chat Bot

A penguin wearing a red coat skating happily on ice.

Call LLM #2 contextAndGenerate02 Step 5

Role: chat
Generation options:
Prompt template:
LLM service: OpenAI
LLM parameters:
LLM model: gpt-4o

All locks:

Instructions: User prompt Knowledge Prior knowledge Content Prior responses

Generated response: Generated content Referenced knowledge Conversation log

Loop controller

Max loop count: Remaining loops:

OK

Metadata

Model: dall-e-3 | Created: 2023-01-04 15:04 | Elapsed: 0 | Prompt Tokens: 0 | Completion Tokens: 0

Instructions #4 getInstructionsQ1 Step 4

Parameters

Instructions: Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dalle-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

Call LLM #2 contextAndGenerate02 Step 5

Parameters

Composite prompt

Type	Content	Approx. tokens	Source step
1. Instructions	Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dalle-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.	85	
2. user	{ dataType: ..., data: ..., mimeType: ... }	9	3
3. user	**User Query** A penguin wearing a red coat skating happily on ice.	12	1

Call LLM: ...

Translated by

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Step Restart Stop Continue ...

Run session tags

Call LLM #1 contextAndGenerate01 Step 8

Call LLM #2 contextAndGenerate02 Step 5

Generated text

A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs under a clear blue sky, creating a cheerful winter scene. The penguin's eyes are closed in happiness, and its cheeks are flushed, enhancing the joyful expression.

Metadata

Model: gpt-4a-2024-08-06 | Created: 2025-01-04 15:06 | Flipped: 5.2% | Prompt Tokens: 680 | Completion Tokens: 67

Loop controller #5 loopController01 Step 6

Merge inputs #6 mergeDataflows01 Step 7

Step output

Most Recent Value: A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs under a clear blue sky, creating a cheerful winter scene. The penguin's eyes are closed in happiness, and its cheeks are flushed, enhancing the joyful expression.

Call LLM #7 contextAndGenerate01 Step 8

Waiting on next step...

React Flow

HyperFlow AI Chat Bot

The screenshot displays the HyperFlow AI application interface. On the left, there is a preview window titled "HyperFlow AI Chat Bot" showing a cartoon penguin in a red coat skating on ice. Below the preview is a descriptive text block. In the center, a flowchart illustrates the data processing steps. The flow starts with an "Instructions" node, followed by two "Call LLM" nodes (Step 8 and Step 5), a "Loop controller" node (Step 6), a "Merge inputs" node (Step 7), and ends with another "Call LLM" node (Step 8). Various parameters and configurations are visible for each step, such as "Content" and "Knowledge". The right side of the interface contains detailed logs and metadata for each step, including generated text, prompt examples, and completion statistics. The top navigation bar includes links for "Docs", "Support", "Logout", and user account information.

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Step Restart Stop Continue

Run session tags

Call LLM #1 contextAndGenerate01 Step 8

Role: image
Generation options:
Prompt template:
Image LLM: OpenAI

OK

Call LLM #2 contextAndGenerate02 Step 5

Parameters

Composite prompt

Type	Content	Approx. tokens	Source step
1. instructions	Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in [datatype: ... data: ... mindatatype: ...]	85	8
2. user	"User Query": A penguin wearing a red coat skating joyfully on ice.	3	9
3. user	"User Query": A penguin wearing a red coat skating joyfully on ice.	12	10

Generated text

A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs under a clear blue sky, creating a cheerful winter scene. The penguin's eyes are closed in happiness, and its cheeks are flushed, enhancing the joyful expression.

Metadata

Model: gpt-4a-2024-04-06 | Created: 2025-01-04 15:06 | Elapsed: 5.2ms | Prompt Tokens: 880 | Completion Tokens: 67

Loop controller #5 loopController01 Step 6

Parameters

Merge inputs #6 mergeDataflows01 Step 7

Parameters

Step output

Most Recent Value: A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs under a clear blue sky, creating a cheerful winter scene. The penguin's eyes are closed in happiness, and its cheeks are flushed, enhancing the joyful expression.

Call LLM #1 contextAndGenerate01 Step 8

Waiting on settings OK...

HyperFlow Chat Bot

A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs under a clear blue sky, creating a cheerful winter scene. The penguin's eyes are closed in happiness, and its cheeks are flushed, enhancing the joyful expression.

Instructions

getinstruction01

Instructions

contextAndGenerate01

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Instructions

contextAndGenerate02

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Step Restart Stop Continue

Run session tags

Parameters

Composite prompt

Type Content Approx tokens Source step

1 user **User Query** A cute penguin wearing a bright red coat with white buttons and a matching scarf, skating joyfully on a shiny ice surface. The background shows tall, majestic icebergs 68 7

Generated image

The screenshot displays the HyperFlow AI application interface. On the left, there is a detailed flowchart titled "HyperFlow AI Chat Bot". The flowchart starts with a "User prompt" input, which branches into "Input 2" and "Merged output". "Input 2" feeds into a "Call LLM" block with "contextAndGenerate01". "Merged output" also feeds into this block. The "Call LLM" block has several outputs: "Instructions", "User prompt", "Knowledge", "Prior knowledge", "Content", and "Prior responses". These outputs feed into another "Call LLM" block with "contextAndGenerate02". This second "Call LLM" block has similar outputs: "Generated response", "Generated content", "Referenced knowledge", and "Conversation log". The right side of the interface shows the results of the process. It includes a preview of the generated image (a cartoon penguin in a red coat), a "Metadata" section with details like Model: dall-e-3, Created: 2025-01-04 15:07, Elapsed: 12.18s, Prompt Tokens: 0, Completion Tokens: 0, and a status message "Waiting on next step...". Above the preview, there are sections for "Instructions", "Parameters", and "Composite prompt".

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause ...

Run session tags

HyperFlow AI Chat Bot
happily skating on ice surrounded by a winter wonderland. The background is a snowy landscape with icy cliffs and snow-covered trees. Sunlight sparkles on the ice, and snowflakes gently fall from the sky, adding a whimsical touch to the scene.

art —

User prompt
Uploaded content
Upload type

Instructions
getInstructionsOf

Call LLM #2 contextAndGenerate02 Step 20

Instructions
contextAndGenerate02

Instructions
User prompt
Knowledge
Prior knowledge
Content
Prior responses

Generated response
Generated content
Referenced knowledge
Conversation log

Call LLM #4 getInstructionsOf Step 19

Parameters

Instructions

Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dalle-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

Call LLM #2 contextAndGenerate02 Step 20

Waiting on settings OK...

HyperFlow

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause ...

Run session tags

dall-e-3 2025-01-04 15:08 12.11s 0 0 Step 19

Instructions #4 getInstructions01

Parameters

Instructions

Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dall-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

Call LLM #2 contextAndGenerate02

Parameters

Composite prompt

Type Content Approx tokens Source step

- 1. instructions Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in 340
- 2. user { dataType: ..., data: ..., mimeType: ... } 9 1
- 3. user **User Query** A penguin wearing a red coat skating happily on ice. Incorporate snowy trees and icy cliffs in the background for a magical winter setting. 12 1

Generated text

A cute penguin wearing a red coat skating joyfully on ice. The penguin is surrounded by a winter wonderland with snowflakes falling gently. The sun is shining brightly, reflecting off the icy surface, creating a cheerful and festive atmosphere. Incorporate snowy trees and icy cliffs in the background for a magical winter setting.

Metadata

Model dall-e-3 Created 2025-01-04 15:08 Elapsed 8.23s Prompt Tokens 880 Completion Tokens 65

Loop controller #5 loopController01

Parameters

Waiting on settings OK...

End of run Translated by AI

React Flow

The screenshot shows the HyperFlow AI application interface. On the left, there is a preview window titled "HyperFlow AI Chat Bot" displaying a cartoon penguin in a red coat skating on ice. Below the preview is a text box containing a user prompt: "A cute penguin wearing a red coat skating joyfully on ice. The penguin is surrounded by a winter wonderland with snowflakes falling gently. The sun is shining brightly, reflecting off the icy surface, creating a cheerful and festive atmosphere. Incorporate snowy trees and icy cliffs in the background for a magical winter setting." To the right of the preview is a flowchart editor showing the internal logic. The flowchart starts with a "User input" node, which branches into "Merge inputs" and "Call LLM". The "Merge inputs" node combines "Input 1" and "Input 2" into a "Merged output". This output is then used as input for the "Call LLM" node. The "Call LLM" node has several parameters: "Instructions", "User prompt", "Knowledge", "Prior knowledge", "Content", "Prior responses", "Generated response", "Generated content", "Referenced knowledge", and "Conversation log". On the far right, the "Runner" tab is active, showing the execution steps: Step 19 (Instructions), Step 20 (Call LLM), and Step 21 (Loop controller). Each step shows its parameters and the generated text or metadata. The "Generated text" section for Step 20 contains the final image description: "A cute penguin wearing a red coat skating joyfully on ice. The penguin is surrounded by a winter wonderland with snowflakes falling gently. The sun is shining brightly, reflecting off the icy surface, creating a cheerful and festive atmosphere. Incorporate snowy trees and icy cliffs in the background for a magical winter setting."

에이전트 AI 애플리케이션 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Agentic image creation Edit

Docs Support john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

Generated image

A cute penguin wearing a red coat skating joyfully on ice. The penguin is surrounded by a winter wonderland with snowflakes falling gently. The sun is shining brightly, reflecting off the icy surface, creating a cheerful and festive atmosphere. Incorporate snowy trees and icy cliffs in the background for a magical winter setting.

User input

- get userInput01
- Prompt buttons
- User prompt
- Uploaded content
- Upload type

Merge inputs

- Input 1
- Input 2
- Merged output

Call LLM

- contextAndGenerate01
- Instructions
- User prompt
- Knowledge
- Prior knowledge
- Content
- Prior responses
- Generated response
- Generated content
- Referenced knowledge
- Conversation log

React Flow

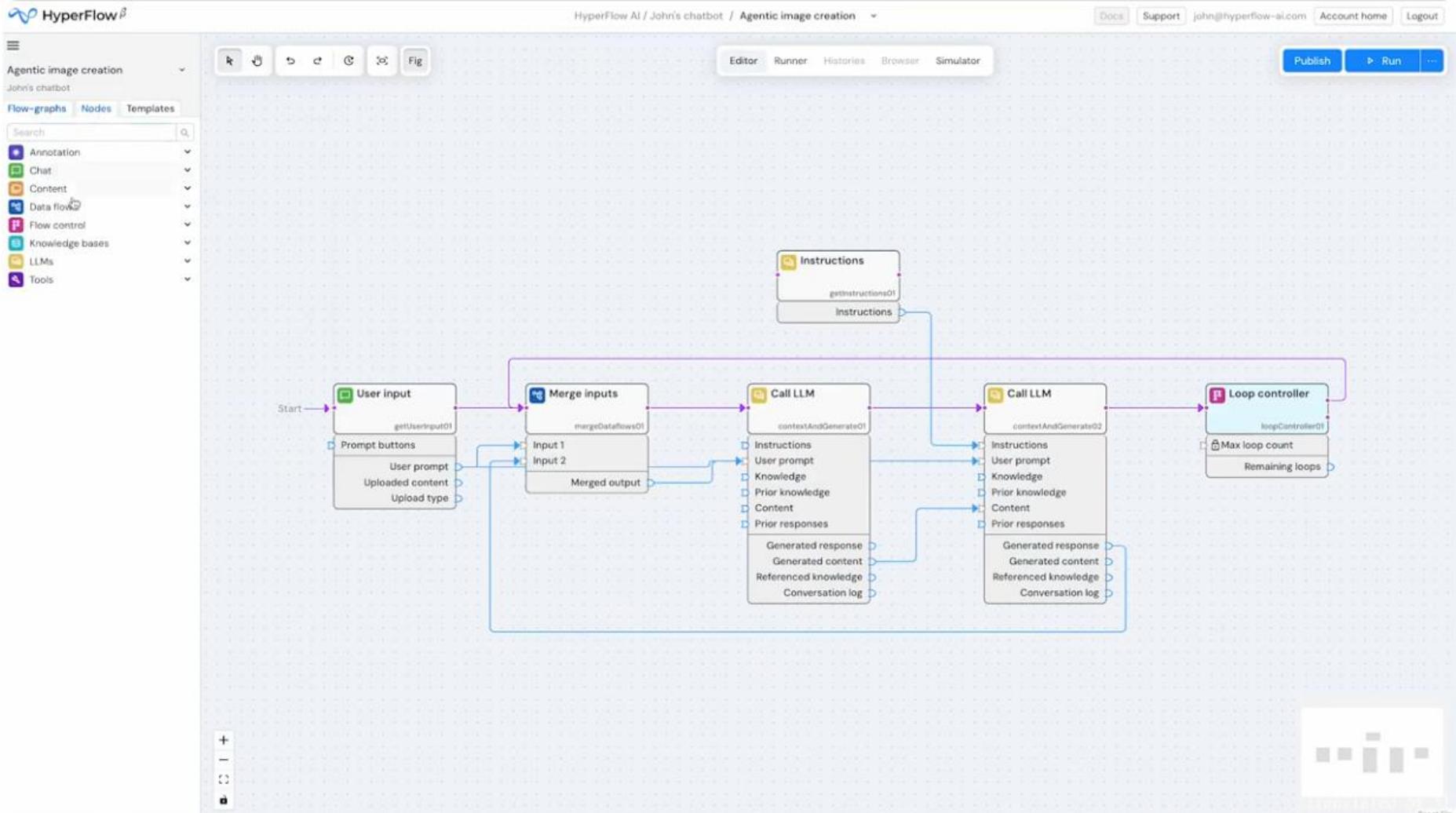
Instructions Step 14

Parameters

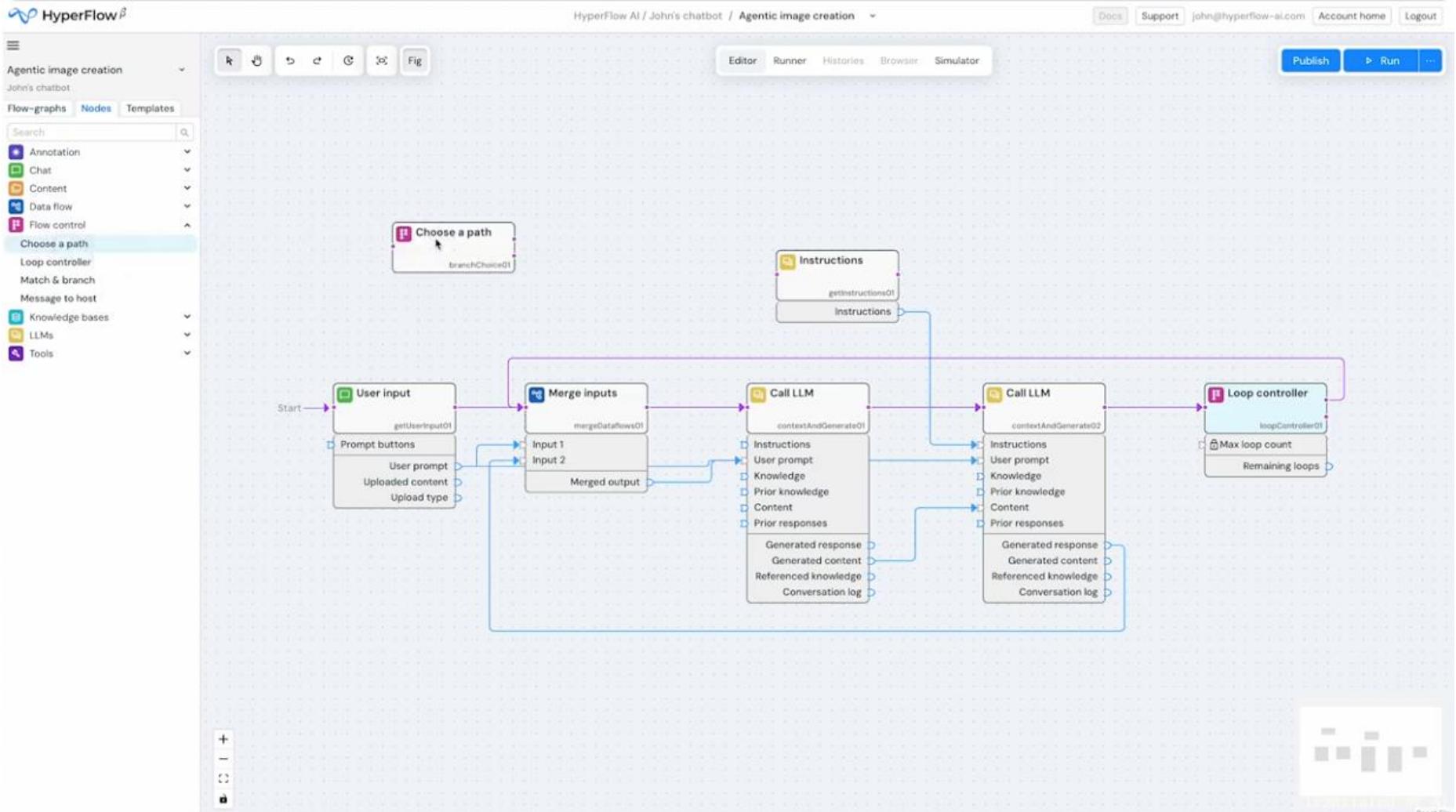
Instructions

Review the user prompt as a request for an image and analyze the supplied image relative to the user's original request prompt, generating suggested improvements in the form of a detailed prompt for Dalle-e-3. Don't necessarily repeat the structure & content of the supplied image, try variations that might fit the original prompt better. You must output only the detailed Dalle-3 prompt, no other comments or critiques.

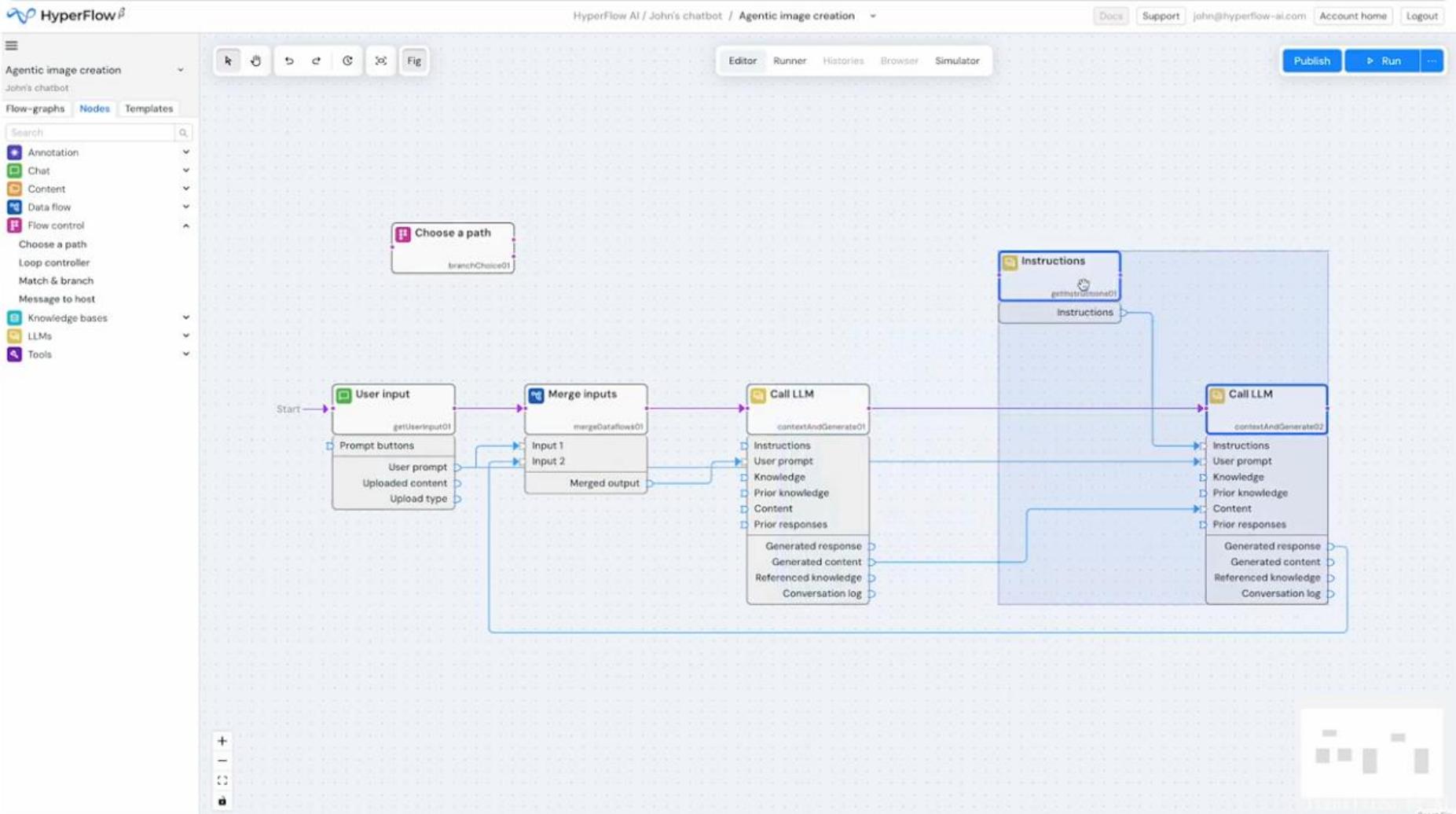
에이전트 AI 애플리케이션 구축



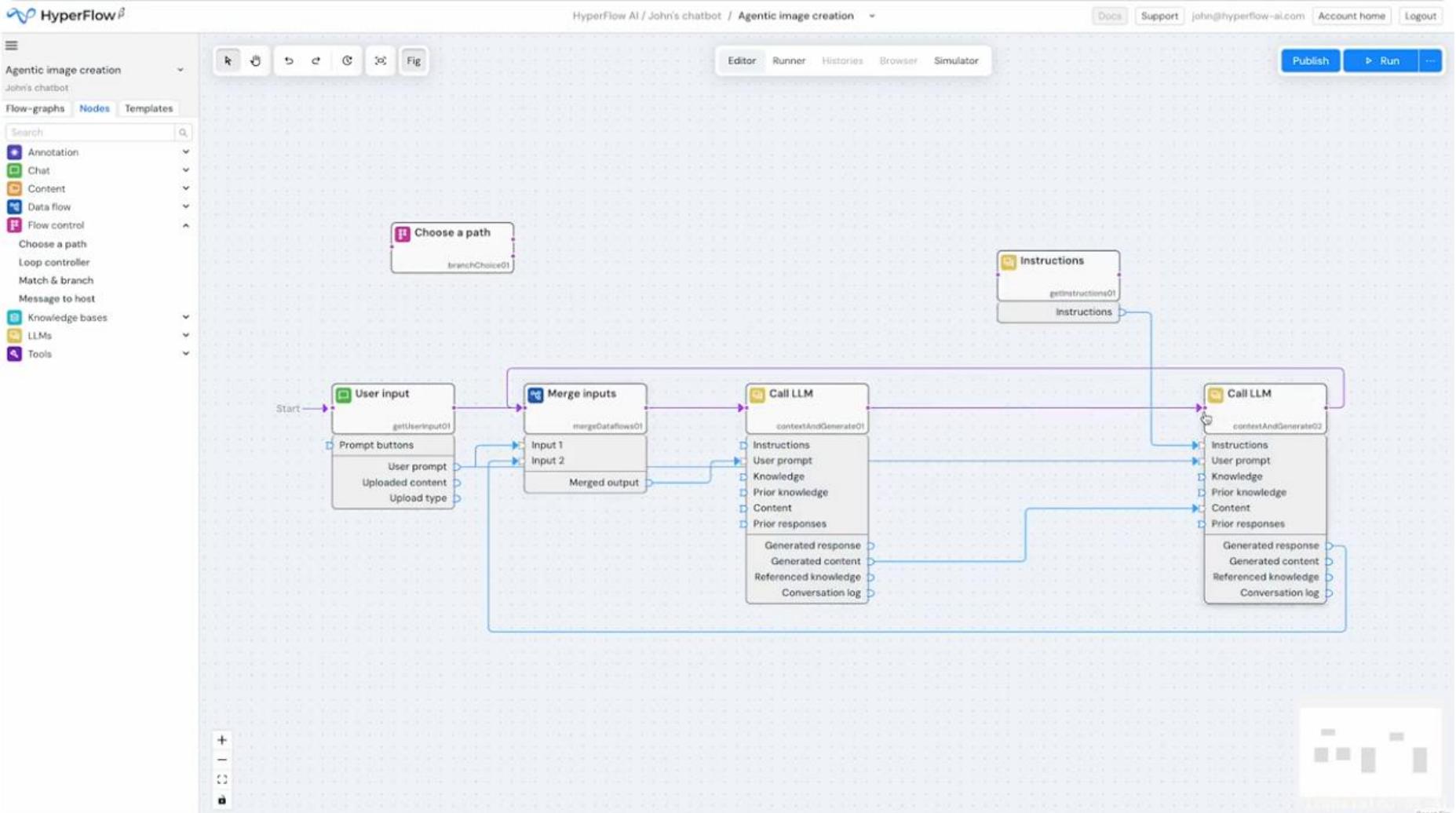
에이전트 AI 애플리케이션 구축



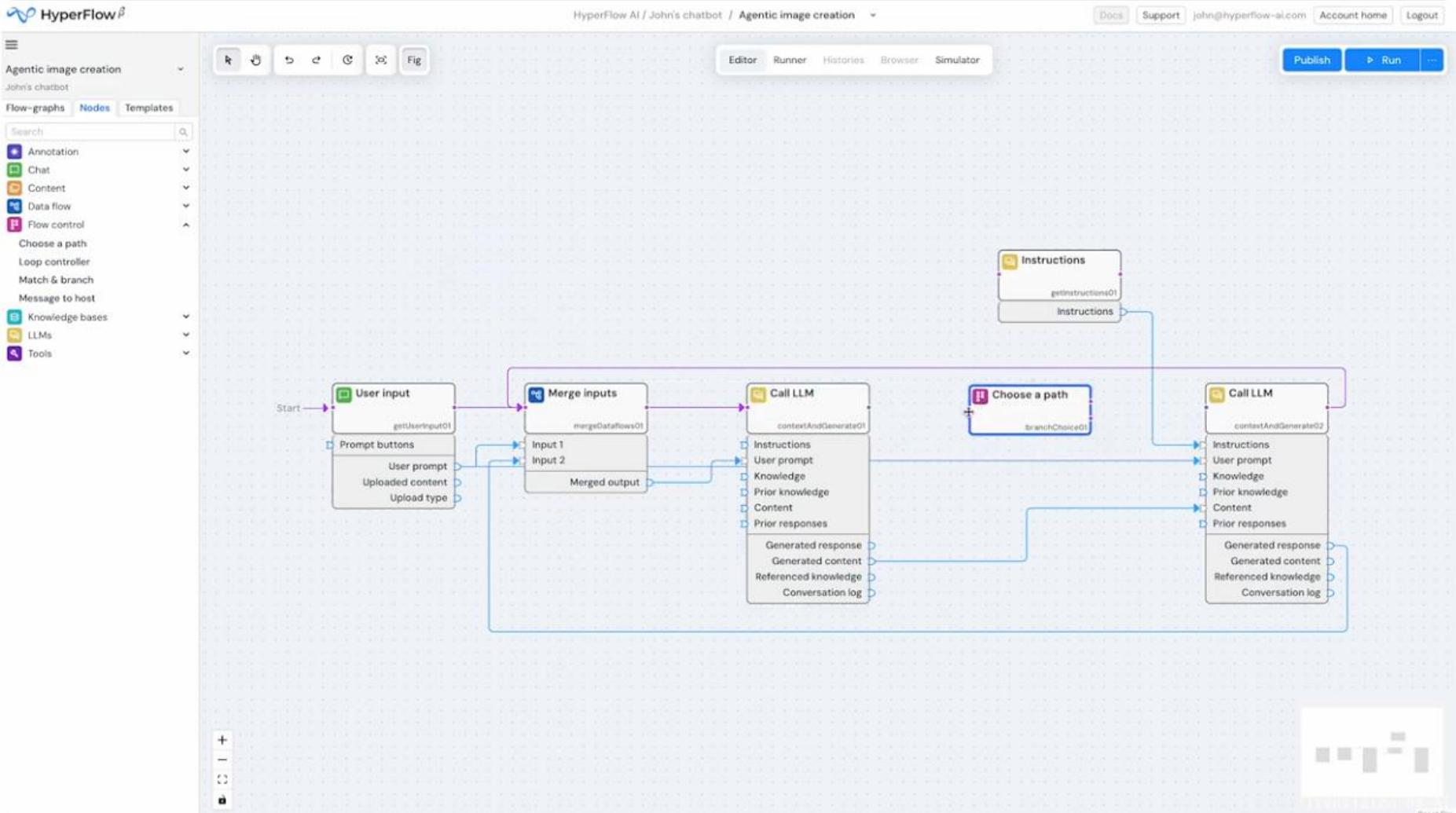
에이전트 AI 애플리케이션 구축



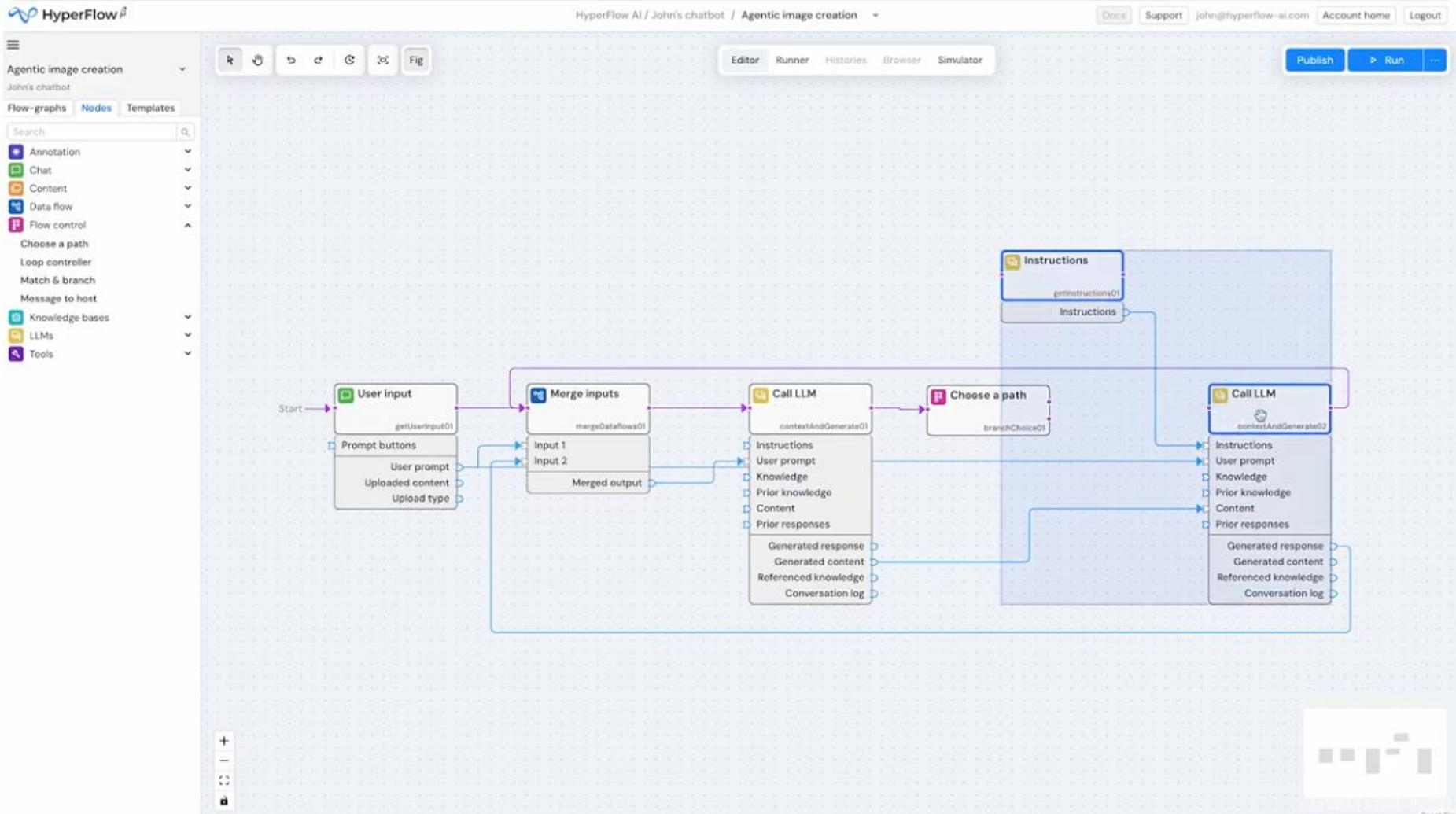
에이전트 AI 애플리케이션 구축



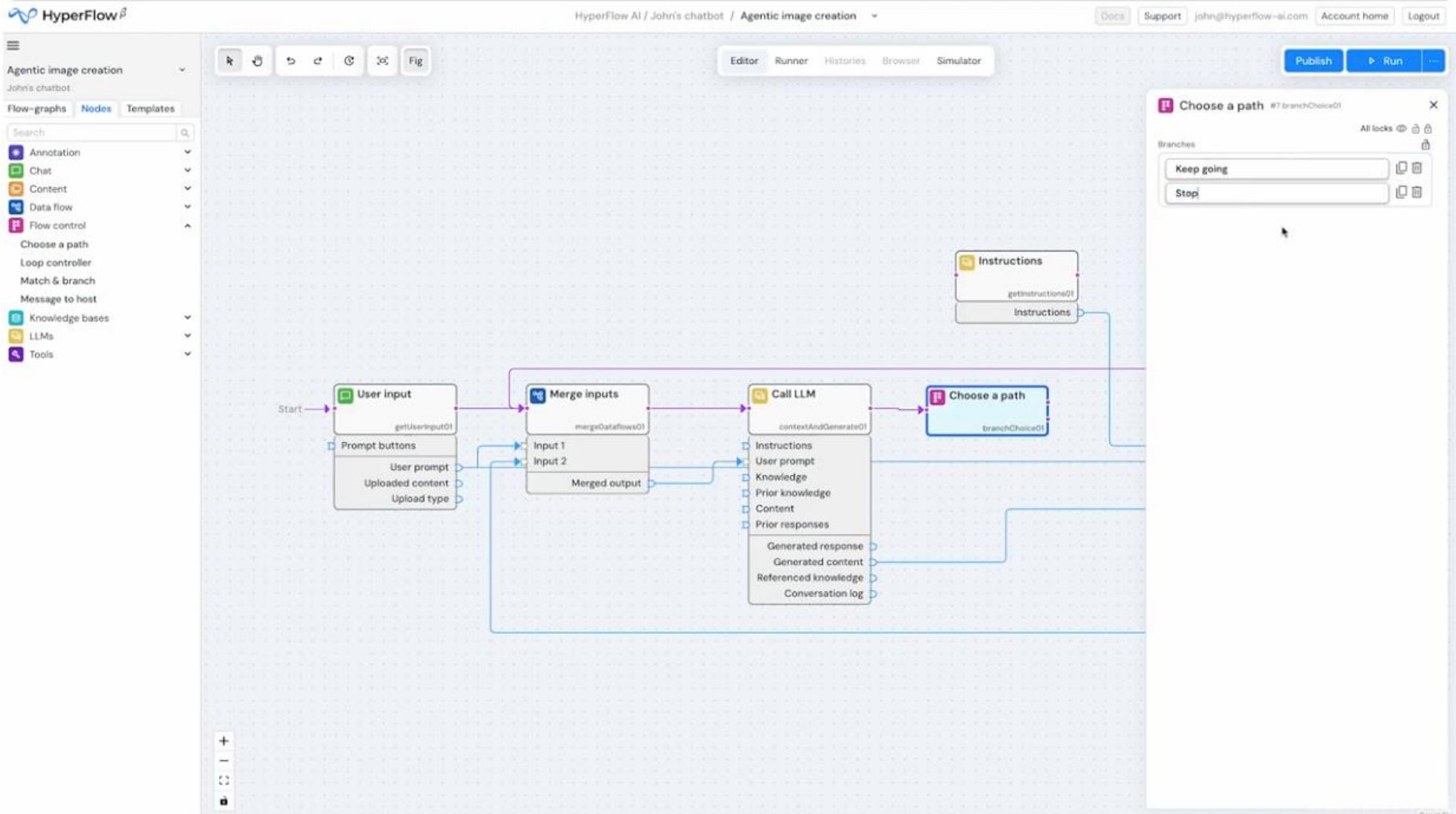
에이전트 AI 애플리케이션 구축



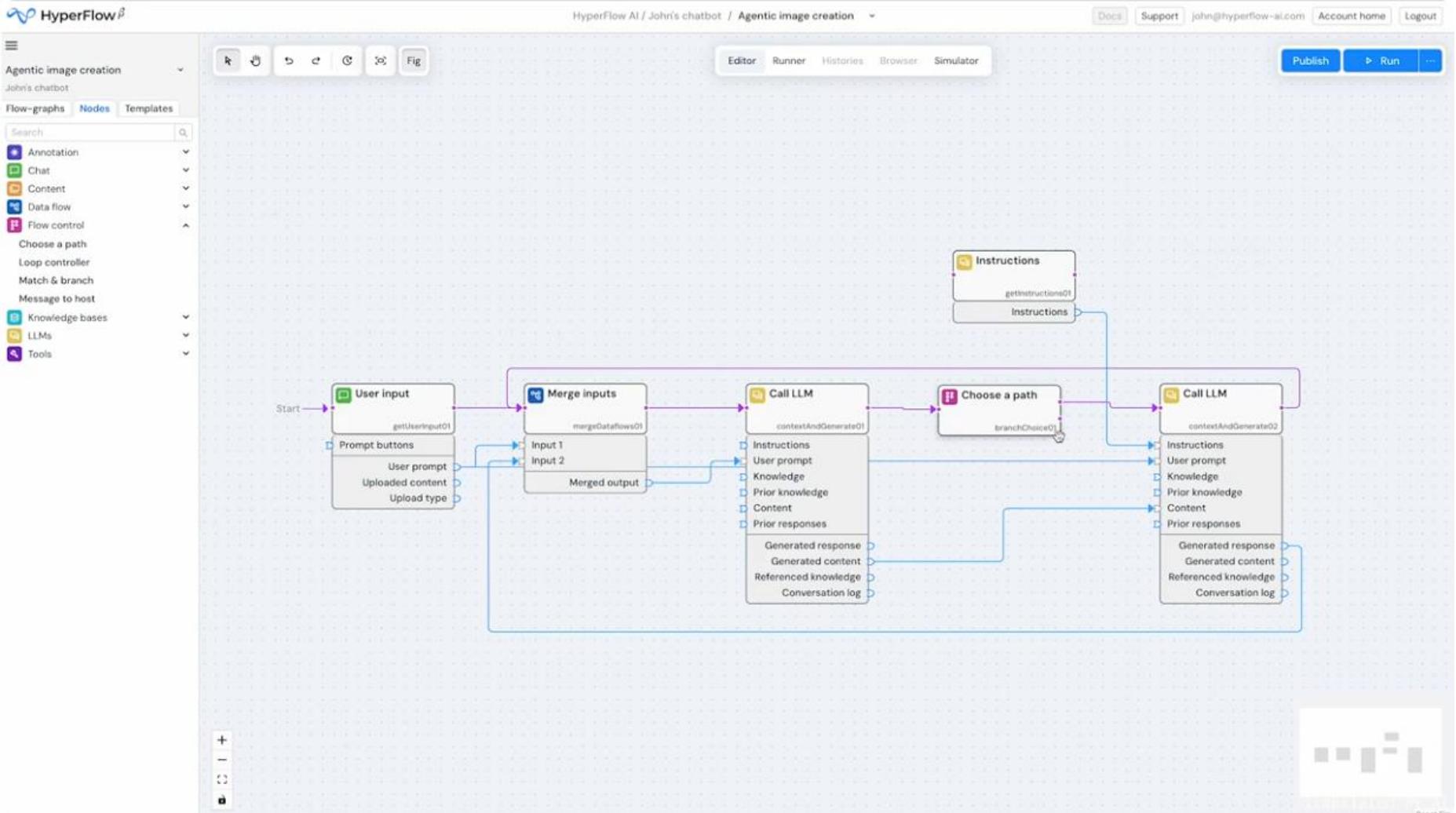
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



HyperFlow^β

Docs Support john@hyperflow-ai.com Account home Logout

Create Organization

Organizations

HyperFlow AI owner

Members

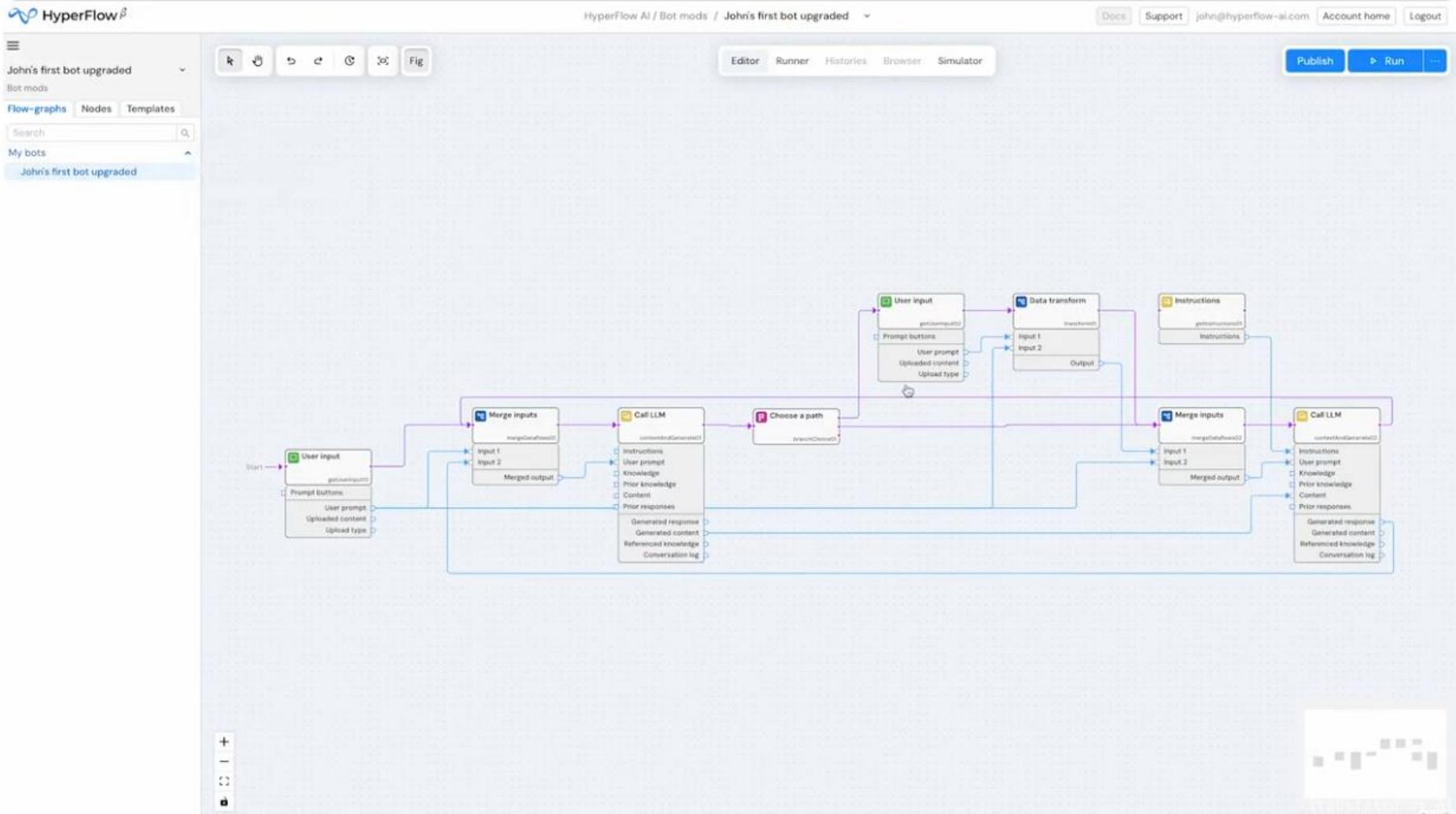
john@hyperflow-ai.com: owner x
+ Invite member

Projects

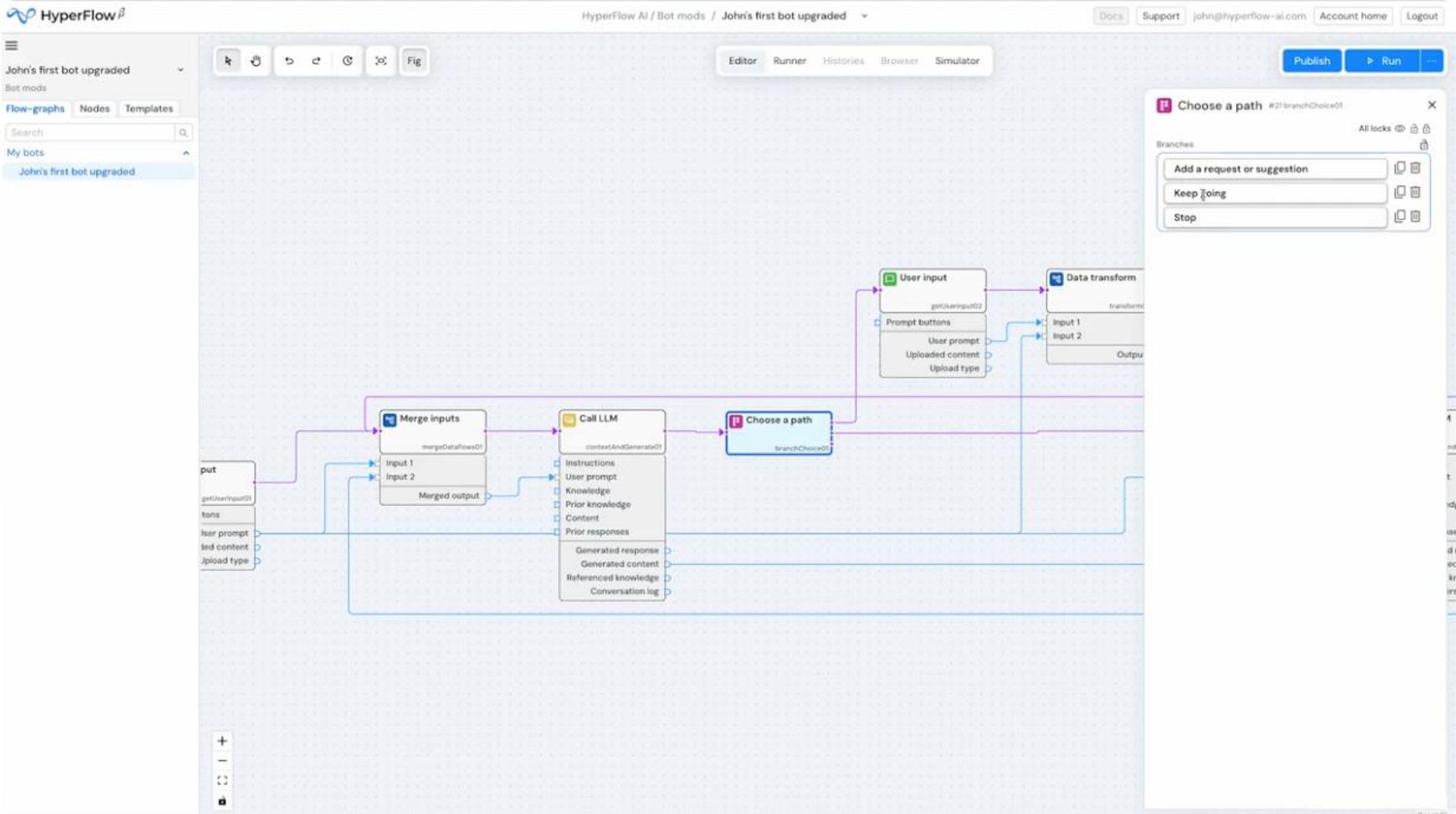
John's chatbot →
Bot mods →
+ Create Project



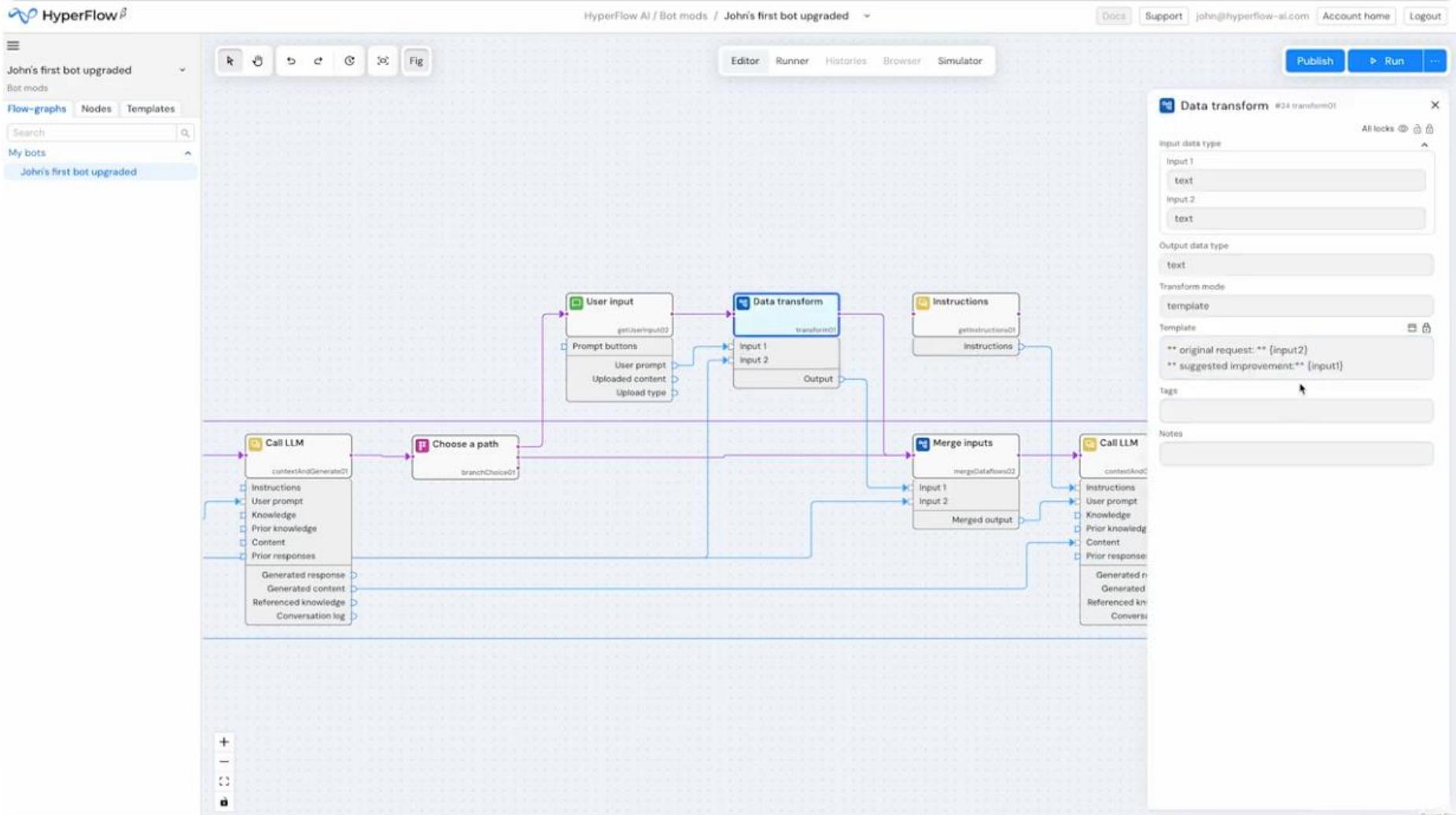
에이전트 AI 애플리케이션 구축



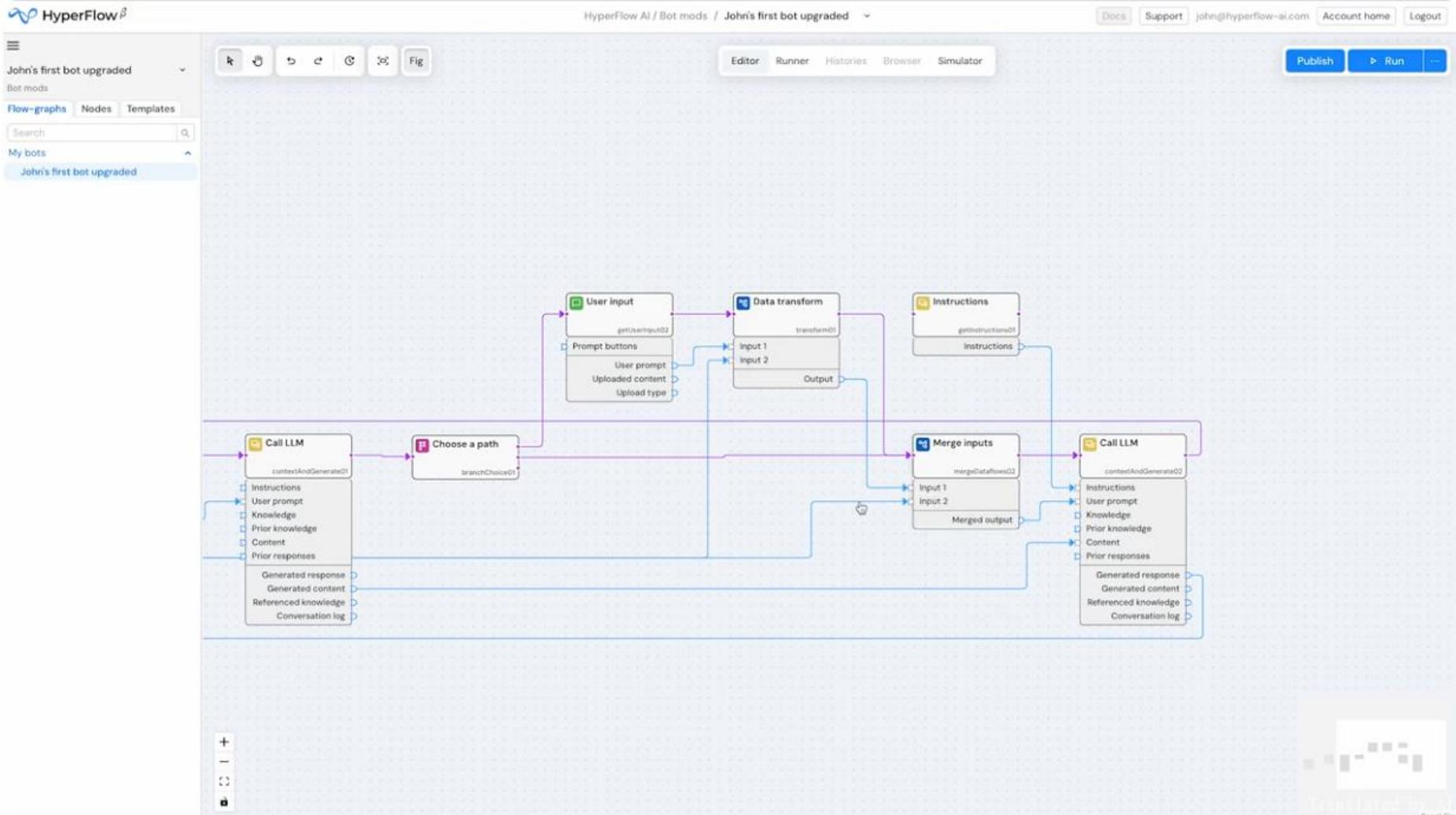
에이전트 AI 애플리케이션 구축



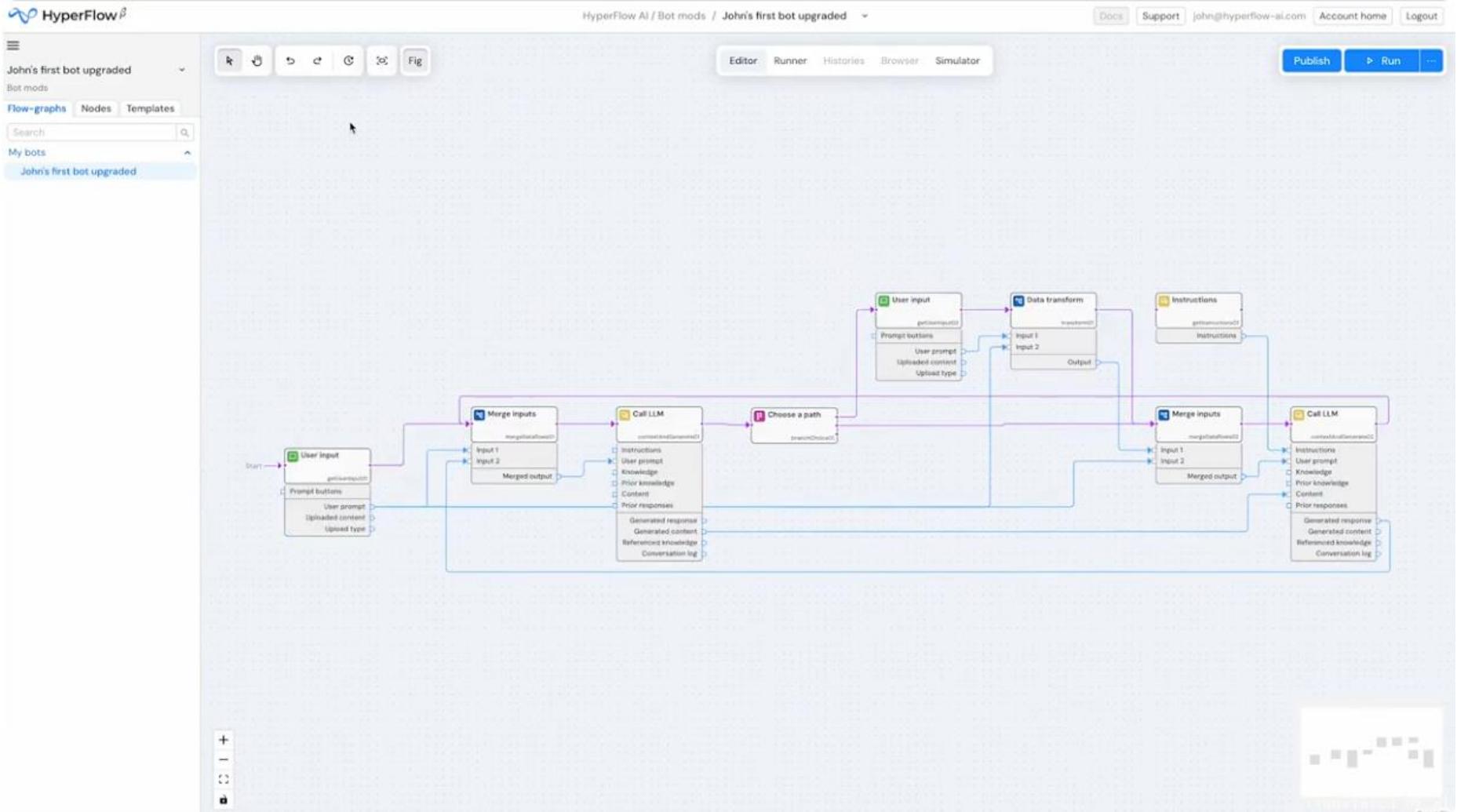
에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



에이전트 AI 애플리케이션 구축



Building a Knowledge Base

Building a Knowledge Base
and a ChatBot expert in that knowledge

Building a Knowledge Base and a Domain-specific Chatbot

Building a Knowledge Base and a Domain-specific Chatbot

- ✖ Model fine-tuning
 - Difficult
 - Time-consuming
 - Expensive

Building a Knowledge Base and a Domain-specific Chatbot

RAG – Retrieval Augmented Generation

RAG – Retrieval Augmented Generation

- Build a domain knowledge database
- Retrieve relevant knowledge chunks
- Send them to them LLM with the query
- Along with instructions to use the knowledge
- Generate a response using that knowledge

RAG 기반 지식 DB 및 챗봇 구축



RAG – Retrieval Augmented Generation

- ▶ Build a domain knowledge database
- ▶ Retrieve relevant knowledge chunks
- ▶ Send them to them LLM with the query
- ▶ Along with instructions to use the knowledge
- ▶ Generate a response using that knowledge

RAG - 검색 증강 생성

- 도메인 지식 데이터베이스 구축
- 관련 지식 청크 검색
- 쿼리와 함께 LLM으로 보내기
- 지식 사용 지침과 함께
- 그 지식을 사용하여 응답 생성

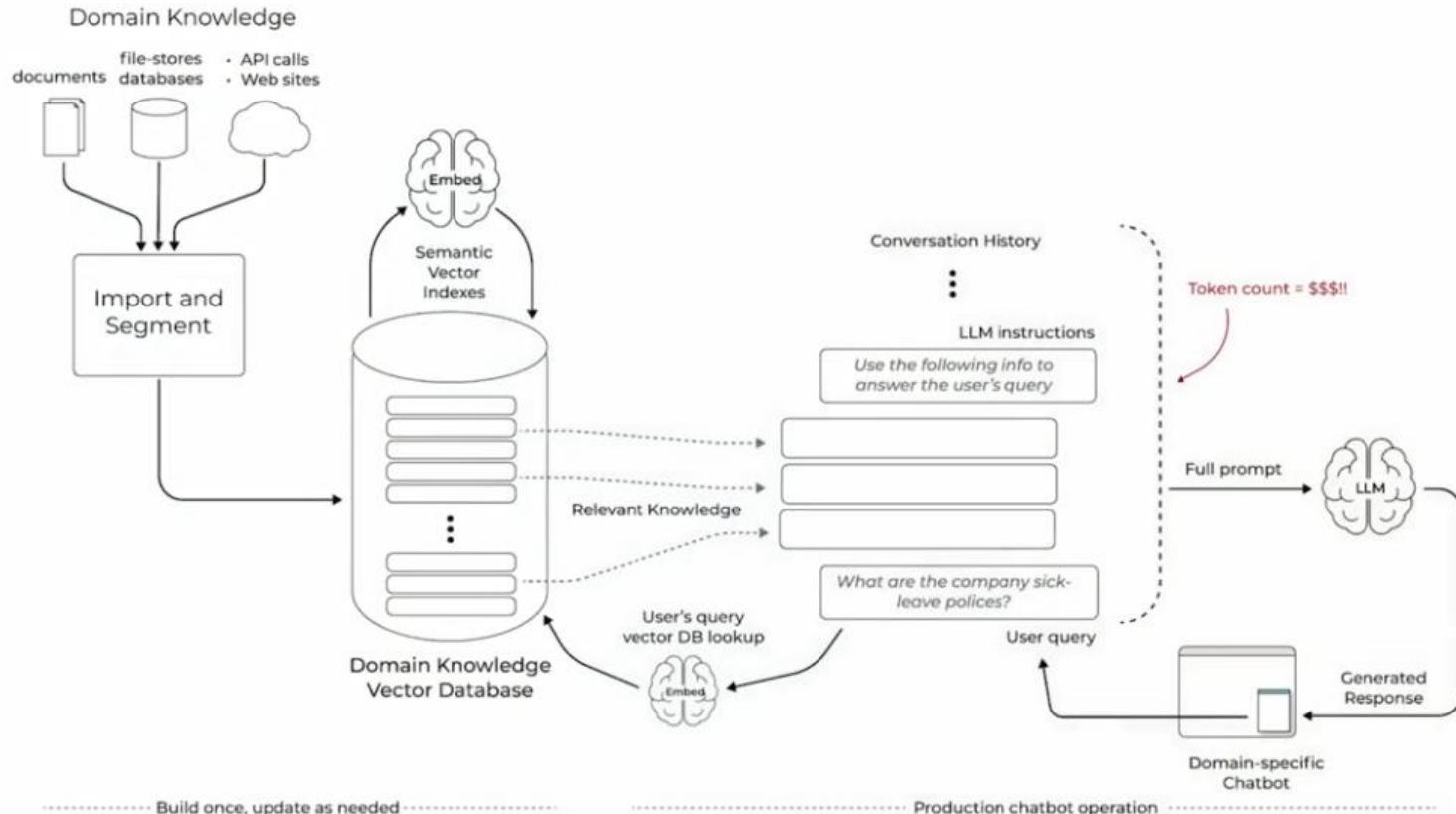


RAG 기반 지식 DB 및 챗봇 구축



RAG Basics

Using domain-specific knowledge with general-purpose LLMs



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow beta

HyperFlow AI / John's chatbot / edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

John's chatbot

Flow-graphs Nodes Templates

Search Examples My bots

Run

A screenshot of the HyperFlow AI web application. The interface includes a top navigation bar with links for Docs, Support, Report a Bug, account information, and Logout. Below this is a toolbar with buttons for Editor, Runner, Histories, Browser, and Simulator. On the left, there's a sidebar with sections for Flow-graphs, Nodes, Templates, Search, Examples, and My bots. The main area is a large, light-gray workspace with a small cursor icon in the center. A vertical toolbar on the far left contains icons for zooming in and out, and other editing tools.

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow beta

HyperFlow AI / John's chatbot / Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator English docs
한국어 문서

John's chatbot

Flow-graphs Nodes Templates

Search Examples My bots

React for

A screenshot of the HyperFlow AI platform. The interface includes a top navigation bar with links for Docs, Support, Report a Bug, account information, and Logout. Below this is a secondary navigation bar with Editor, Runner, Histories, Browser, and Simulator tabs, along with language options for English docs and 한국어 문서. The main workspace is titled "John's chatbot" and contains tabs for Flow-graphs, Nodes, and Templates. On the left side, there are sections for Search, Examples, and My bots. A vertical toolbar on the far left features icons for creating nodes (+), deleting (-), and zooming (zoom in/out). The central area is a large, empty white space intended for building flow graphs.

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta Documentation

...



HyperFlow Beta Documentation

- [Beta program welcome](#)
- [Release notes](#)
- [hyperflow-ai.com/hyperflow - the HyperFlow IDE](#)
- [HyperFlow-AI community support on Discord](#)
- [HyperFlow-AI community support on KakaoTalk](#)
- [HyperFlow-AI tutorial video channel on Youtube](#)

-
- [Nodes reference](#)
 - [Services reference](#)
 - [Data types](#)
 - [Custom AI App Control API Reference](#)



RAG 기반 지식 DB 및 챗봇 구축



The screenshot shows a web-based documentation interface for HyperFlow. At the top, there's a navigation bar with links like 'HyperFlow Beta Docu...', 'Nodes reference', and 'Made with 🎨'. Below the header is a large blue banner featuring the HyperFlow logo (a stylized infinity symbol) and the word 'HyperFlow'.

The main content area has a sidebar on the left containing a list of nodes with icons:

- Add Content
- Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- Data Range
- Data Set
- Data Transform
- Define Tool
- Import content
- Instructions
- Key-value Store
- Launch Parameters

The main content area includes sections like 'Nodes Overview' and 'Alphabetical node listing', along with detailed descriptions for each node type.

Nodes Overview

HyperFlow's capabilities reside largely in the nodes that can be used to construct flow-graphs. This reference guide provides detailed definitions of each of the available node types in HyperFlow.

Alphabetical node listing

Add content

Adds input data to the content store as an importSet
Category: Content
Internal name: addContent

Annotation

Add text or labeled boxes to annotate a flow-graph.
Category: Annotation
Internal name: annotation

Batch content

Batches up content sources for downstream nodes.
Category: Data flow
Internal name: batcher

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta Docu... / ... / Code Docs / Import content

Q ⌂ ⌂ Made with

Import content

- ☰ Menu
- >Add Content
- * Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- ↔ Data Range
- ↳ Data Set
- ↳ Data Transform
- ✍ Define Tool
- ↳ Import content
- ↳ Instructions
- 🔍 Key-value Store
- ↳ Launch Parameters
- ✍ LLM Tool Agent
- ⌚ Loop Controller
- ↳ Match & Branch
- ↳ Match & Message
- ↳ Match & Prompt Button
- 🕒 Match & Select
- ↳ Merge Inputs
- ✉ Message
- ✉ Message to Host
- ⌚ Nested Loop Data
- ✍ Prompt Buttons
- ⌚ Re-rank Search

Import Content Node

Content node group

Overview

The Import Content node enables importing content from various sources into the workflow. It supports multiple import services, currently including an advanced build-in importing service supporting file uploads, web content retrieval, and sourcing from API endpoints, providing flexible content ingestion capabilities.

There are also import service adapters from various LangChain modules. More adapters will be added over time, and the built-in importer service is undergoing constant enhancements.

Built-in Importer Service

This built-in HyperFlow importer service available in this node supports content import through a configurable interface that adapts based on the selected import source. It supports different content formats and provides options for handling various types of inputs, from local files to web content and API responses.

Configuration Guidelines

- Importer Service Selection
 - Choose built-in importer service
 - Refer to [importer service](#) if you pick from other available services
- Import Source Configuration
 - Upload file(s)
 - Web
 - API
- Source-Specific Parameters
 - File Upload Parameters
 - Web Import Parameters
 - API Parameters

Workflow Patterns

Table of Contents

Import Content Node

Overview

Built-in Importer Service

Configuration Guidelines

Workflow Patterns

When to use

Best Practices

Key Features

Parameters

Common Parameters

File Upload Parameters

Web Import Parameters

API Import Parameters

Wireable Inputs

Wireable Outputs

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta Docu... / Custom AI App Control API Referenc...

Made with

Here is all the information you need to interact with your newly published flow-graphs in any custom chatbots or other AI apps you might build. The HyperFlow Control API is a classic HTTP RESTful API, the end-point URLs for which are available in the Published Flow-graph Manager section of the Project Manager page.

HyperFlow AI also provides ready-made interactive chatbots that you can embed in your host applications or web-sites, and these chatbots also use this Control API. See the [Embed code](#) for these in the published flow-graph manager.

Finally, there is a **Chatbot Simulator**, available via the Simulator mode button in the HyperFlow IDE that can be attached to a ready-to-publish flow-graph for testing that shows a detailed log of the API calls between flow-graph and AI app, which can be helpful in understanding the Control API in use.

Introduction
Getting Started
 Publishing your production-ready flow-graph
 Control API Sequencing
 General call forms
 Understanding API responses
 Data field
Responses
 Start
 Message
 Interaction
 Image
 End
 Generated Text
References
 Busy Message
Interaction Specs types
API Reference
 Start a session
 Key Features
 When to use
 Best Practices
 Headers
 Body
 Response

Introduction

The Control API provides comprehensive access to a published flow-graphs functionality, allowing it to drive and interact with custom chatbots, AI assistants and other AI applications you might choose to develop to provide custom user-interfaces to your HyperFlow apps.

This documentation covers the available API calls and their returned properties. If you have any questions, please reach visit the HyperFlow Discord support server or email team@hyperflow-ai.com.

Getting Started

Publishing your production-ready flow-graph

Once you are satisfied with your flow-graph's development and wish to use it through either one of HyperFlow's built-in embeddable chatbots, or a chatbot or other custom UI, you will need to publish it using the Publish button in the main HyperFlow editor UI. Only published flow-graphs can be controlled and interacted with via the Control API documented here.

Control API Sequencing

The Control API call sequencing is relatively simple, just two main calls, but the details within each require careful attention, as these embody all the state exchanges & UI interactions between the app's user and the running flow-graph.

The main call sequence comprises:

1. Calling the **Control Start** call ([/api/flowgraph/control/start](#)) that selects and launches a published flow-graph, and,
2. Repeatedly making the **Control Progress** call ([/api/flowgraph/control/progress](#)) that passes up any input or other interaction from the user and retrieves any available output from the flow-graph along with any further requests for user interaction it makes.

The **Control Proceed** call is effectively a long-poll call, waiting for the flow-graph executing within the HyperFlow flow-graph engine on the server to generate output or request interaction, and so the client-side app in all cases except those awaiting input from the user, will immediately make another Control Progress call to wait for any newly-available output or requests from the flow-graph and deliver it to the user in a timely manner.



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

John's chatbot

Flow-graphs Nodes Templates

Search Examples My bots

React Flow

RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow beta

HyperFlow AI / John's chatbot / Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

John's chatbot

Flow-graphs Nodes Templates

Search

Chatbots Knowledge DBs

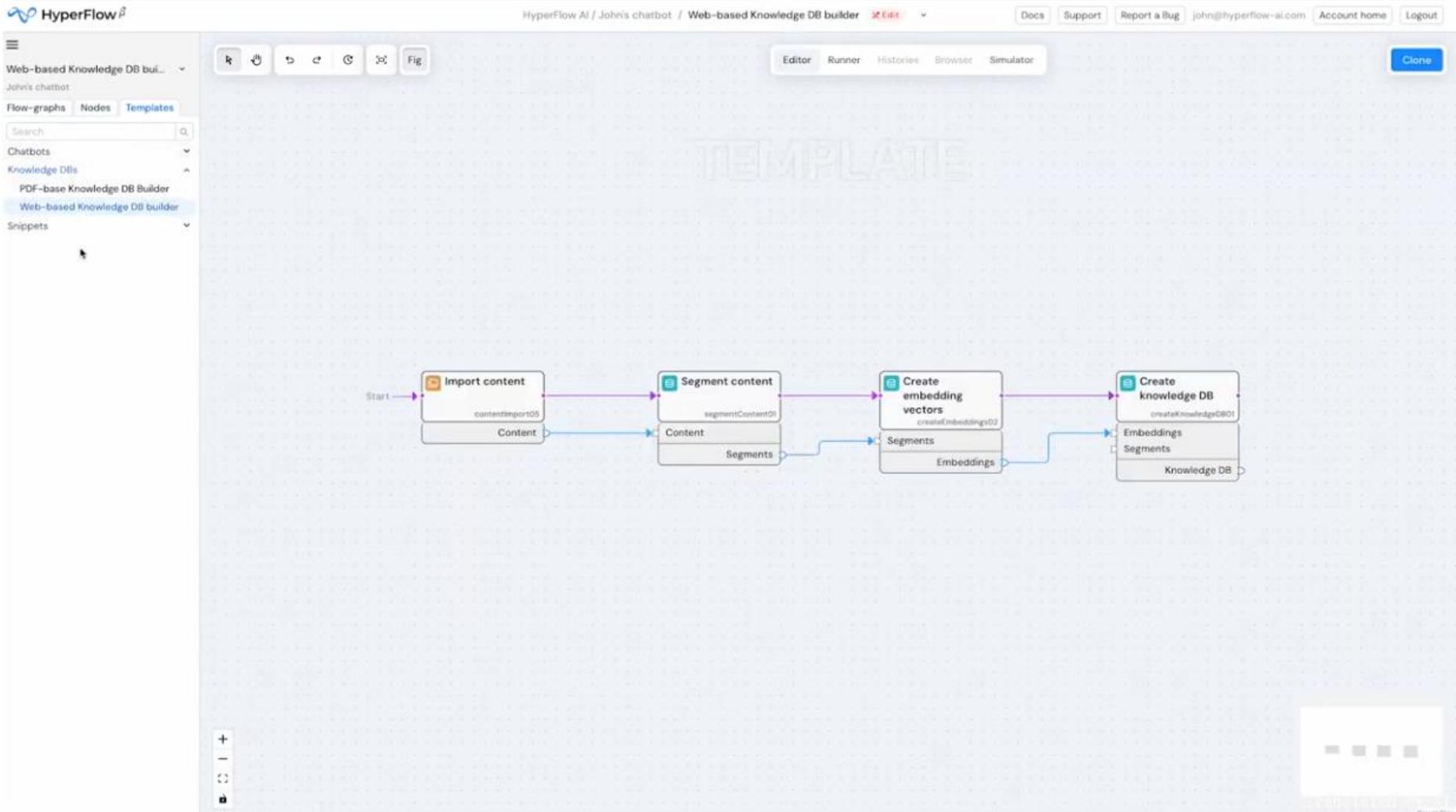
PDF-base Knowledge DB Builder Web-based Knowledge DB builder

Snippets

React Flow

A screenshot of the HyperFlow AI web application. The interface includes a header with the logo, version information, and navigation links. On the left, there's a sidebar with sections for Chatbots, Knowledge DBs, and Snippets, along with search and filter tools. The main area is a large, light-gray workspace with a grid pattern, intended for creating flow graphs. A vertical toolbar on the far left contains icons for adding nodes and connecting them. A small watermark 'React Flow' is visible in the bottom right corner of the workspace.

RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper HyperFlow Helper Saved

Import content #45 contentimport01

All locks

Importer service Built-in Importer

Import source Web

Tags

Notes

Page or site URL

Content form Heading-separated text

Dynamic HTML site

Enabled

Page loaded condition documentloaded

Delay after load (secs)

Web import method Site crawl

Depth

Maximum pages maximum

Subtree only

Enabled

Click-crawl CSS selectors

URL filter pattern

CSS content filter

Start → Import content (contentimport01) → Segment content (segmentContent01) → Create embedding vectors (createEmbeddings02) → Embedding (embedding01)

Import content (contentimport01) node details:

- Tags
- Page or site URL
- Depth
- Content

Segment content (segmentContent01) node details:

- Content
- Segments

Create embedding vectors (createEmbeddings02) node details:

- Segments
- Embeddings

Embedding (embedding01) node details:

- Emb
- Seg

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Create[Create embedding vectors]; Create --> Embedding[Embedding];
```

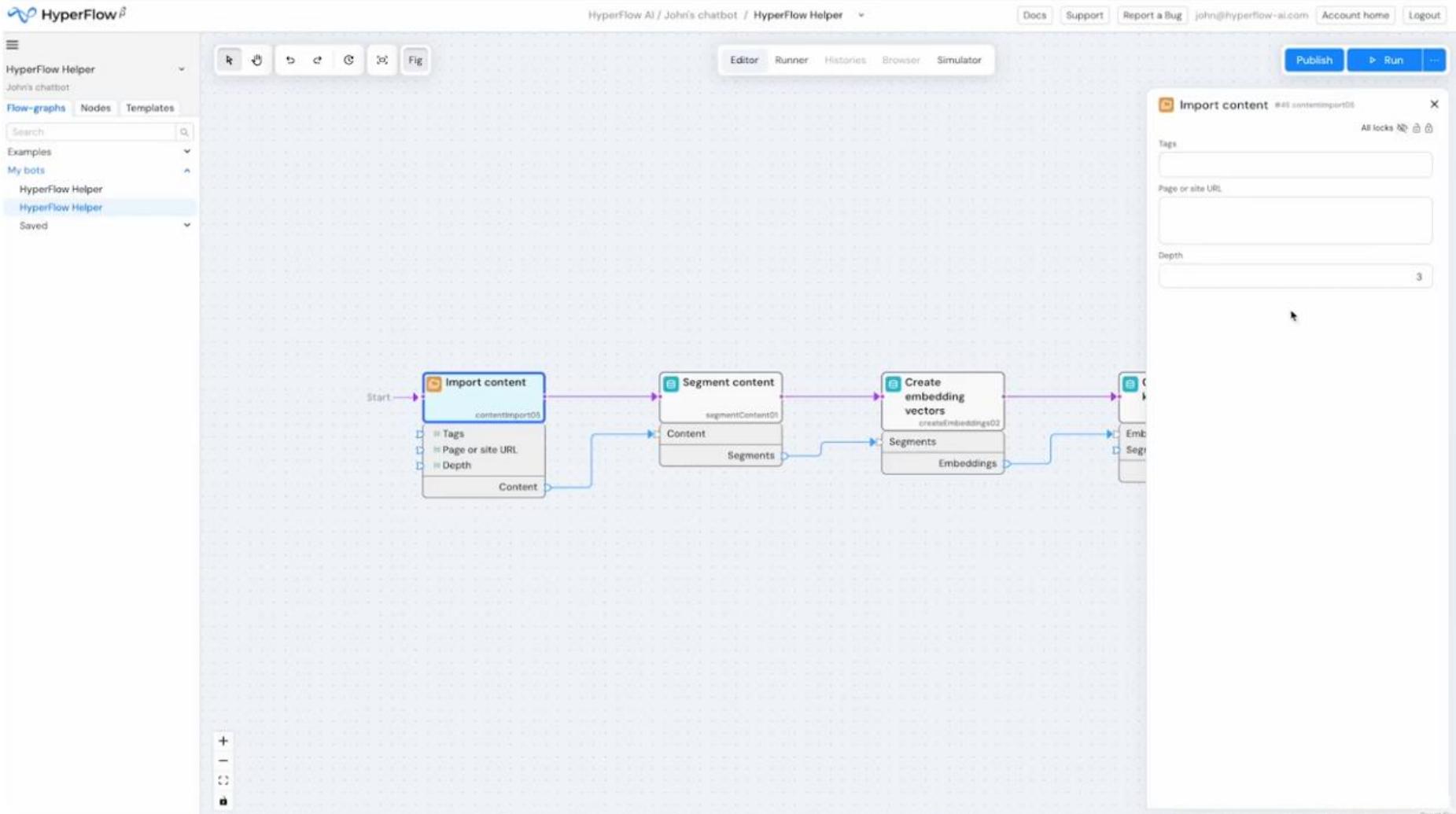
contentimport01

segmentContent01

createEmbeddings02

embedding01

RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow Helper / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content #45 contentimport05

All locks

Importer service: Built-in Importer

Import source: Web

Tags:

Notes:

Page or site URL:

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (secs): 3

Web import method: Site crawl

Depth:

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:

Restart Flow

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper

HyperFlow Helper

Saved

Start → Import content (contentimport05) → Segment content (segmentContent01) → Create embedding vectors (createEmbeddings02) → Embedding (embedding03)

Segment content (segmentContent01) receives Content and Segments from Import content (contentimport05), and sends Content and Segments to Create embedding vectors (createEmbeddings02).

Create embedding vectors (createEmbeddings02) receives Content and Segments from Segment content (segmentContent01), and sends Embeddings to Embedding (embedding03).

Embedding (embedding03) receives Embeddings from Create embedding vectors (createEmbeddings02).

The Import content (contentimport05) node has a detailed configuration panel on the right:

- Importer service: Built-in Importer
- Import source: Web
- Tags:
- Notes:
- Page or site URL:
- Content form: Heading-separated text
- Dynamic HTML site
- Page loaded condition: DOM content loaded
- Delay after load (secs): 3
- Web import method: Site crawl
- Depth:
- Maximum pages: maximum
- Subtree only
- Click-crawl CSS selectors:

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content #45 contentimport00

Importer service: Built-in importer

Built-in importer: Langchain importer

Clear selection

Notes:

Page or site URL:

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (secs): 3

Web import method: Site crawl

Depth:

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:

React for:

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Embed[Create embedding vectors]; Embed --> End((End)); Segment -- Content --> Segment; Segment -- Segments --> Embed;
```

Import content node details:

- Importer service: Built-in importer
- Import source:
- Tags:
- Notes:
- Page or site URL:
- Content form:
- Dynamic HTML site:
- Page loaded condition:
- Delay after load (secs):
- Web import method:
- Site crawl:
- Depth:
- Maximum pages:
- Subtree only:
- Click-crawl CSS selectors:

Segment content node details:

- Content:
- Segments:

Create embedding vectors node details:

- Segments:
- Embeddings:

RAG 기반 지식 DB 및 챗봇 구축



The screenshot shows the HyperFlow AI interface with a workflow titled "Import content". The workflow consists of four main nodes connected sequentially:

- Import content**: This node is set to "Built-in importer" and "Web" as the import source. It has a "Content" output port.
- Segment content**: This node takes "Content" as input and produces "Segments".
- Create embedding vectors**: This node takes "Segments" as input and produces "Embeddings".
- Export content**: This node takes "Embeddings" as input and has an "Embed" output port.

A "Start" node is connected to the first "Import content" node. A feedback loop from the "Embed" port of the final node goes back to the "Content" port of the first "Import content" node. The right side of the interface features a detailed configuration panel for the "Import content" node, showing settings for importer service, import source (Web), API selection, and various crawl options like "Dynamic HTML site" and "Subtree only".

RAG 기반 지식 DB 및 챗봇 구축



The screenshot shows the HyperFlow Helper interface with a flow graph and a detailed configuration panel.

Flow Graph:

- Start** node connected to **Import content** node.
- Import content** node has a dropdown menu open, showing options like Importer service, Import source, Tags, Notes, Page or site URL, Content form, Dynamic HTML site, Page loaded condition, Delay after load (secs), Web import method, Depth, Maximum pages, Subtree only, Click-crawl CSS selector, URL filter pattern, and CSS content filter.
- Import content** node connects to **Segment content** node.
- Segment content** node has two outputs:
 - A blue arrow points to the **Content** input of the **Create embedding vectors** node.
 - A purple arrow points to the **Segments** output of the **Create embedding vectors** node.
- Create embedding vectors** node has two outputs:
 - A purple arrow points to the **Embeddings** output of the **Segment content** node.
 - A blue arrow points to the **Embeddings** input of the **Segment content** node.
- Segment content** node also has a **Segments** output, which loops back to its own **Content** input.

Configuration Panel (Import content node):

- Importer service:** Built-in importer
- Import source:** Web
- Tags:** hyperflow-helper
- Notes:**
- Page or site URL:**
- Content form:** Heading-separated text
- Dynamic HTML site:**
- Page loaded condition:** DOM content loaded
- Delay after load (secs):** 3
- Web import method:** Site crawl
- Depth:** 3
- Maximum pages:** maximum
- Subtree only:**
- Click-crawl CSS selectors:**

RAG 기반 지식 DB 및 챗봇 구축



The screenshot shows the HyperFlow AI interface with the following details:

- Header:** HyperFlow AI / John's chatbot / HyperFlow Helper
- Top Bar:** Docs, Support, Report a Bug, john@hyperflow-ai.com, Account home, Logout
- Left Sidebar:** HyperFlow Helper (selected), Flow-graphs, Nodes, Templates, Search, Examples, My bots, HyperFlow Helper, HyperFlow Helper (highlighted), Saved.
- Toolbar:** Back, Forward, Home, Refresh, Fig (highlighted).
- Main Area:** A flow graph titled "Import content". The graph consists of four main nodes connected sequentially:
 - Import content** (contentImport01) - Contains configuration for Importer service (Built-in importer), Import source (Web), Tags, Notes, Page or site URL (<https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb78040886df2878b90336f>), Content form (Heading-separate text), Dynamic HTML site (checked), Page loaded condition (DOM content loaded), Delay after load (secs: 3), Web import method (Site crawl), Depth, Maximum pages, Subtree only (checked), Click-crawl CSS selectors, URL filter pattern, and CSS content filter.
 - Segment content** (segmentContent01) - Contains Content and Segments.
 - Create embedding vectors** (createEmbeddings01) - Contains Segments and Embeddings.
 - End node** (highlighted)
- Right Panel:** Import content configuration panel with fields for Importer service (Built-in importer), Import source (Web), Tags, Notes, Page or site URL, Content form (Heading-separate text), Dynamic HTML site (checked), Page loaded condition (DOM content loaded), Delay after load (secs: 3), Web import method (Site crawl), Depth, Maximum pages, Subtree only (checked), Click-crawl CSS selectors, and URL filter pattern.

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper HyperFlow Helper

Saved

Import content #45-contentimport01

Importer service: Built-in importer

Import source: Web

Tags: hyperflow-helper

Notes:

Page or site URL: <https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb78040886df2878b90336f>

Content form: Heading-separated text

Heading-separated text

Original HTML

Plain text

Clear selection

Delay after load (secs): 3

Web import method: Site crawl

Depth:

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:

Start

Import content

Segment content

Create embedding vectors

Embedding

Content

Segments

Segments

Embeddings

Content

```
graph LR; Start((Start)) --> ImportContent[Import content]; ImportContent --> SegmentContent[Segment content]; SegmentContent --> CreateEmbedding[Create embedding vectors]; CreateEmbedding --> Embedding[Embedding]; Content[Content] --> SegmentContent; Content --> CreateEmbedding;
```

Import content

Segment content

Create embedding vectors

Embedding

Content

Segments

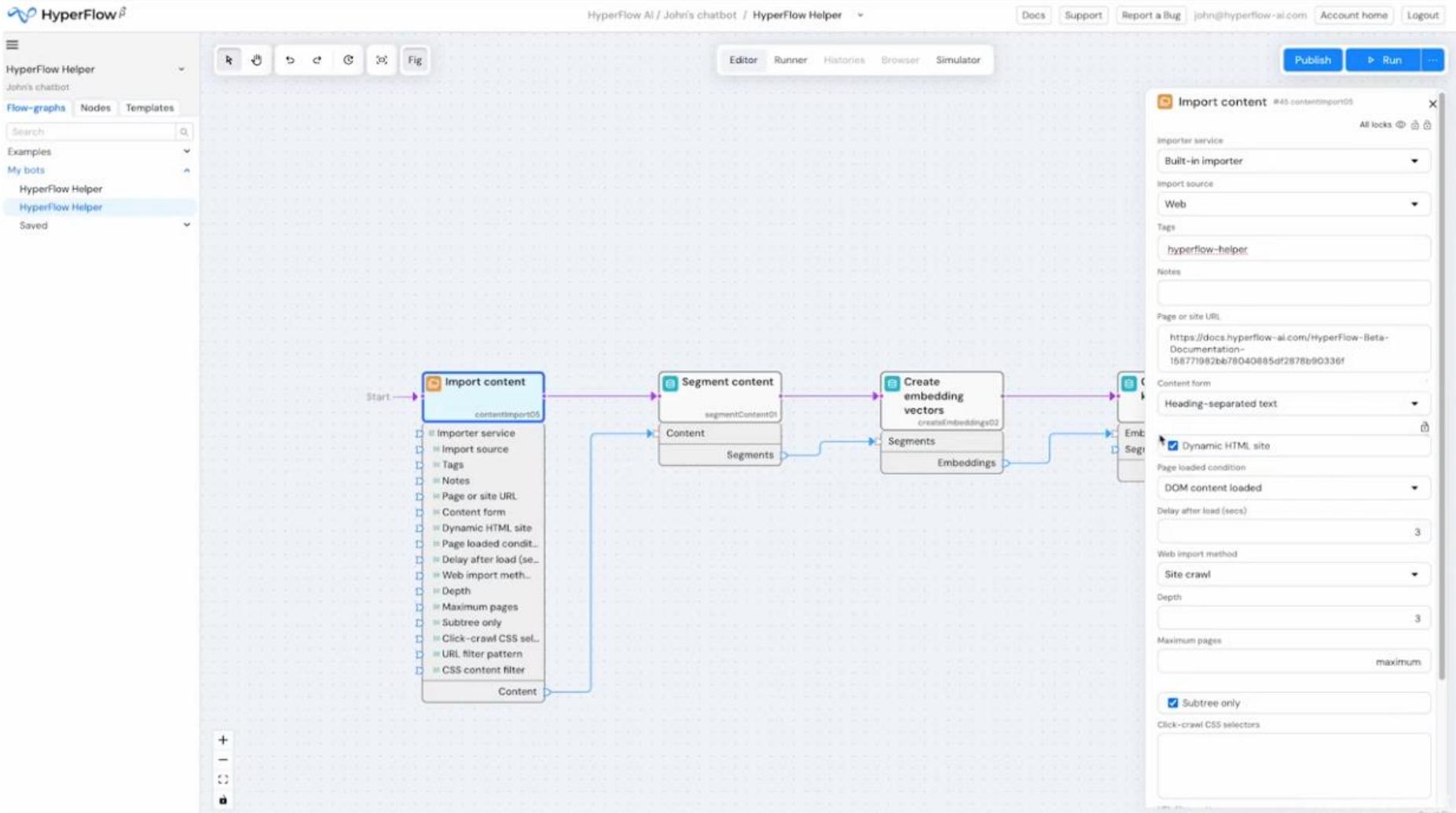
Segments

Embeddings

Content

Content

RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content #40 contentimport05

All locks

Importer service: Built-in importer

Import source: Web

Tags: hyperflow-helper

Notes:

Page or site URL: https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb7040885df2ff78b90330f

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (sec): 4

Web import method: Site crawl

Subtree only

Click-crawl CSS selectors

Start

Import content (contentimport05)

Segment content (segmentContent01)

Create embedding vectors (createEmbeddings02)

Embedding (embedding03)

Content

Segments

Embeddings

Content

Diagram description: A flow graph showing the process of importing content. It starts with an 'Import content' node (contentimport05), which has a dropdown menu with options like 'Importer service', 'Import source', 'Tags', etc. An arrow points from this node to a 'Segment content' node (segmentContent01). This node has two outputs: 'Content' and 'Segments'. The 'Content' output goes to a 'Create embedding vectors' node (createEmbeddings02), which then produces 'Embeddings'. Finally, these embeddings are processed by an 'Embedding' node (embedding03).

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content #45-contentimport05

Importer service: Built-in importer

Import source: Web

Tags: hyperflow-helper

Notes:

Page or site URL: https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb78040885df2ff78b90330f

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (secs): 4

Web import method: Site crawl

Depth: 4

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:

Ready for...

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper

HyperFlow Helper

Saved

Import content

Segment content

Create embedding vectors

Embedding

Content

Segments

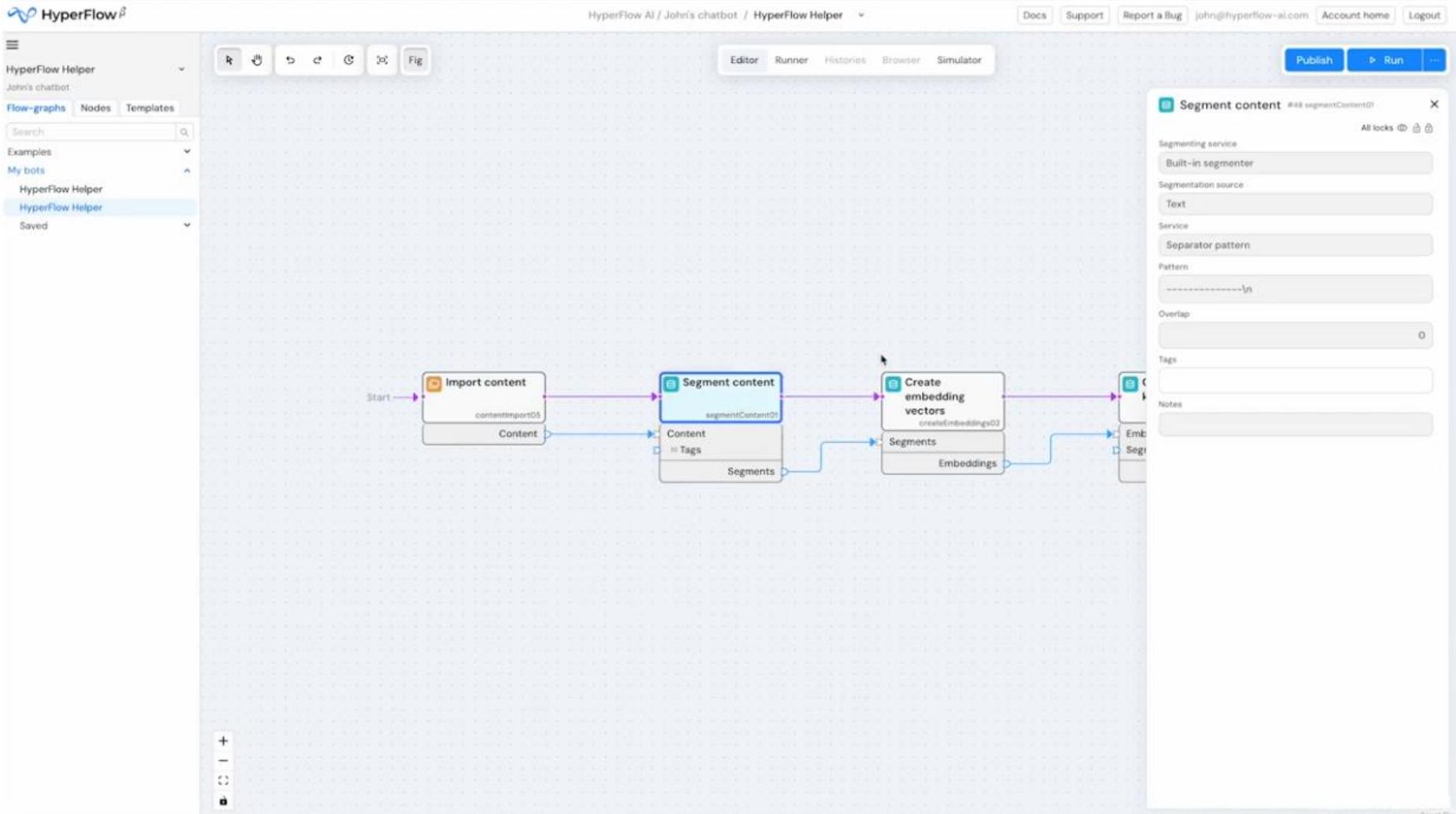
Segments

Embeddings

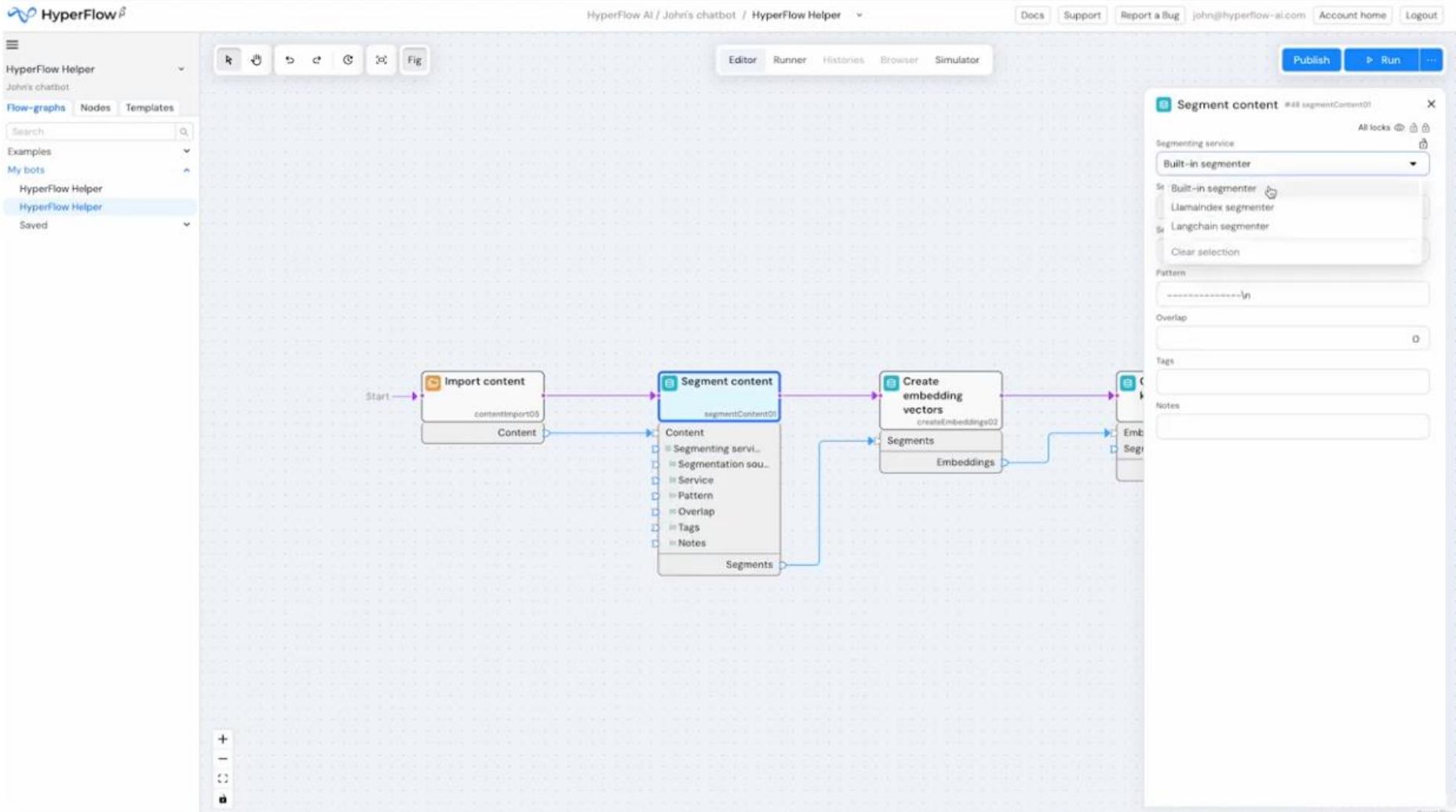
```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Create[Create embedding vectors]; Create --> Embedding[Embedding]; Content[Content] --> Segment; Content --> Create; Segments[Segments] --> Create; Segments --> Embedding; Embedding --> Embedding;
```

The screenshot shows the HyperFlow interface for building a RAG-based knowledge database and chatbot. The main area displays a flow graph with four main nodes: 'Import content', 'Segment content', 'Create embedding vectors', and 'Embedding'. The 'Import content' node is set up to import from a 'Web' source using a 'Built-in importer'. It has configuration options for 'Content form' (Heading-separated text), 'Dynamic HTML site', 'Page loaded condition' (DOM content loaded), 'Delay after load (secs)' (4), 'Web import method' (Site crawl), 'Depth' (4), 'Maximum pages' (maximum), and 'Subtree only'. The 'Segment content' node takes 'Content' and 'Segments' as inputs. The 'Create embedding vectors' node takes 'Segments' and 'Embeddings' as inputs. The final 'Embedding' node is shown. On the left, there is a detailed configuration panel for the 'Import content' node, listing various import settings like 'Importer service', 'Import source', 'Tags', 'Page or site URL', 'Content form', 'Dynamic HTML site', 'Page loaded condition', 'Delay after load (secs)', 'Web import method', 'Depth', 'Maximum pages', and 'Subtree only'. A 'Click-crawl CSS selectors' field is also present.

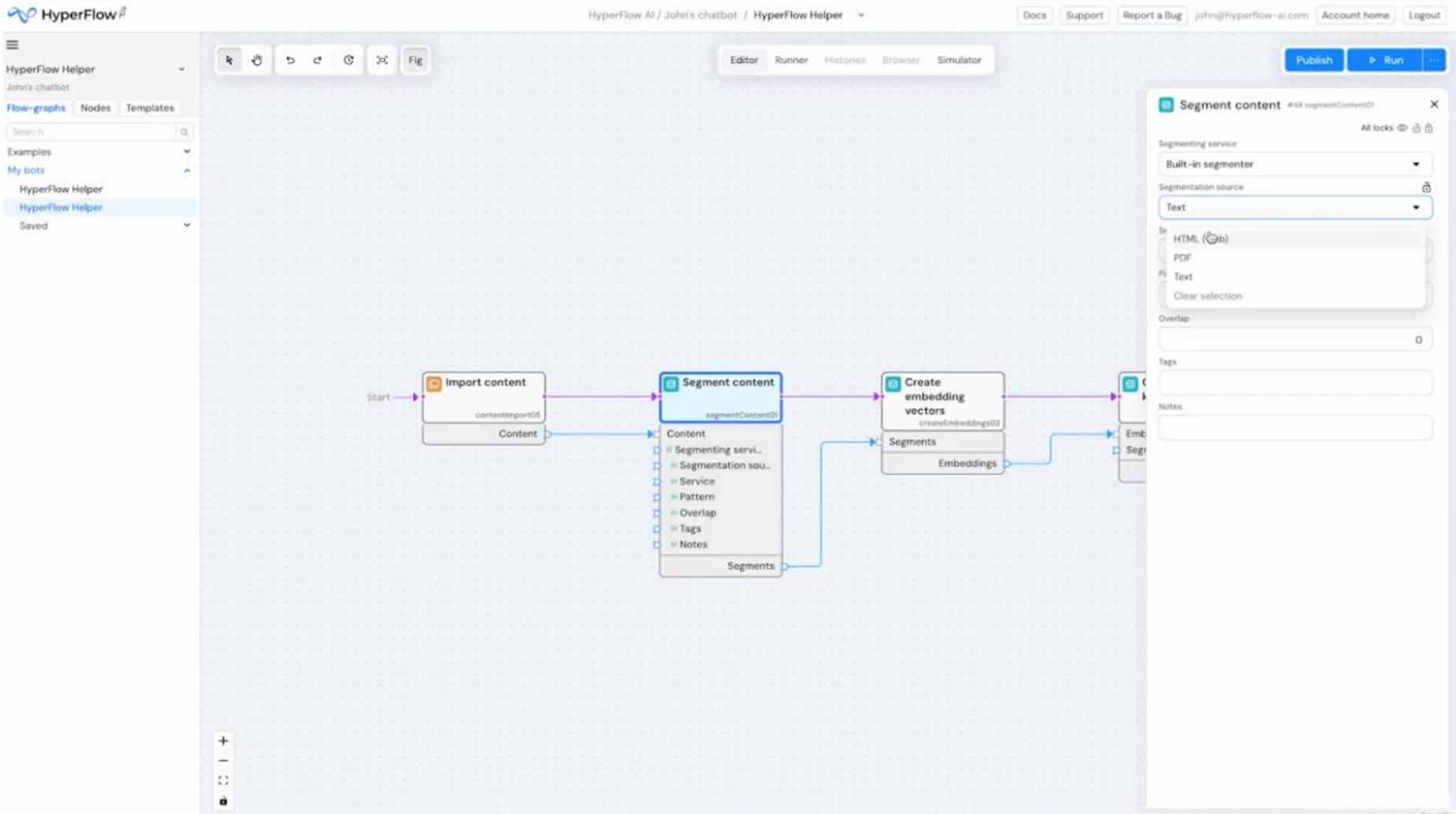
RAG 기반 지식 DB 및 챗봇 구축



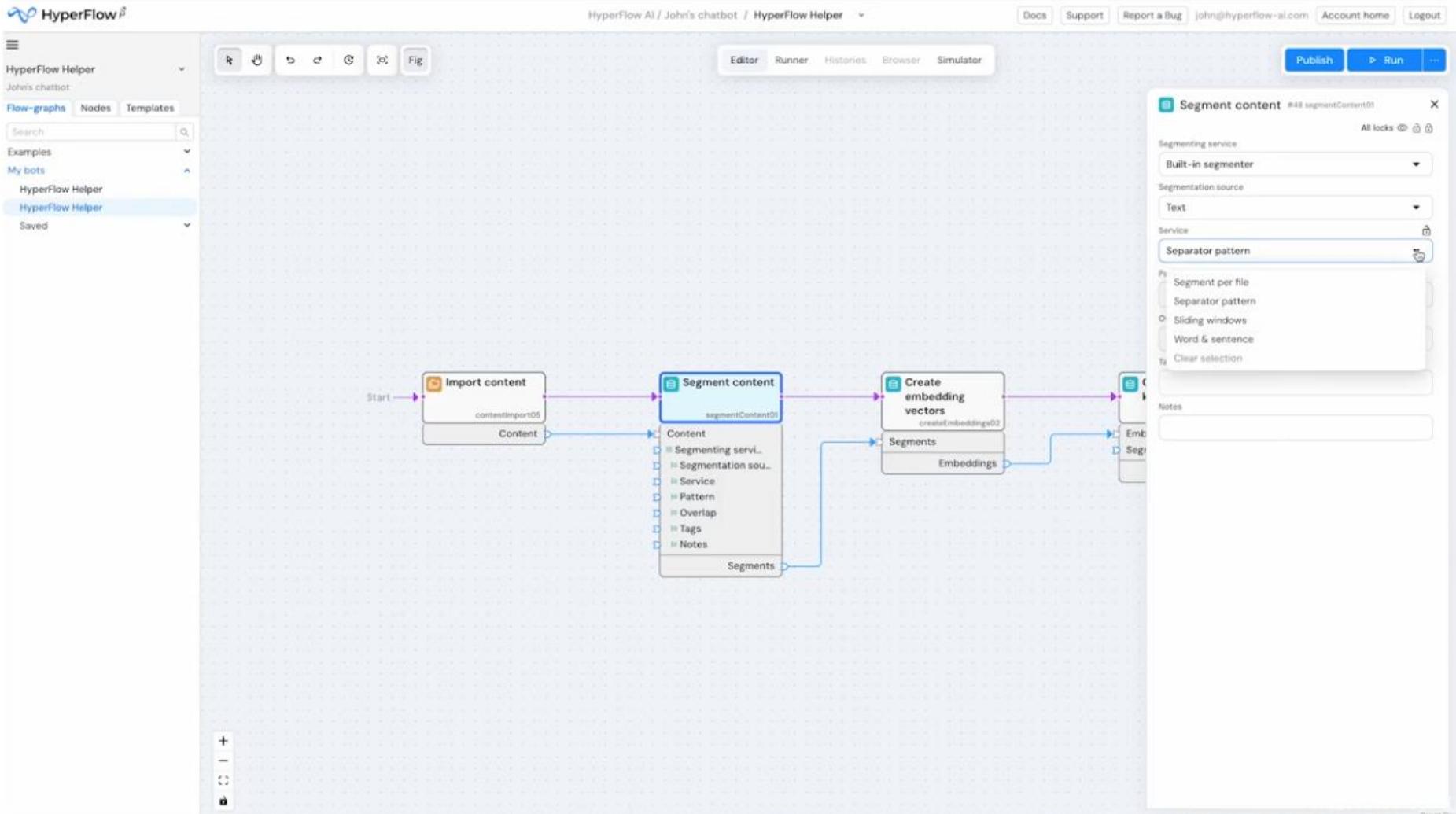
RAG 기반 지식 DB 및 챗봇 구축



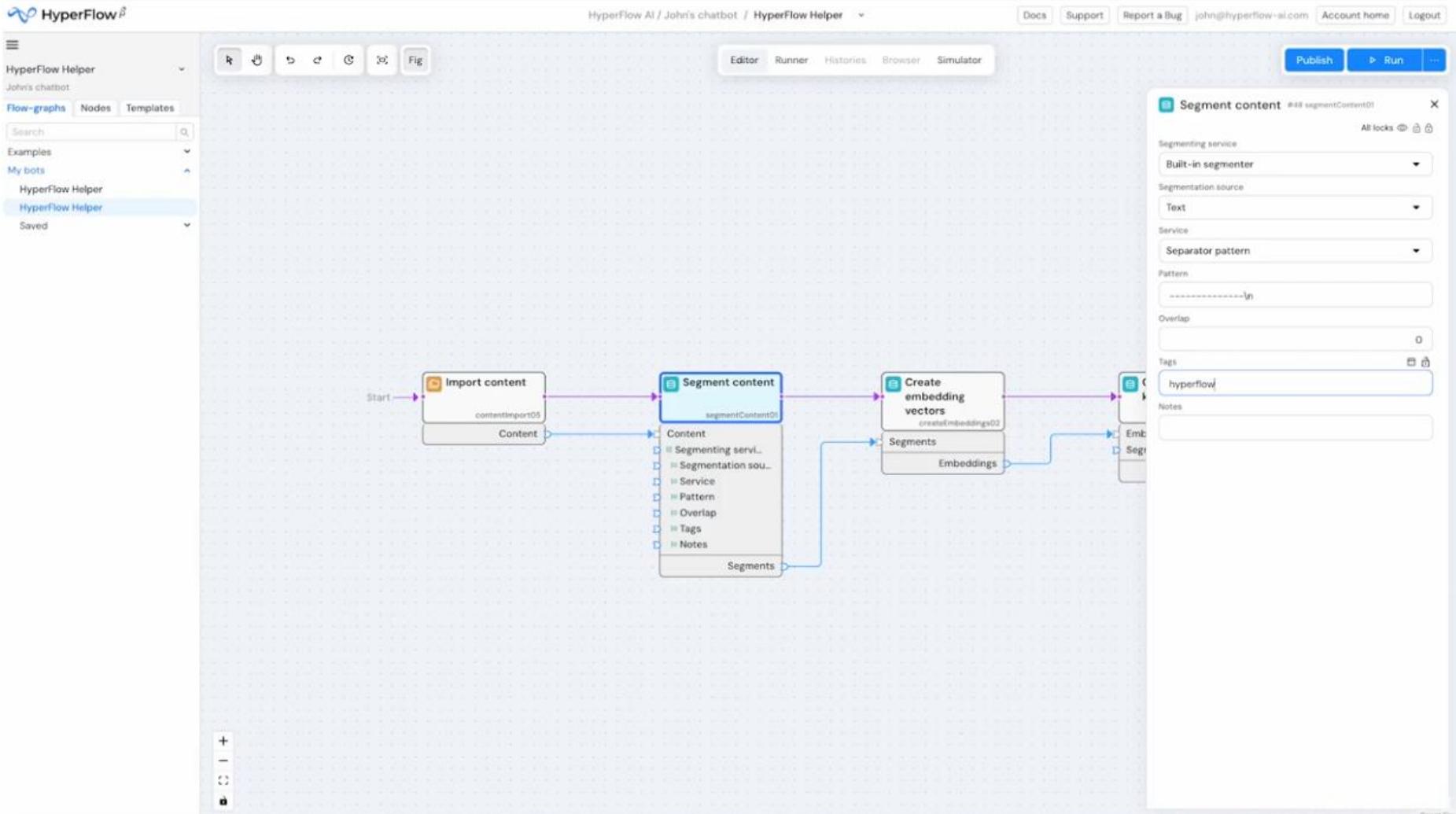
RAG 기반 지식 DB 및 챗봇 구축



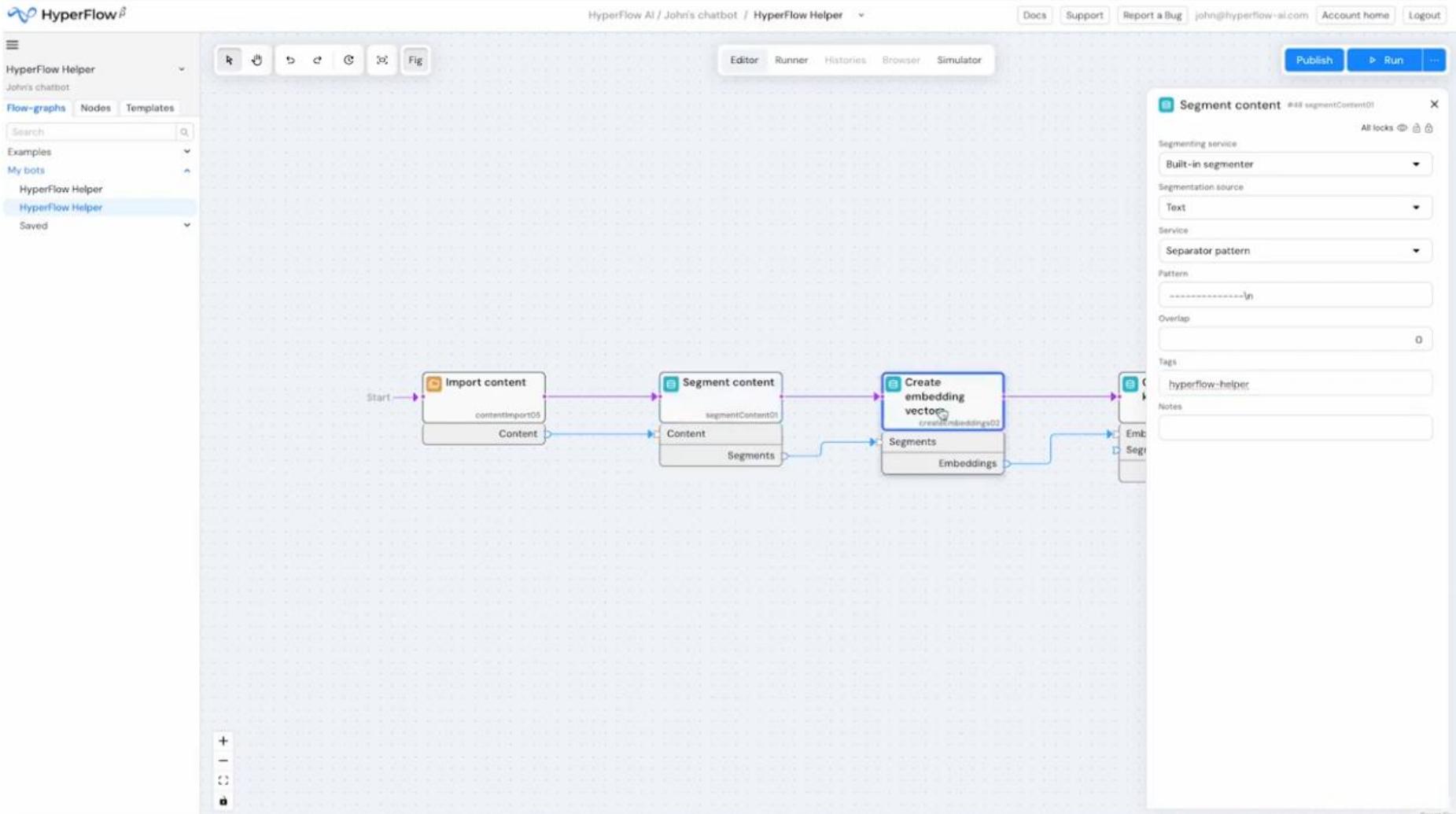
RAG 기반 지식 DB 및 챗봇 구축



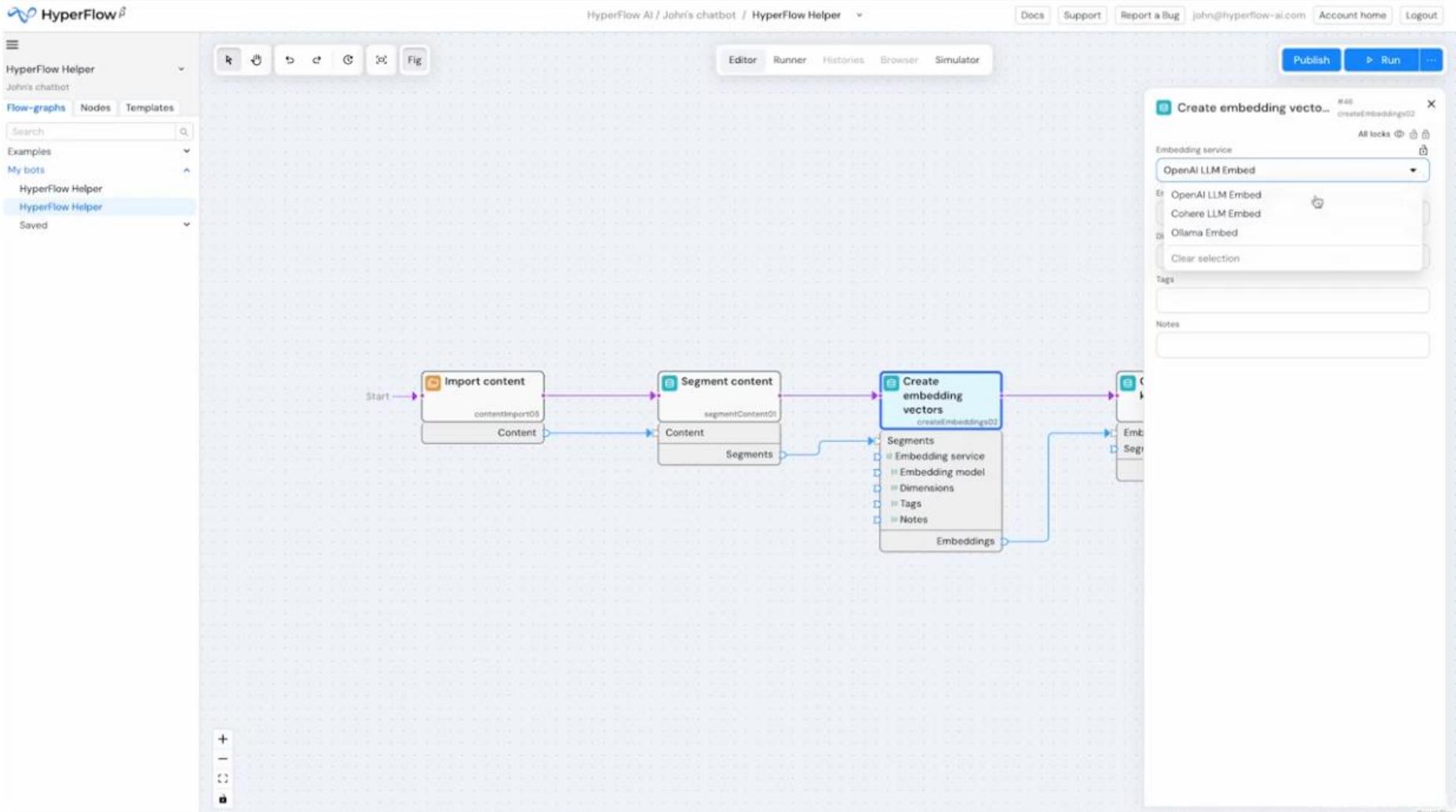
RAG 기반 지식 DB 및 챗봇 구축



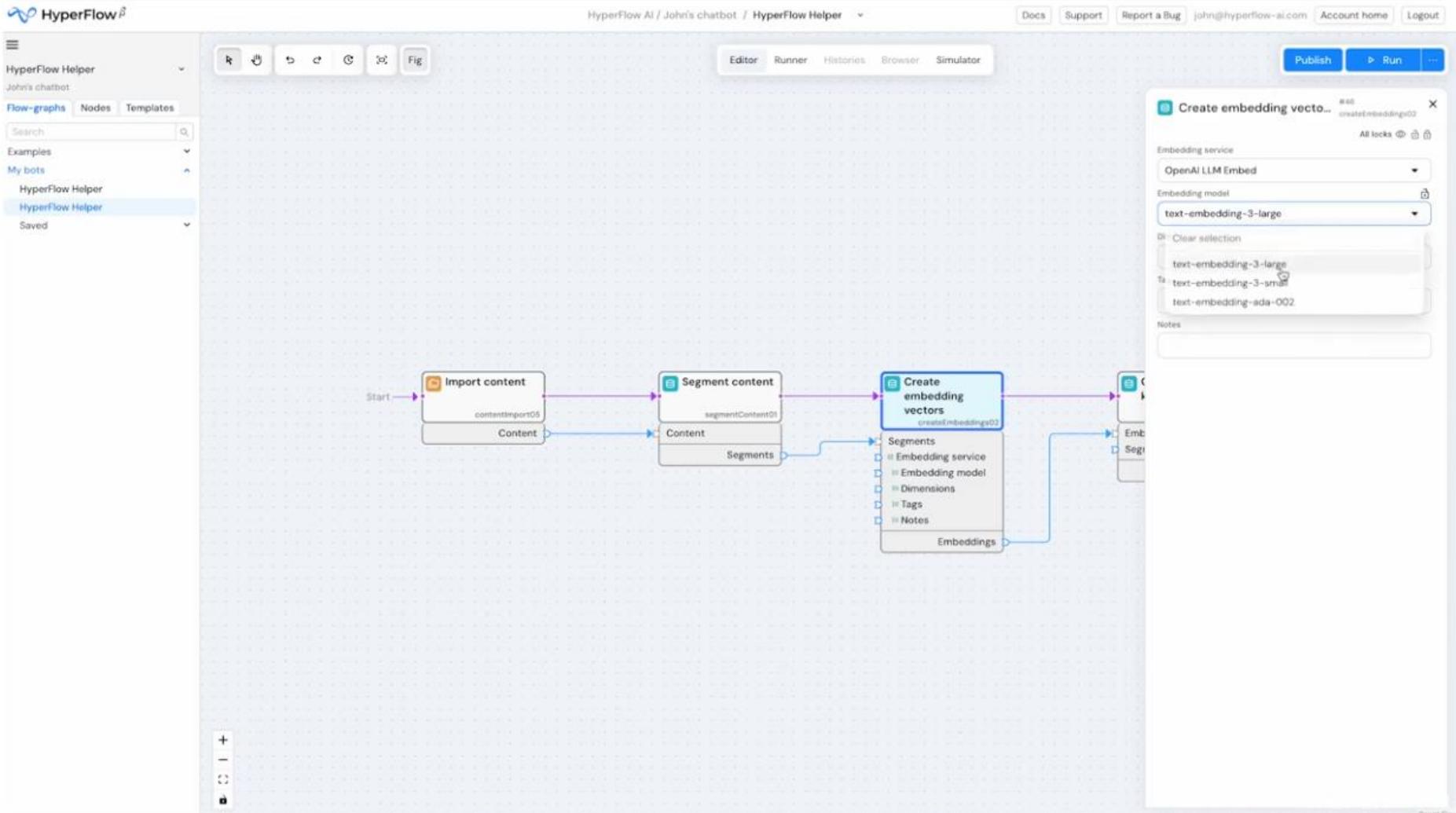
RAG 기반 지식 DB 및 챗봇 구축



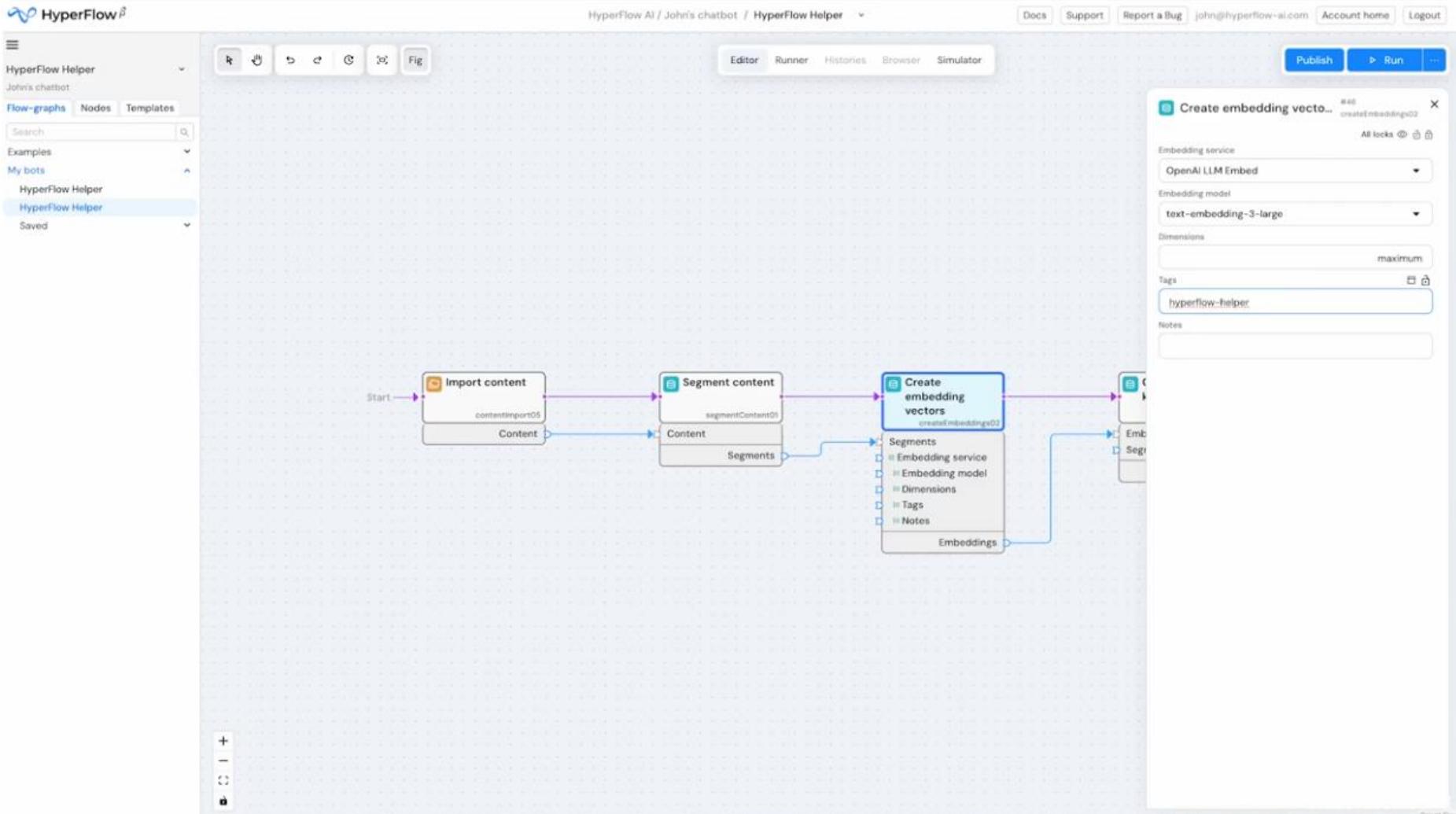
RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow 

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

HyperFlow Helper

John's chatbot

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper

HyperFlow Helper

Saved

Import content

Segment content

Create embedding vectors

Create knowledge DB

contentImport01

segmentContent01

createEmbedding02

createKnowledgeDB01

Content

Segments

Segments

Embeddings

Embeddings

Segments

Number of index c...

Tags

Knowledge DB

1

contentImport01

segmentContent01

createEmbedding02

createKnowledgeDB01

Content

Segments

Segments

Embeddings

Embeddings

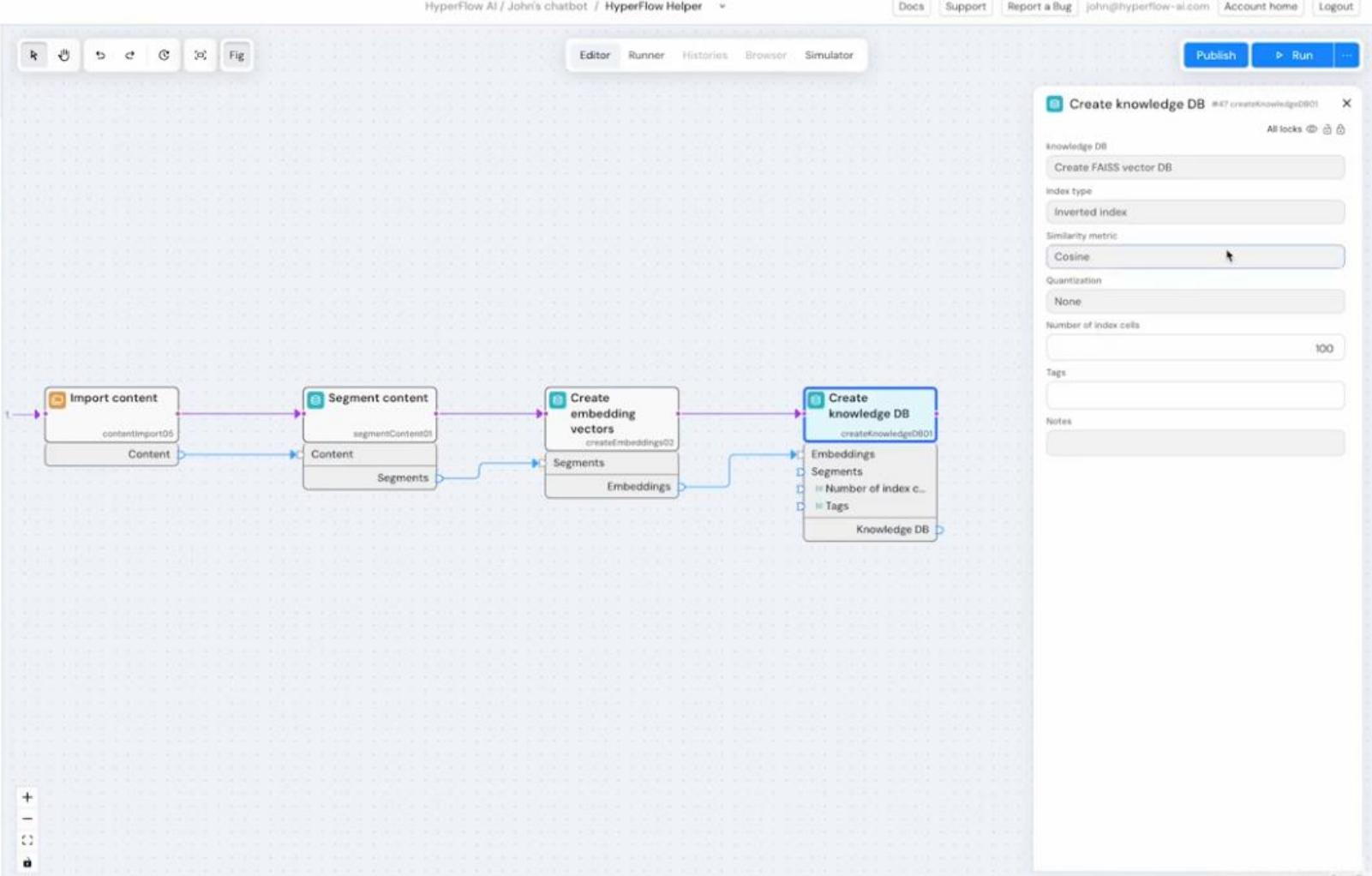
Segments

Number of index c...

Tags

Knowledge DB

React File



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper HyperFlow Helper Saved

Import content → Segment content → Create embedding vectors → Create knowledge DB

Content → Segments → Embeddings → Knowledge DB

Create knowledge DB

Create FAISS vector DB

Create Pinecone DB

Create ElasticSearch DB

Create MongoDB Vector DB

Clear selection

None

Number of index cells: 100

Tags:

Notes:

```
graph LR; A[Import content] --> B[Segment content]; B --> C[Create embedding vectors]; C --> D[Create knowledge DB]; A -- Content --> B; B -- Segments --> C; C -- Embeddings --> D; D -- Knowledge DB --> E[Knowledge DB]
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper HyperFlow Helper Saved

Import content → Segment content → Create embedding vectors → Create knowledge DB

Content → Segments → Embeddings

Create knowledge DB (createKnowledgeDB01)

knowledge DB

- Create FAISS vector DB
- Index type: Inverted index
- Similarity metric: Cosine
- Quantization: None
- Number of index cells: 100
- Tags
- Notes

Knowledge DB

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates

Search Examples My bots

HyperFlow Helper HyperFlow Helper

Saved

Import content → Segment content → Create embedding vectors → Create knowledge DB

Content → Segments → Embeddings → Knowledge DB

Create knowledge DB

knowledge DB

Create FAISS vector DB

index type

Inverted index

Similarity metric

Cosine

Quantization

None

Number of index cells

100

Tags

hyperflow-helper

Notes

```
graph LR; Import[Import content] --> Segment[Segment content]; Segment --> Embedding[Create embedding vectors]; Embedding --> KnowledgeDB[Create knowledge DB];
```

Knowledge DB

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Import content #45 contentImport05 Step 1

All locks 0

Importer service: Built-in importer

Import source: Web

Tags: hyperflow-helper

Notes:

Page or site URL: <https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb78040885df2878b90338f>

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (secs): 4

Web import method: Site crawl

Depth: 4

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:

React Flow

Translating in AI

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper ✗ Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

Import content #45 contentImport06 Step 1

All locks ✗ ✓ ✗

Importer service: Built-in importer

Import source: Web

Tags: hyperflow-helper

Notes:

Page or site URL: <https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation-158771982bb78040885df2878b90338f>

Content form: Heading-separated text

Dynamic HTML site

Page loaded condition: DOM content loaded

Delay after load (secs): 4

Web import method: Site crawl

Depth: 4

Maximum pages: maximum

Subtree only

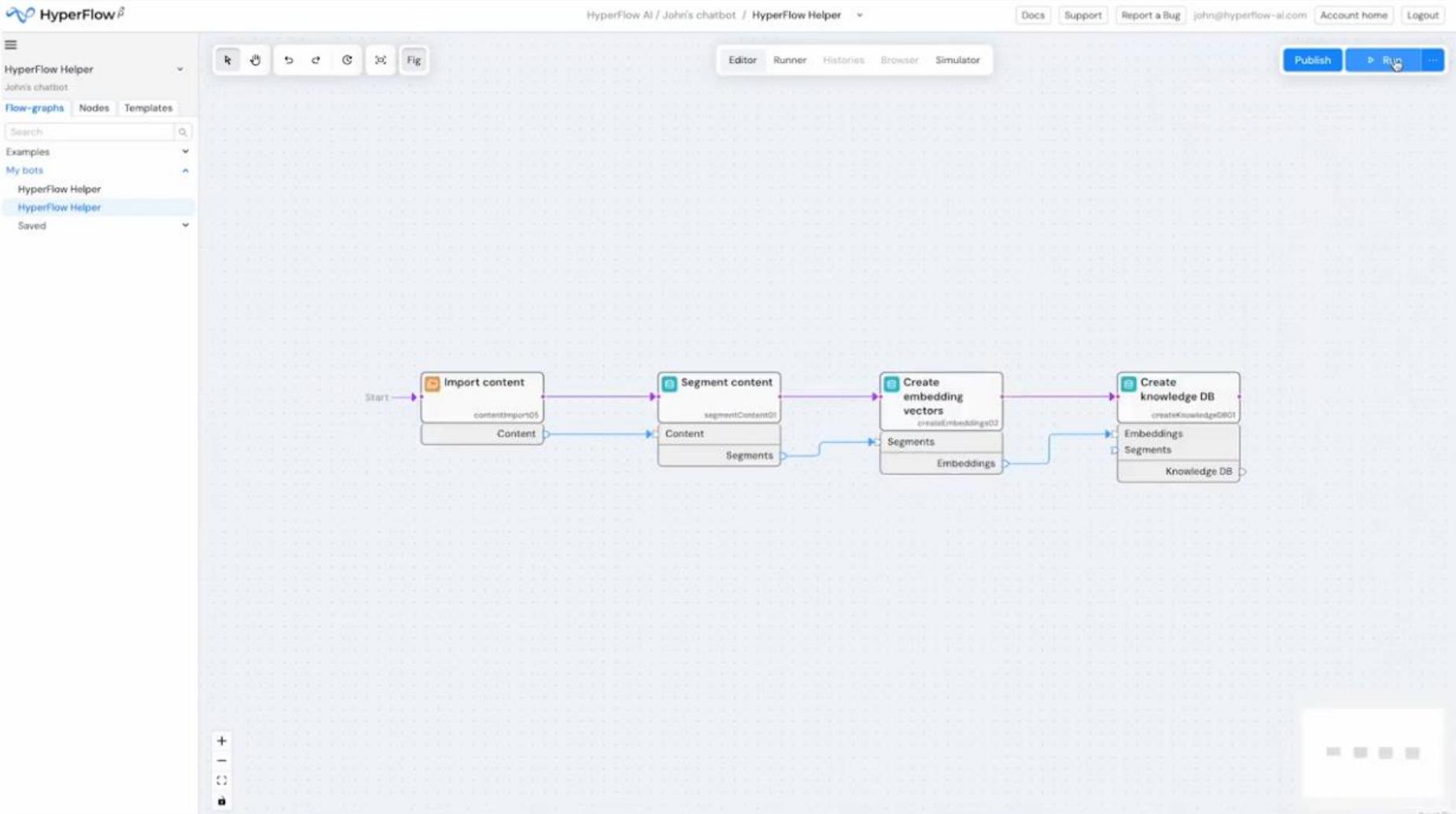
Click-crawl CSS selectors:

React Flow

Translated in AI

```
graph LR; Start((Start)) --> Import[Import content]; Import -- Content --> Segment[Segment content]; Segment -- Content --> CreateEmbedding[Create embedding vectors]; Segment -- Segments --> CreateEmbedding; CreateEmbedding -- Segments --> CreateDB[Create knowledge DB]; CreateEmbedding -- Embedding --> CreateDB; CreateDB -- Embedding --> KnowledgeDB[Knowledge DB]
```

RAG 기반 지식 DB 및 챗봇 구축

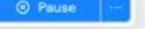


RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  / John's chatbot / HyperFlow Helper 

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator 

Run session tags:

Import content #45: contentImport06 Step 1

Parameters:

Import content: ...

Content form: Heading-separated text

Dynamic HTML site: Page loaded condition: DOM content loaded

Delay after load (secs): 4

Web import method: Site crawl

Depth: 4

Maximum pages: maximum

Subtree only

Click-crawl CSS selectors:  

URL filter pattern:

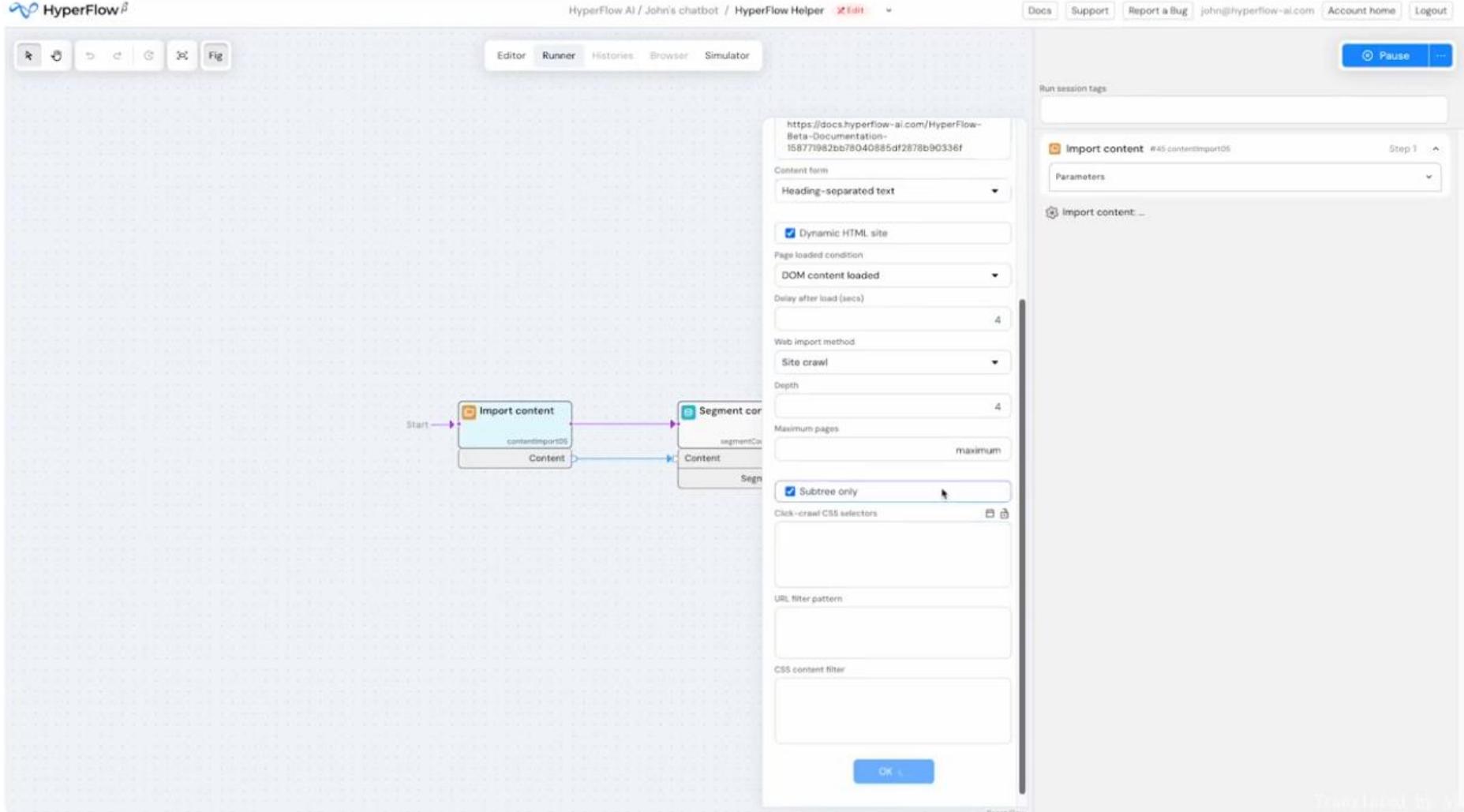
CSS content filter:

OK 

Start → Import content → Segment content

Content → Content

Segment content → Segments



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  HyperFlow AI / John's chatbot / HyperFlow Helper 

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator  Pause

Run session tags

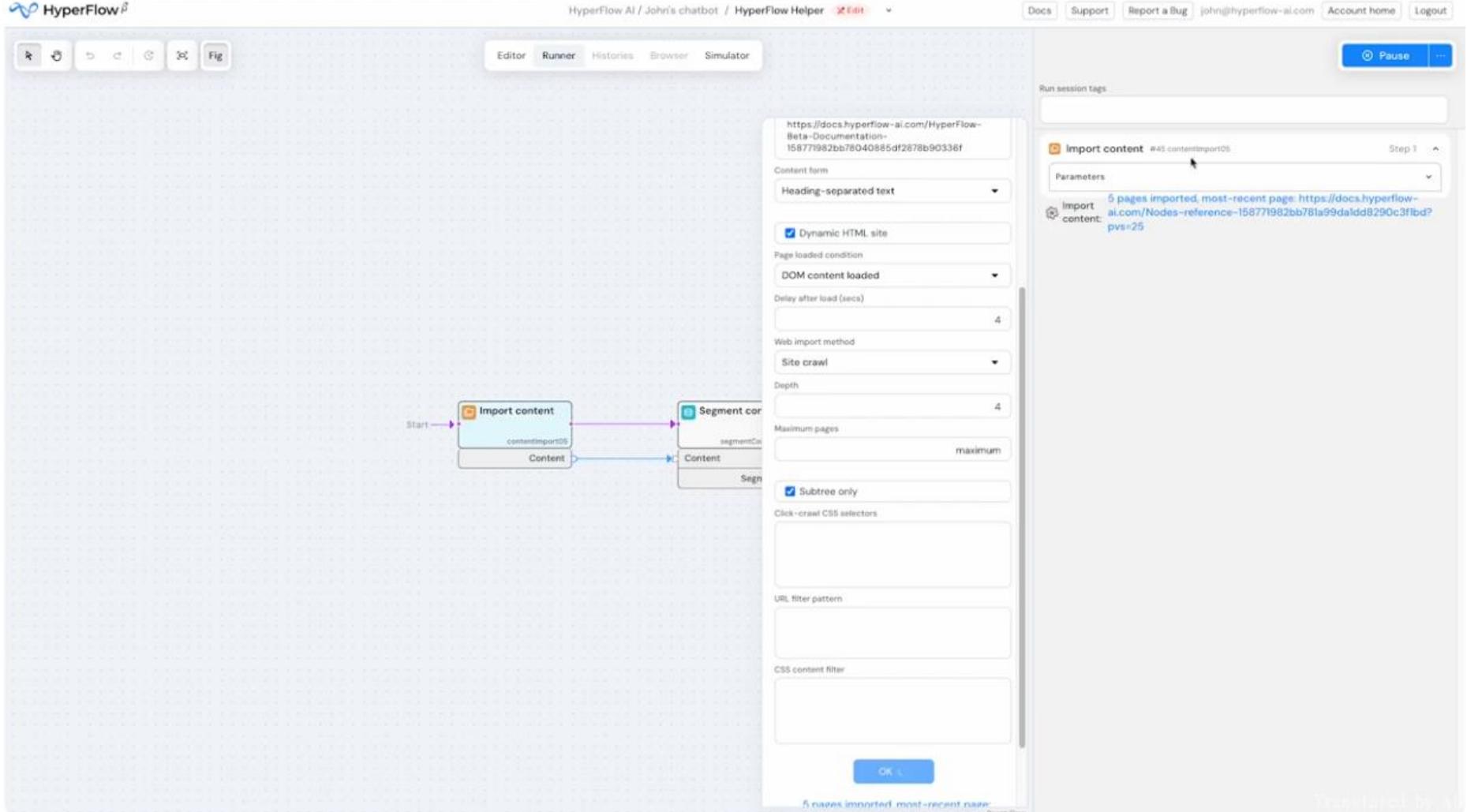
Import content #45 contentImport06 Step 1

Parameters

import content: 5 pages imported, most-recent page: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fbfd?pvs=25

Content form: Heading-separated text
Dynamic HTML site: checked
Page loaded condition: DOM content loaded
Delay after load (secs): 4
Web import method: Site crawl
Depth: 4
Maximum pages: maximum
Subtree only: checked
Click-crawl CSS selectors:
URL filter pattern:
CSS content filter:
OK

5 pages imported, most-recent page: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fbfd?pvs=25



```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment -- Content --> End((End))
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Segment content #48 segmentContent01 Step 2

All locks: 0

Segmenting service: Built-in segmenter

Segmentation source: Text

Service: Separator pattern

Pattern: \n

Overlap: 0

Tags: hyperflow-helper

Notes:

Import content #45 contentImport06 Step 1

Parameters:

Metadata: Import Count: 88 Import Set Tag: Content_Files.Import.2026-01-12 10:10:30

Segment content #48 segmentContent01 Step 2

Waiting on settings OK..

Import content

Content

Segment content

Content

Segments

Create embedding vectors

Segments

Embedding

Start

Content

Segments

Embedding

React Flow

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Segment content #48 segmentContent01 Step 2

All locks ⊕ ⊖ ⊕

Segmenting service: Built-in segmenter

Segmentation source: Text

Service: Separator pattern

Pattern: `-----\n`

Overlap: 0

Tags: hyperflow-helper

Notes:

Import content #45 contentImport06 Step 1

Parameters:

Metadata: Import Count: 89 Import Set Tag: Content_Files.Import.2028-01-12:10:18:50

Segment content #48 segmentContent01 Step 2

Waiting on settings OK...

React Flow

Translated by AI

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Embed[Create embedding vectors];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Import content #45 contentImport06 Step 1

Segment content #48 segmentContent01 Step 2

Segment content #48 segmentContent01 Step 2

Segmentation preview

Segment set Created: 2025-01-12 20:36 Type: separator-pattern Preview count: 20

HyperFlow Beta Documentation

Made with Drag image to reposition

Title: HyperFlow Beta Documentation

URL: https://docs.hyperflow-ai.com/H

HyperFlow Beta Documentation

Beta program welcome

Made with

Title: Beta program welcome

URL: https://docs.hyperflow-ai.com/Beta-program-welcome-534719830b780fba0f6c0834daa4553?pv=2

Beta program welcome: Welcome!

Welcome to the HyperFlow-AI Beta Release! This is a free, invitation-only release, and requests for access can be made on this page.

We trust that you will find HyperFlow-AI a useful tool for...

Beta program welcome: Terms of service

By creating a user account on hyperflow-ai.com under the HyperFlow AI beta program, you agree to the beta program terms of service that can be seen here.

Beta program welcome: Initial service quotas

The initial beta release is free and provides access to many third-party services.

Beta program welcome: Potential service interruptions

As a beta release, the service may experience interruptions at...

Start → Import content → Segment content → Create embedding vectors

Import content contentImport05 Content

Segment content segmentContent01 Content Segments

Create embedding vectors createEmbedding01 Embedding Segments

React Flow

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Embedding[Create embedding vectors];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  HyperFlow AI / John's chatbot / HyperFlow Helper  Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator 

Segmentation preview

Welcome to the hyperflow-AI Beta Release. This is a free, invitation-only release, and requests for access can be made on this page.

We trust that you will find HyperFlow-AI a useful tool for generative AI experimentation, rapid prototyping and production application development & deployment, and we will strive to continue its improvement and availability.

We look forward to receiving active feedback, and to engaging with you on our support Discord server, via email, and perhaps even in person. Our main headquarters is in Seoul, South Korea, and we are setting up an international office in Silicon Valley in early 2025.

Import content #45 contentImport06 Step 1
Segmenting service: Built-in segmenter

Segment content #48 segmentContent01 Step 2
Segment set: Created: 2025-01-12 20:36 Type: separator-pattern Preview count: 20

Segmentation preview

Segment set Created: 2025-01-12 20:36 Type: separator-pattern Preview count: 20

Segment content #48 segmentContent01 Step 2

Segmentation preview

Beta program welcome: Terms of service:
By creating a user account on hyperflow-ai.com under the HyperFlow AI beta program, you agree to the beta program terms of service that can be seen here.

Beta program welcome:
title: Beta program welcome
url: [https://docs.hyperflow-ai.com/Beta-program>Welcome-\(74771982bb780feb0f6c0834daa4553?pv=25](https://docs.hyperflow-ai.com/Beta-program>Welcome-(74771982bb780feb0f6c0834daa4553?pv=25)
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Beta program welcome: Initial service quotas:
The initial beta release is free and provides access to many 3rd-party commercial services like OpenAI, Cohere, Pinecone, etc., as part of this free release. There will be resource quotas in place, though we will make them generous initially.

Beta program welcome:
As usage grows and we release BETA (bring your own key) capabilities, the quotas may be tightened and you will be invited to make use of your own external service API keys to have related quotas removed.

Beta program welcome:
Some quotas relate to inbound API calls to our chatbot API servers, and CPU & GPU & storage usage within our hosting clusters. If you hit any of these quotas and would like to request more, please contact us via email with details of your request and we will be happy to discuss this with you.

Beta program welcome:
title: Beta program welcome
url: [https://docs.hyperflow-ai.com/Beta-program>Welcome-\(74771982bb780feb0f6c0834daa4553?pv=25](https://docs.hyperflow-ai.com/Beta-program>Welcome-(74771982bb780feb0f6c0834daa4553?pv=25)
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Beta program welcome: Potential service interruptions:
As a beta release, the service may experience interruptions at any time, though we will attempt to minimize any such. We may also announce scheduled

Segmentation preview

Beta program welcome: Terms of service:
Welcome to the HyperFlow-AI Beta Release. This is a free, invitation-only release, and requests for access can be made on this page.

Beta program welcome:
We trust that you will find HyperFlow-AI a useful tool for generative AI experimentation, rapid prototyping and production application development & deployment, and we will strive to continue its improvement and availability.

Beta program welcome: Initial service quotas:
The initial beta release is free and provides access to many 3rd-party commercial services like OpenAI, Cohere, Pinecone, etc., as part of this free release. There will be resource quotas in place, though we will make them generous initially.

Beta program welcome: Potential service interruptions:
As a beta release, the service may experience interruptions at any time, though we will attempt to minimize any such. We may also announce scheduled

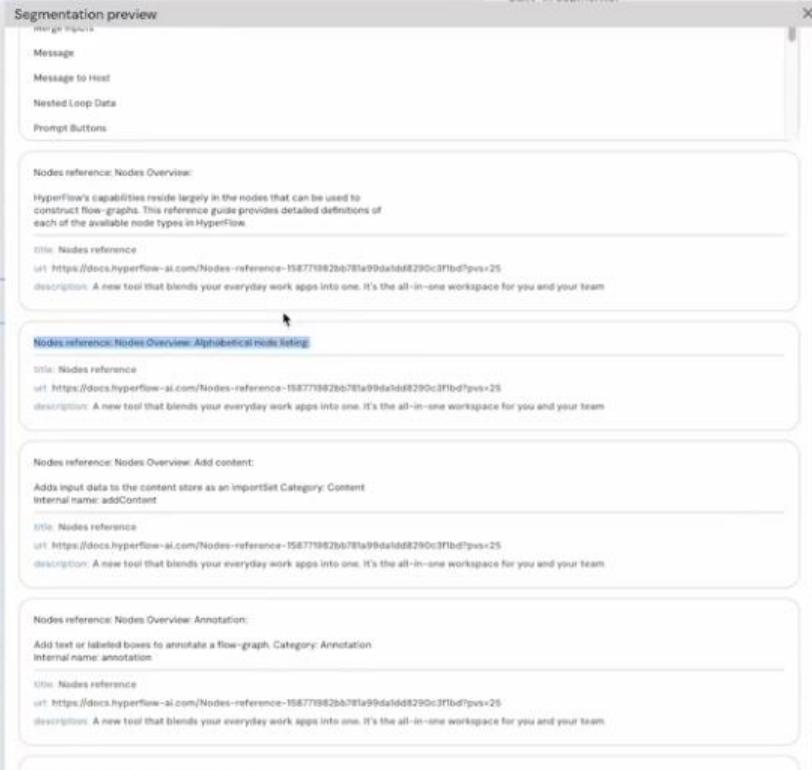
RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  HyperFlow AI / John's chatbot / HyperFlow Helper 

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator  Run session tags

Segment cont... #48 segmentContent01 Step 2 Segmenting service Built-in segmenter All locks 

import content #45 contentImport06 Step 1 Parameters Metadata Import Count: 88 Import Set Tag: Content_Files.Import.2028-01-12T10:50:00

Segment content #48 segmentContent01 Step 2 Segment set Created: 2025-01-12 20:36 Type: separator-pattern Preview count: 20

Nodes reference: Nodes Overview: [Nodes reference](#)
HyperFlow's capabilities reside largely in the nodes that can be used to construct flow-graphs. This reference guide provides detailed definitions of each of the available node types in HyperFlow.

title: Nodes reference
url: <https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fb0d7pvsx25>
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Alphabetical node listing: [Nodes reference](#)
title: Nodes reference
url: <https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fb0d7pvsx25>
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Add content: [Nodes reference](#)
Add input data to the content store as an importSet. Category: Content Internal name: addContent

title: Nodes reference
url: <https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fb0d7pvsx25>
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Annotation: [Nodes reference](#)
Add text or labeled boxes to annotate a flow-graph. Category: Annotation Internal name: annotation

title: Nodes reference
url: <https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99daldd8290c3fb0d7pvsx25>
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: [Nodes reference](#)

HyperFlow Beta Documentation Made with Drag image to reposition [HyperFlow Beta Documentation](#)
title: HyperFlow Beta Documentation
url: <https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation>

Beta program welcome Release notes [Beta program welcome](#)
title: Beta program welcome
url: <https://docs.hyperflow-ai.com/Beta-program-welcome-15471982bb781a99daldd8290c3fb0d7pvsx25>

HyperFlow Beta Documentation / Beta program welcome [HyperFlow Beta Documentation](#)
title: Beta program welcome
url: <https://docs.hyperflow-ai.com/Beta-program-welcome-15471982bb781a99daldd8290c3fb0d7pvsx25>

Beta program welcome: Welcome! [Beta program welcome](#)
Welcome to the HyperFlow-AI Beta Release! This is a free, invitation-only release, and requests for access can be made on this page.

We trust that you will find HyperFlow-AI a useful tool for... [Beta program welcome](#)

Beta program welcome: Terms of service: [Beta program welcome](#)
By creating a user account on hyperflow-ai.com under the HyperFlow AI beta program, you agree to the beta program terms of service that can be seen here.

Beta program welcome: Initial service quotas: [Beta program welcome](#)
The initial beta release is free and provides access to many limited features.

Beta program welcome: Potential service interruptions: [Beta program welcome](#)
As a beta release, the service may experience interruptions at...

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  HyperFlow AI / John's chatbot / HyperFlow Helper 

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator 

Segment content #48 segmentContent01 Step 2
Segmenting service: Built-in segmenter

Run session tags

import content #45 contentImport06 Step 3
Parameters
Metadata Import Count: 89 Import Set Tag: Content_Files.Import.2028-01-12:19:56

Segment content #48 segmentContent01 Step 2
Segment set Created: 2025-01-12 20:36 Type: separator-pattern Preview count: 20

HyperFlow Beta Documentation
Made with Drag Image to reposition
URL: https://docs.hyperflow-ai.com/HyperFlow-Beta-Documentation
Beta program welcome
Release notes
hyperflow-ai.com/hyperflow - the HyperFlow IDE

HyperFlow Beta Documentation /
Beta program welcome
Made with
Title: Beta program welcome
URL: https://docs.hyperflow-ai.com/Beta-program-welcome-154719820b780f6c0f634daa45537pv-25

Beta program welcome: Welcome!
Welcome to the HyperFlow-AI Beta Release! This is a free, invitation-only release, and requests for access can be made on this page.
We trust that you will find HyperFlow-AI a useful tool for...

Beta program welcome: Initial service quotas:
The initial beta release is free and provides access to many 3rd-party services.

Beta program welcome: Potential service interruptions:
As a beta release, the service may experience interruptions at...

Segmentation preview

HyperFlow's capabilities reside largely in the nodes that can be used to construct flow-graphs. This reference guide provides detailed definitions of each of the available node types in HyperFlow.

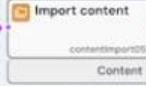
Nodes reference
url: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99da0dd8290c3fb0d7pvsx25
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

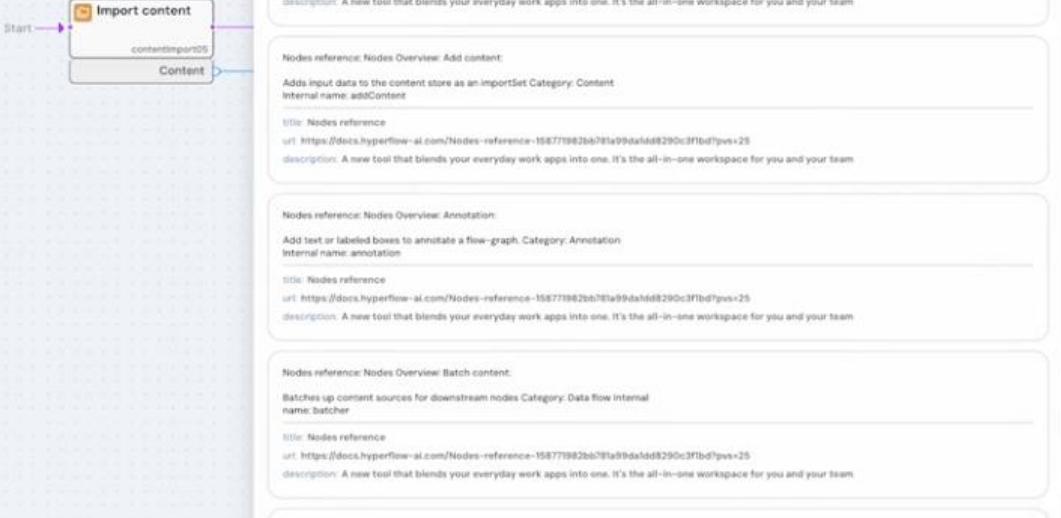
Nodes reference: Nodes Overview: Alphabetical node listing:
title: Nodes reference
url: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99da0dd8290c3fb0d7pvsx25
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Add content:
Adds input data to the content store as an importSet Category: Content Internal name: addContent
title: Nodes reference
url: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99da0dd8290c3fb0d7pvsx25
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Annotation:
Add text or labeled boxes to annotate a flow-graph. Category: Annotation Internal name: annotation
title: Nodes reference
url: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99da0dd8290c3fb0d7pvsx25
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Nodes reference: Nodes Overview: Batch content:
Batches up content sources for downstream nodes Category: Data flow Internal name: batcher
title: Nodes reference
url: https://docs.hyperflow-ai.com/Nodes-reference-158771982bb781a99da0dd8290c3fb0d7pvsx25
description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team.

Start  contentImport05 Content



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Segment content #48 segmentContent01 Step 2

All locks

Segmenting service: Built-in segmenter

Segmentation source: Text

Service: Separator pattern

Pattern: \n

Overlap: 0

Tags: hyperflow-helper

Notes:

Preview

OK

200 segments created

Import content #45 contentImport06 Step 1

Parameters:

Metadata: Import Count: 88 Import Set Tag: Content_Files.Import.2028-01-12 01:15:30

Segment content #48 segmentContent01 Step 2

Parameters:

Segment content: 200 segments created

React Flow

Translated by AI

```
graph LR; Start((Start)) --> Import[Import content]; Import --> Segment[Segment content]; Segment --> Embedding[Create embedding vectors];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

Import content #45 contentImport06 Step 1

Parameters

Metadata Import Count: 89 Import Set Tag: Content.Files.Import.2025-01-12.11:18:50

Segment content #48 segmentContent01 Step 2

Parameters

Metadata Segment Count: 82 Segmentation Set Tag: Segments.segmentation.2025-01-12.11:18:43

Create embedding vectors #46 createEmbeddings02 Step 3

Waiting on settings OK...

Insert flow Translated by AI

```
graph LR; Import[Import content] --> Segment[Segment content]; Segment --> Embedding[Create embedding vectors]; Embedding --> Knowledge[Create knowledge D]
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Import content #45 contentImport05 Step 1

Parameters

Metadata Import Count: 89 Import Set Tag: Content.Files.Import.2025-01-12 11:18:48

Segment content #48 segmentContent01 Step 2

Parameters

Metadata Segment Count: 804 Segmentation Set Tag: Segments.segmentation.2025-01-12 11:18:48

Create embedding vectors #46 createEmbeddings02 Step 3

Parameters

Create embedding vectors: 600 embeddings created, 33231 total input tokens

Create embedding vectors: 600 embeddings created, 33231 total input tokens

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow  HyperFlow AI / John's chatbot / HyperFlow Helper 

Editor Runner Histories Browser Simulator 

Run session tags:

Import content #45 contentImport06 Step 1

Parameters:

Metadata: Import Count: 89 Import Set Tag: Content.Files.Import.2025-01-12T10:18:48

Segment content #48 segmentContent01 Step 2

Parameters:

Metadata: Segment Count: 804 Segmentation Set Tag: Segments.segmentation.2025-01-12T10:45

Create embedding vectors #46 createEmbeddings02 Step 3

Parameters:

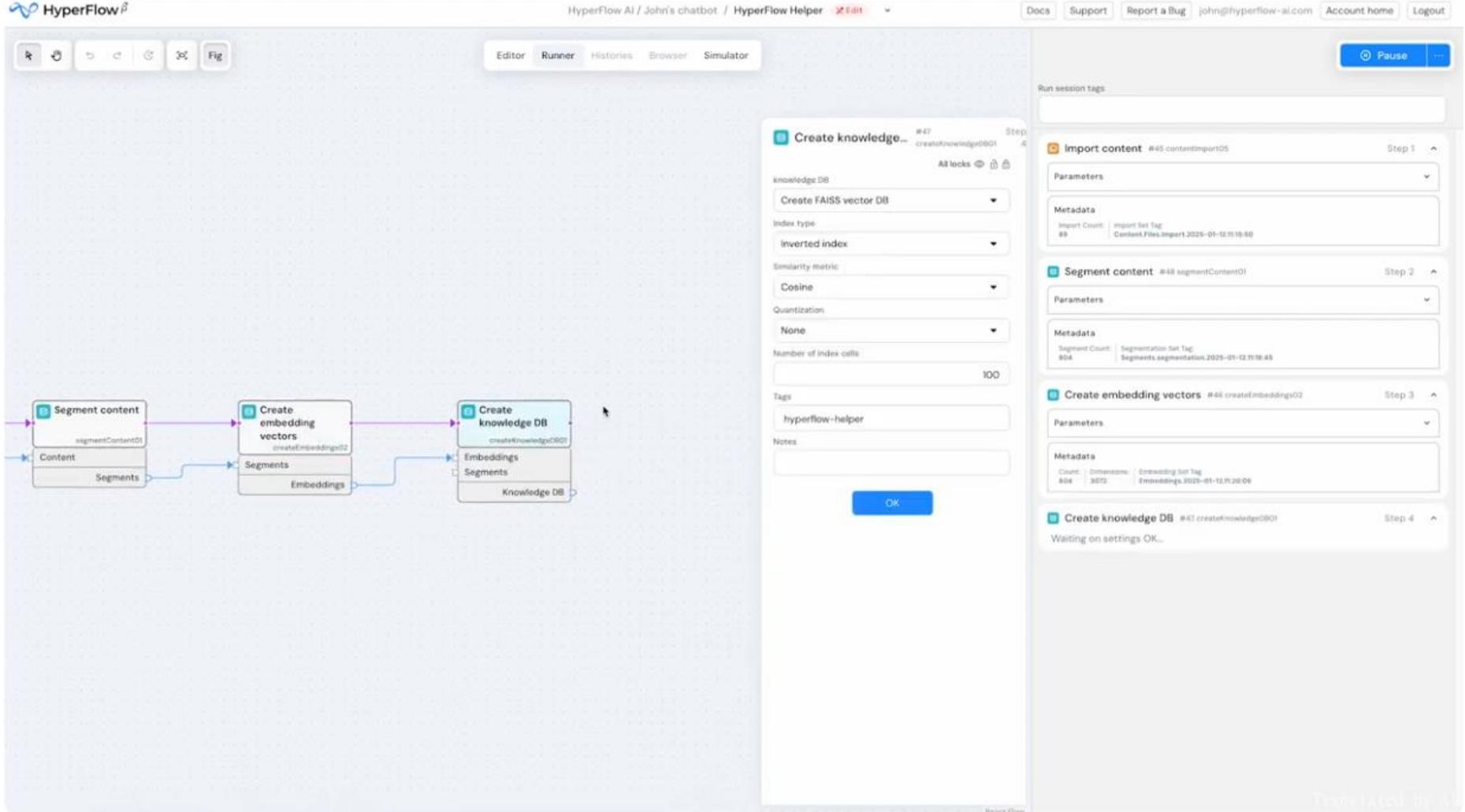
Metadata: Count: 804 Dimensions: 3072 Embedding Set Tag: Embeddings.2025-01-12T10:20:08

Create knowledge DB #47 createKnowledgeDB01 Step 4

Waiting on settings OK...

React Flow

Translated by AI



```
graph LR; A[Segment content] --> B[Create embedding vectors]; B --> C[Create knowledge DB];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper ✖ Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Run session tags

Step 1

import content #45:contentImport06

Parameters

Metadata

Import Count: 89 Import Set Tag: Content_Files.Import.2025-01-12:10:18:46

Step 2

Segment content #48:segmentContent01

Parameters

Metadata

Segment Count: 804 Segmentation Set Tag: Segments.segmentation.2025-01-12:11:18:45

Step 3

Create embedding vectors #46:createEmbeddings02

Parameters

Metadata

Count: 804 Dimensions: 3072 Embedding Set Tag: Embeddings.2025-01-12:11:30:06

Step 4

Create knowledge DB #47:createKnowledge0301

Parameters

Metadata

Last vector Count: 804 vector 0 tag: vvectorDBs.2025-01-12:11:29:31

End of run

React Flow

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Pause

Run session tags

Import content #45 contentImport06 Step 1

Parameters

Metadata Import Count: 88 Import Set Tag: Content.Files.Import.2025-01-12 10:10:40

Segment content #48 segmentContent01 Step 2

Parameters

Metadata Segment Count: 804 Segmentation Set Tag: Segments.segmentation.2025-01-12 10:10:40

Create embedding vectors #46 createEmbeddings02 Step 3

Parameters

Metadata Count: 804 Dimensions: 3072 Embedding Set Tag: Embeddings.2025-01-12 10:10:40

Create knowledge DB #47 createKnowledge0801 Step 4

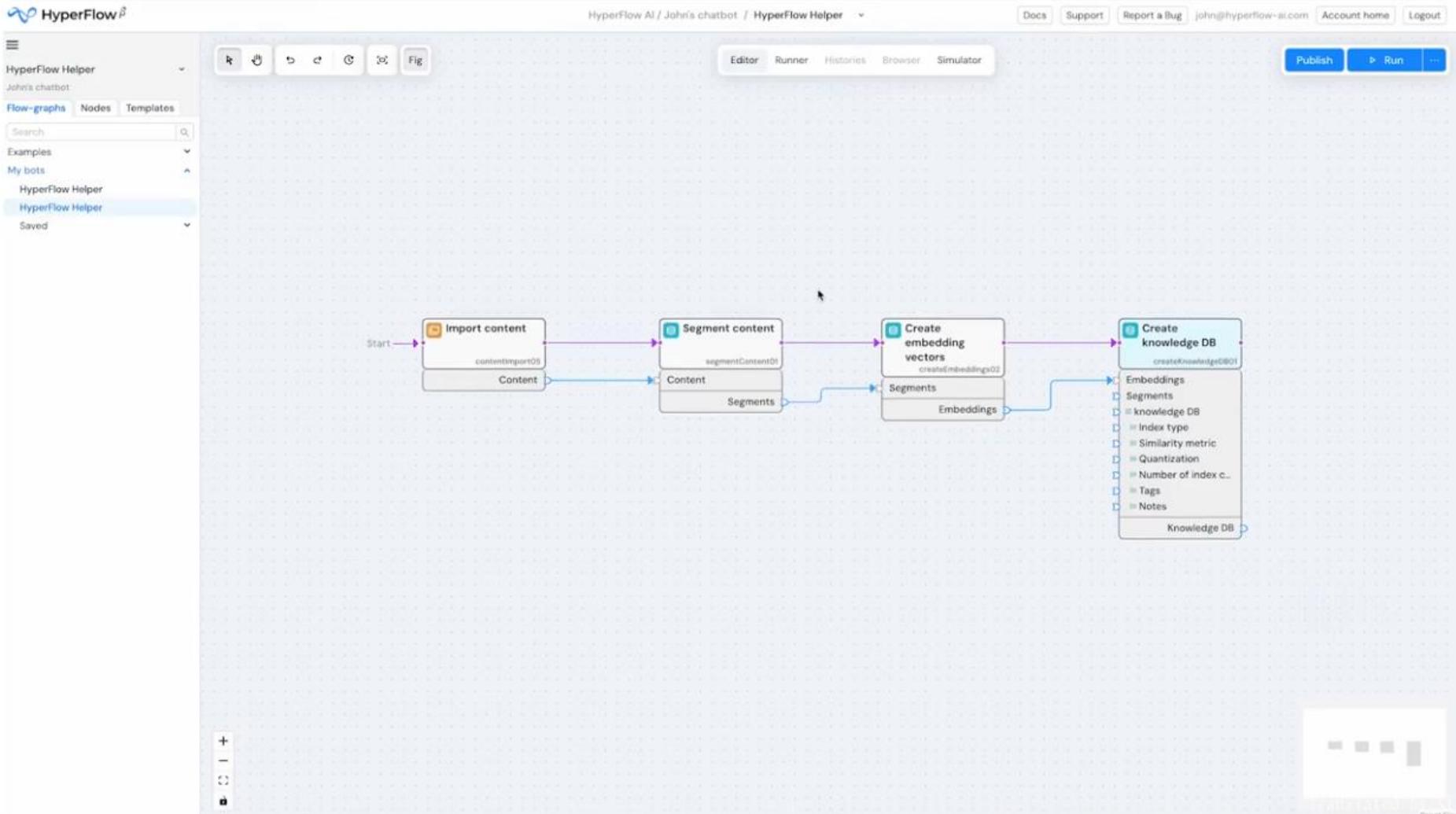
Parameters

Metadata Waiting on settings OK... Loaded Vector Count: 804 Vector DB Tag: VectorDBs.2025-01-12 10:10:40

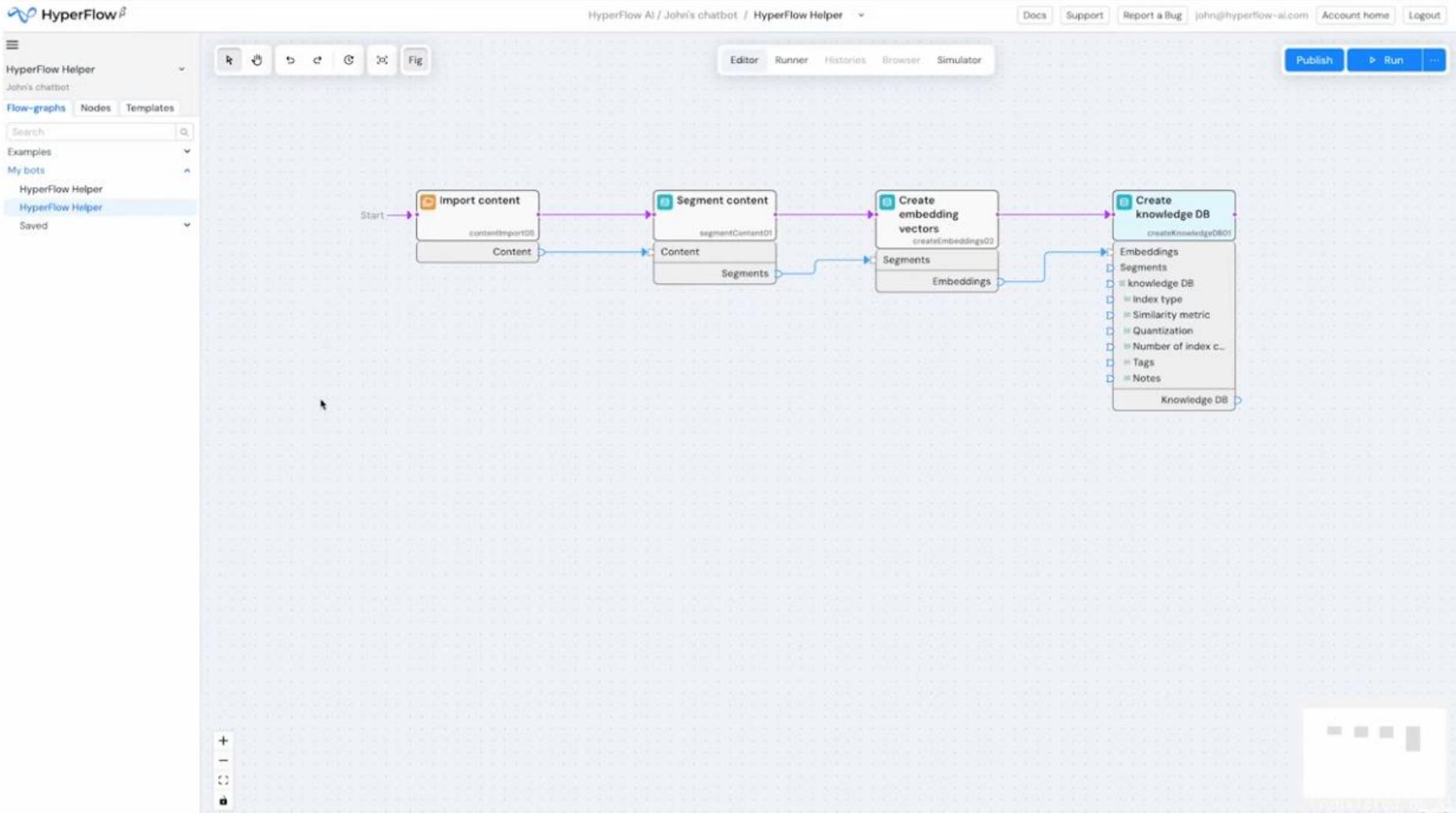
End of run

Translated by

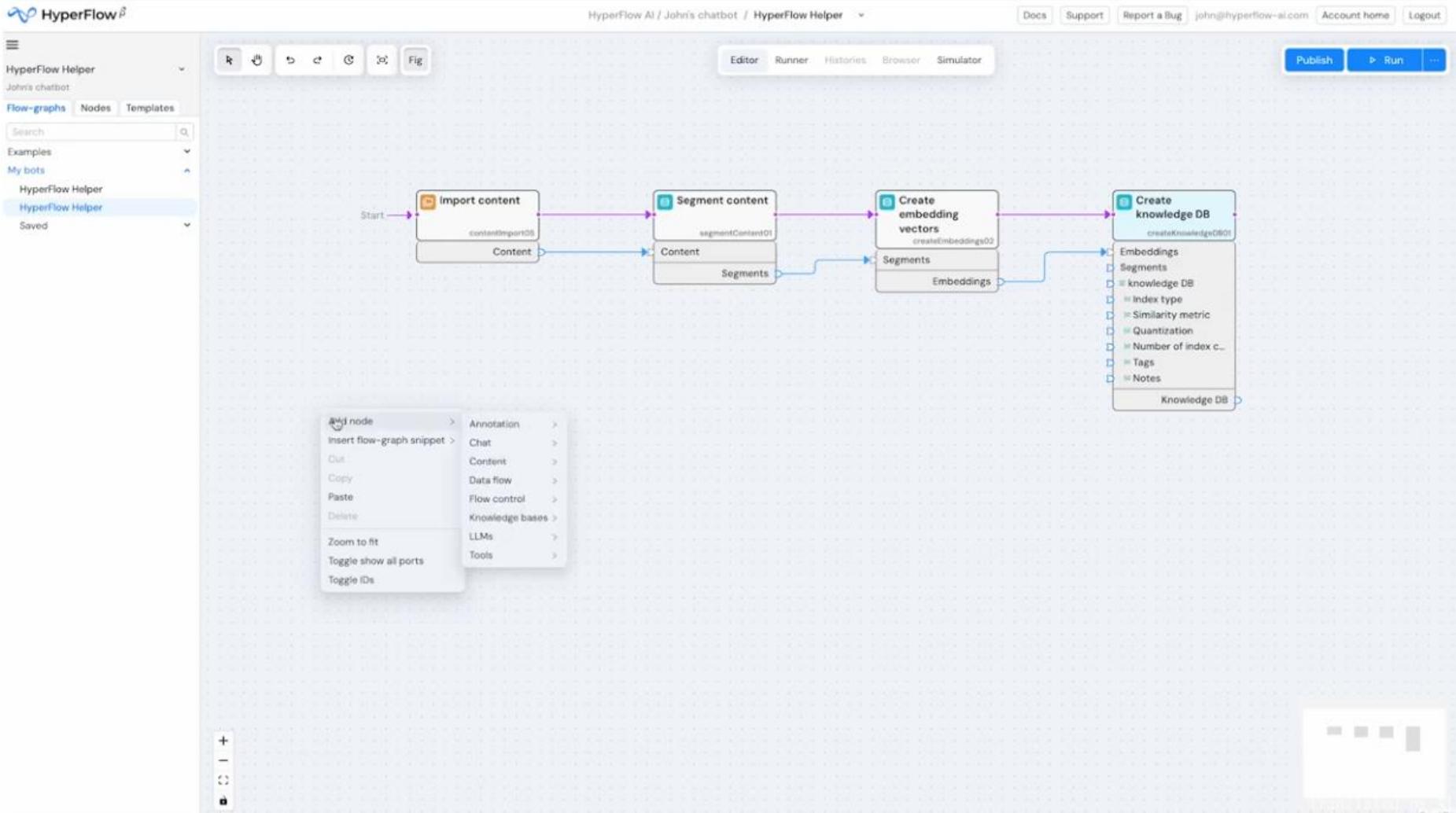
RAG 기반 지식 DB 및 챗봇 구축



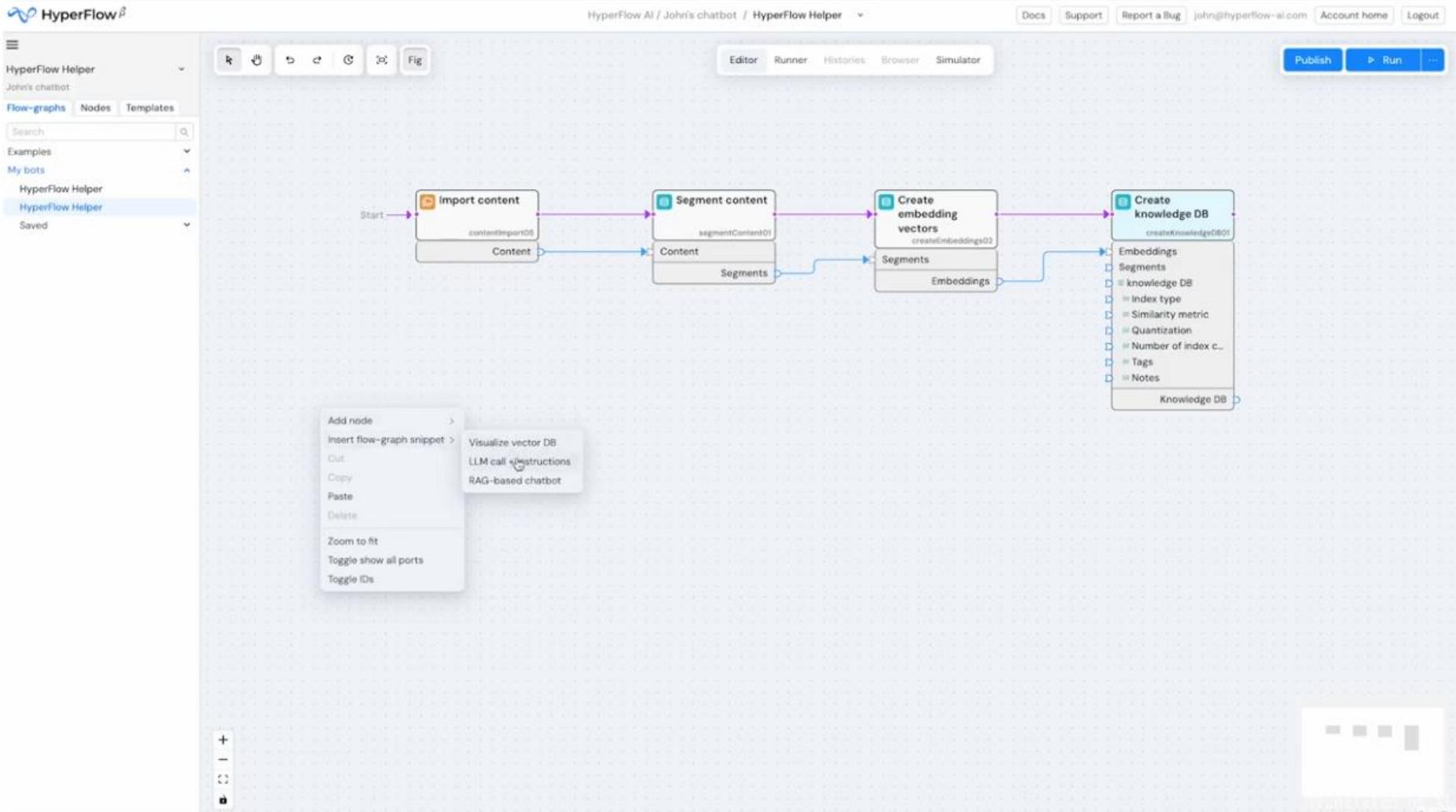
RAG 기반 지식 DB 및 챗봇 구축



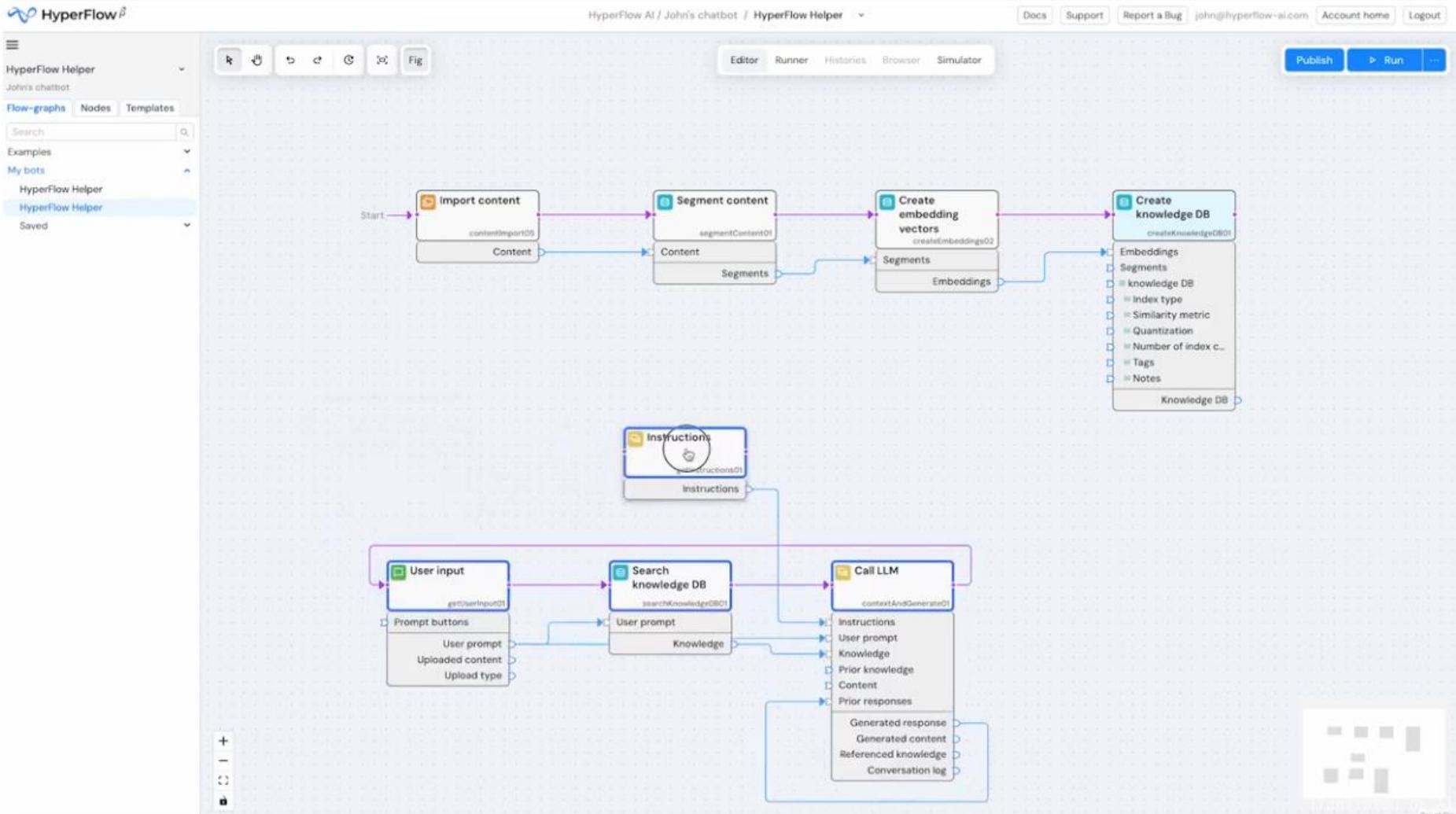
RAG 기반 지식 DB 및 챗봇 구축



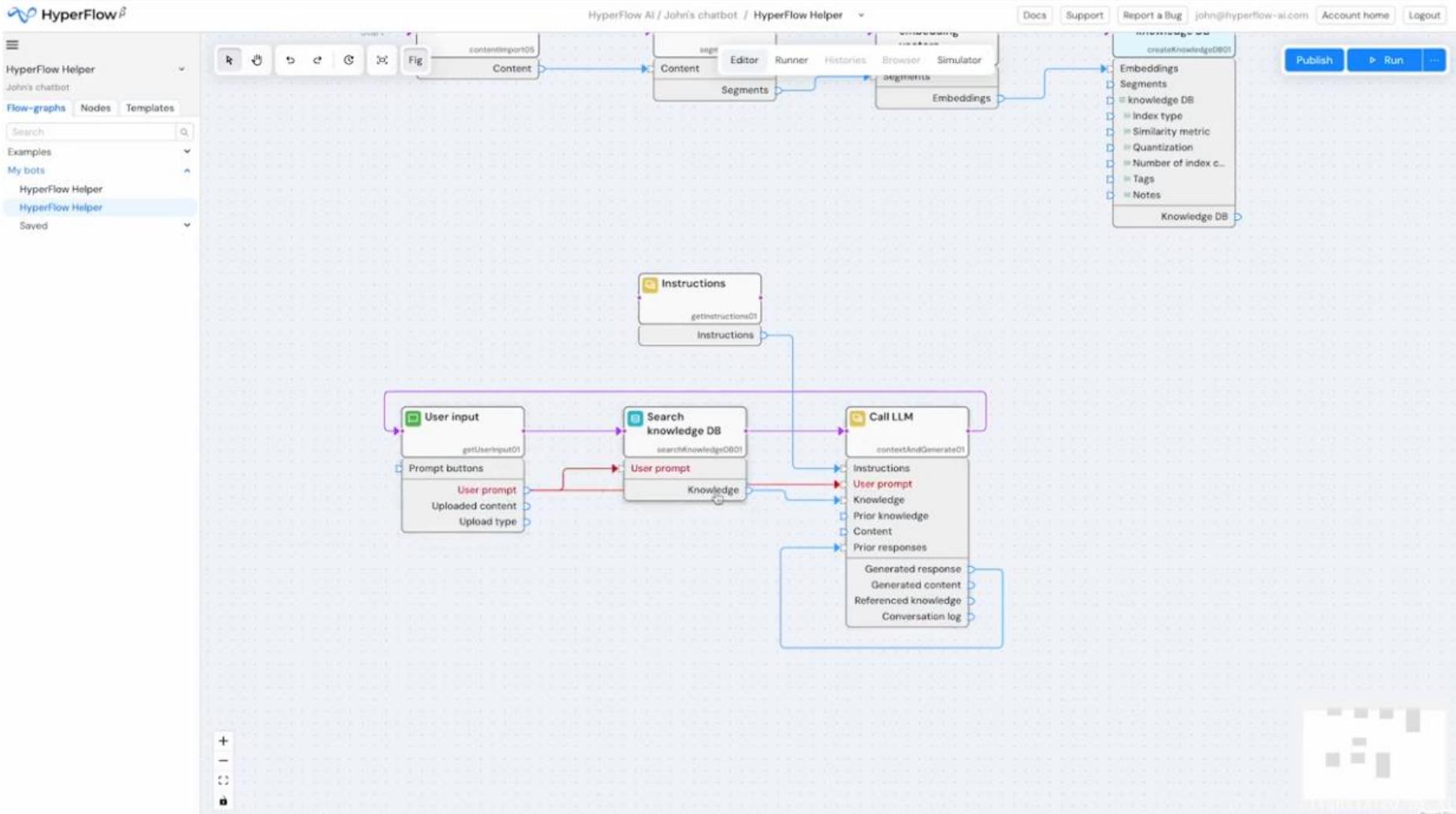
RAG 기반 지식 DB 및 챗봇 구축



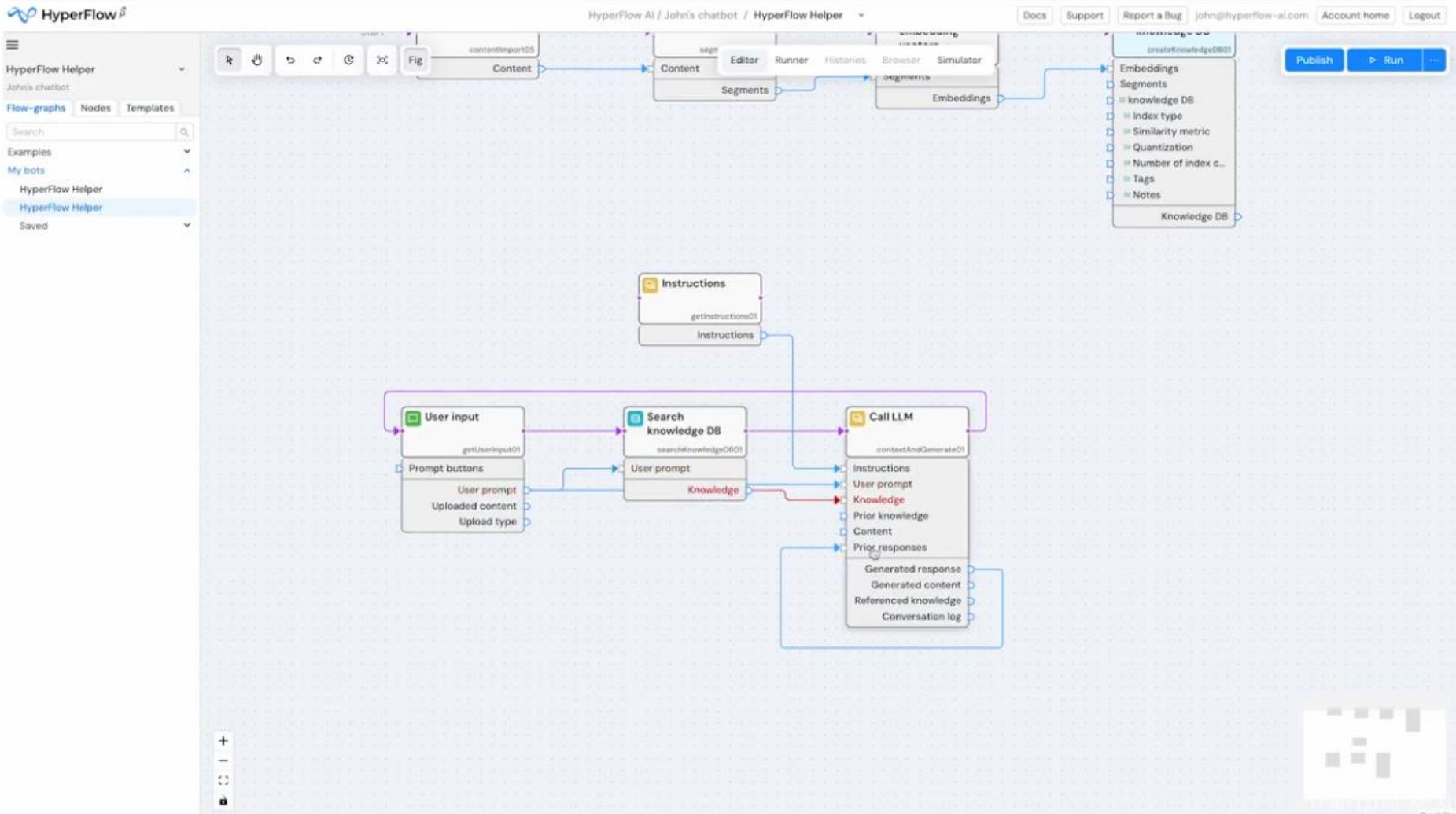
RAG 기반 지식 DB 및 챗봇 구축



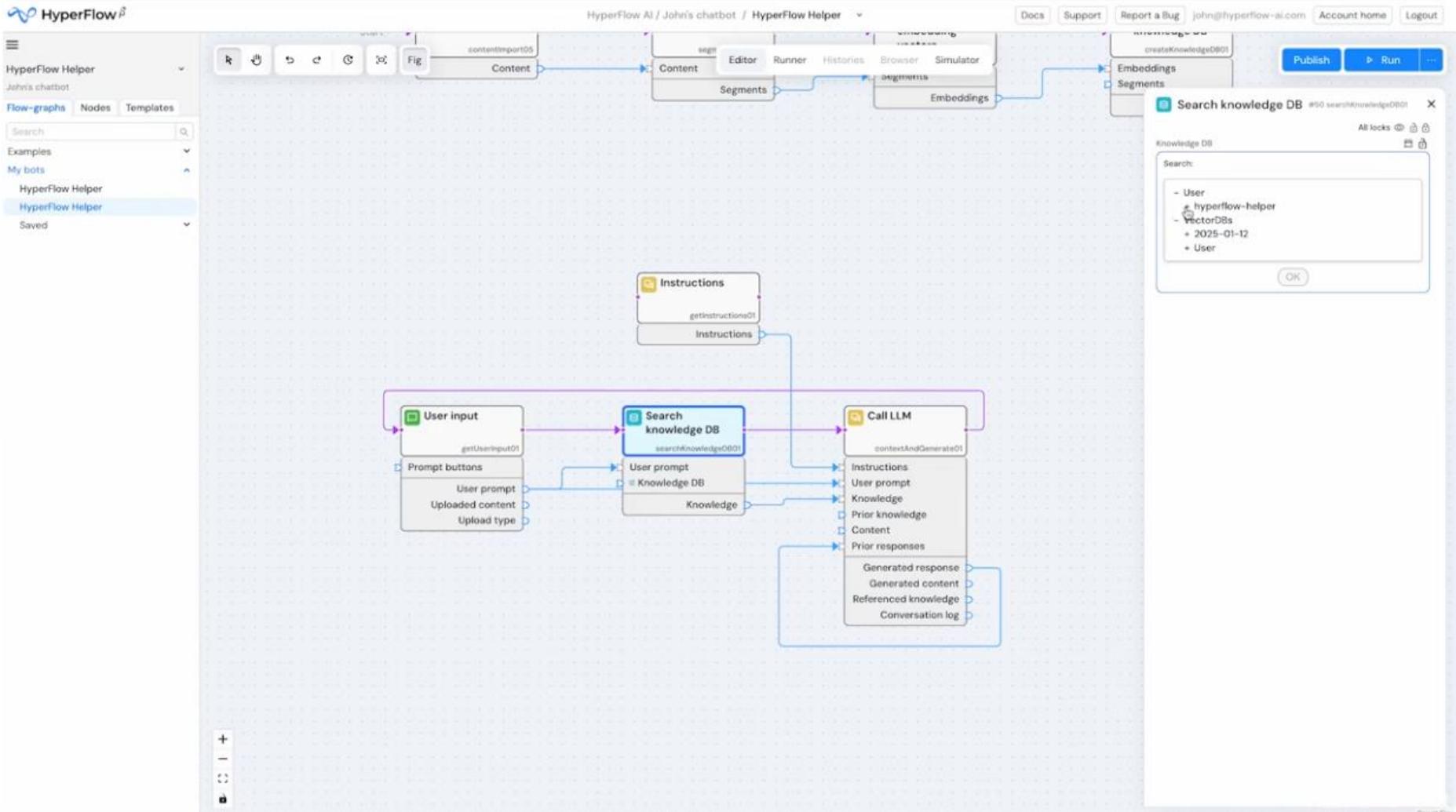
RAG 기반 지식 DB 및 챗봇 구축



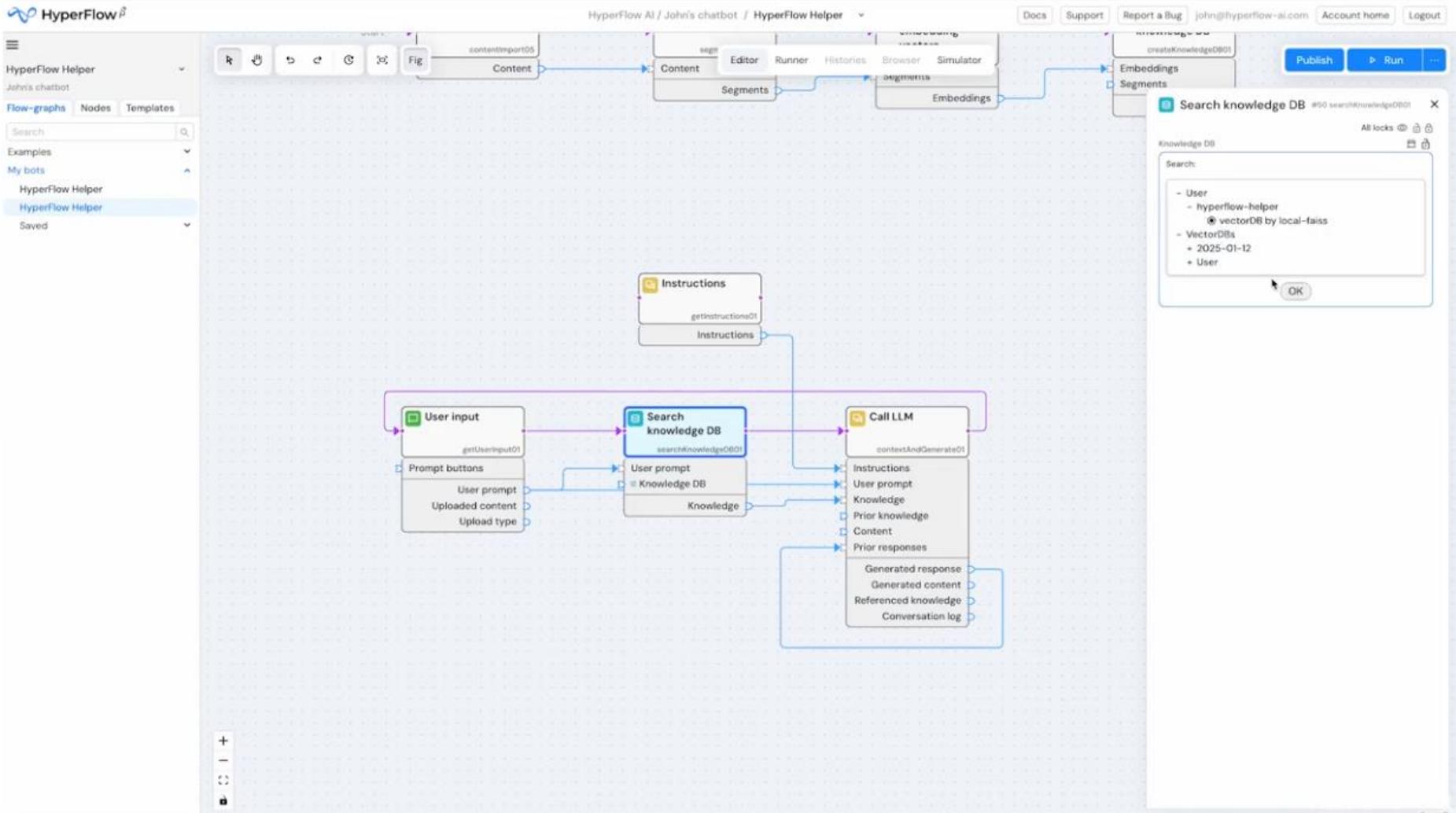
RAG 기반 지식 DB 및 챗봇 구축



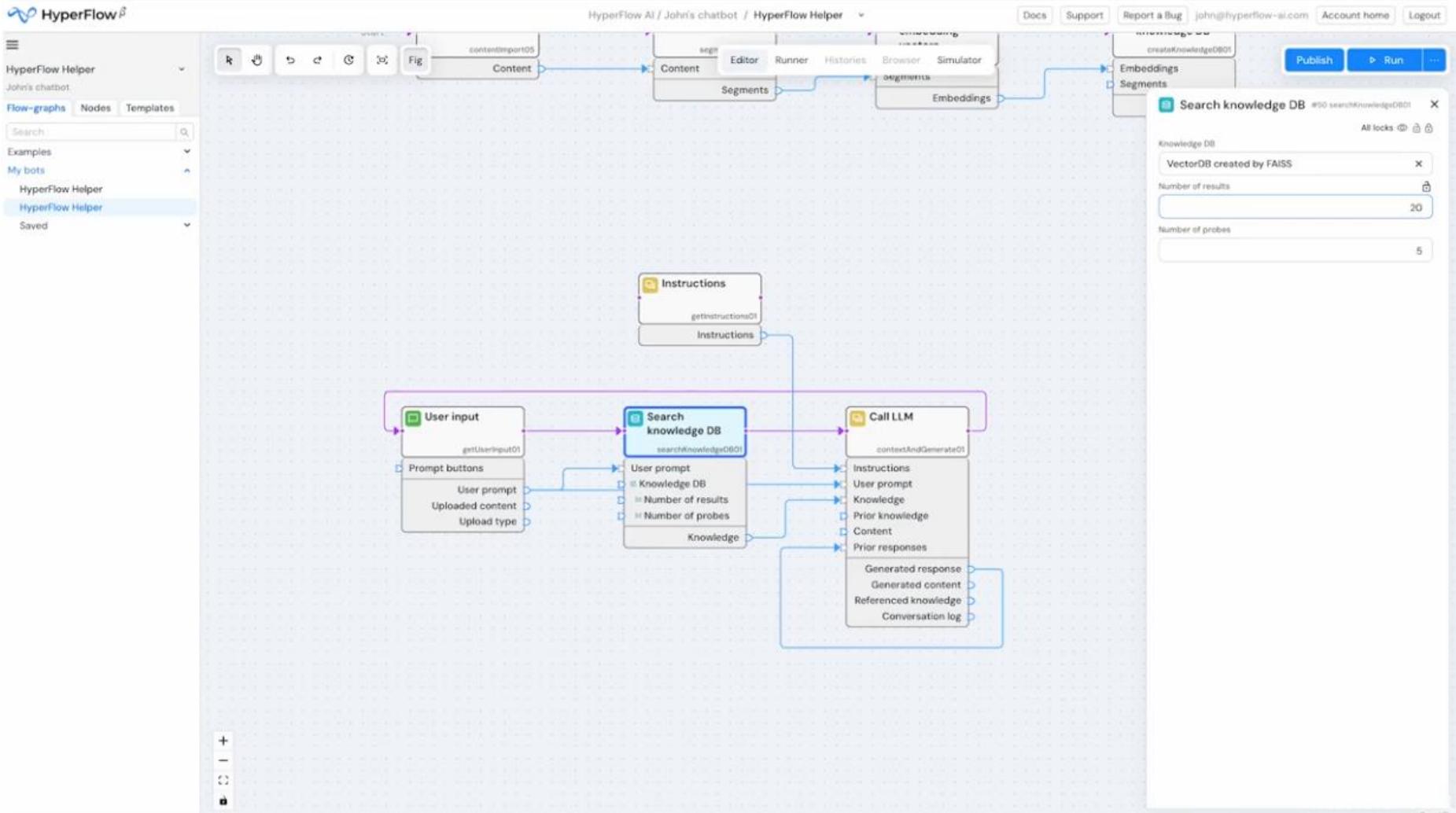
RAG 기반 지식 DB 및 챗봇 구축



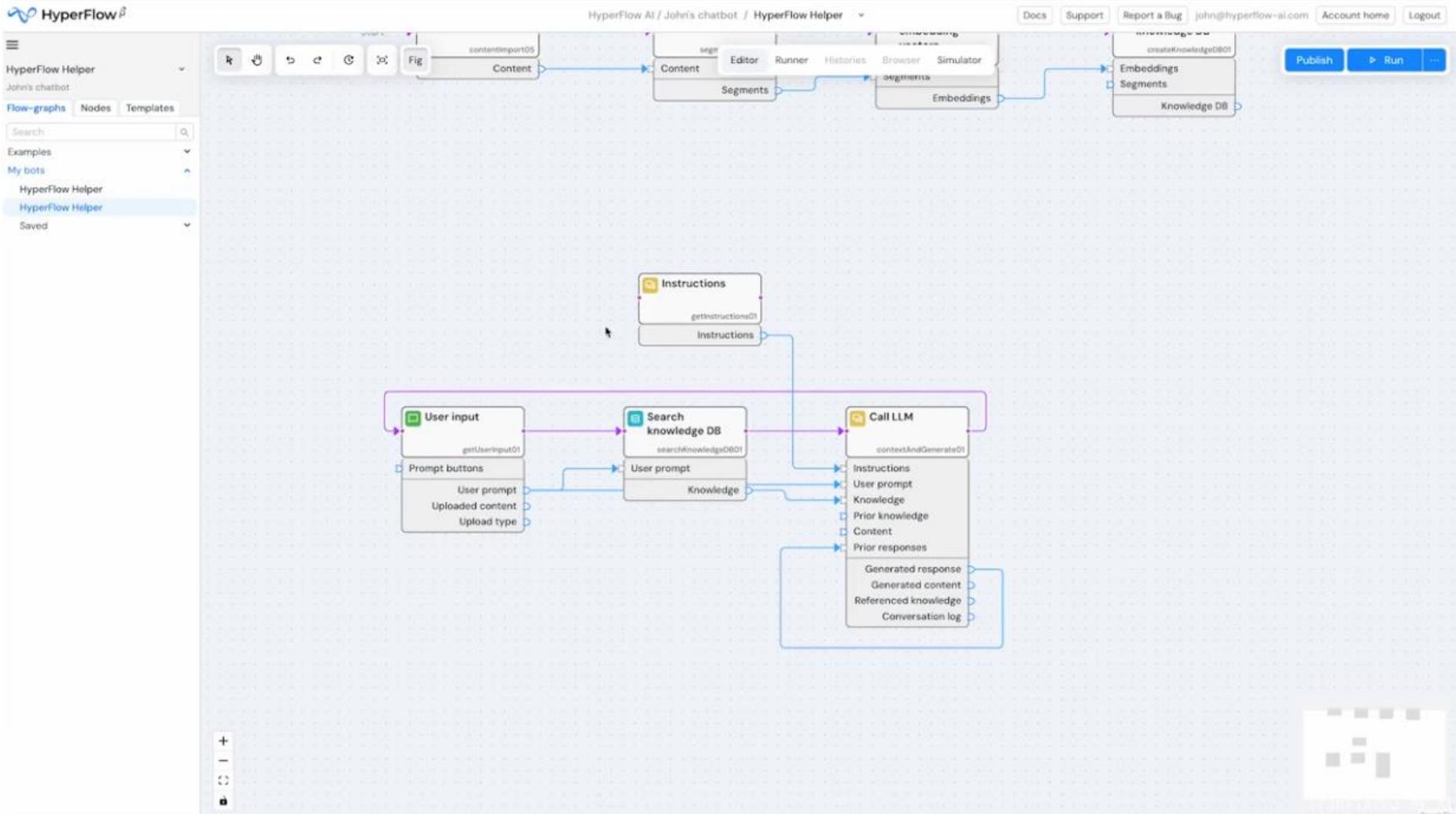
RAG 기반 지식 DB 및 챗봇 구축



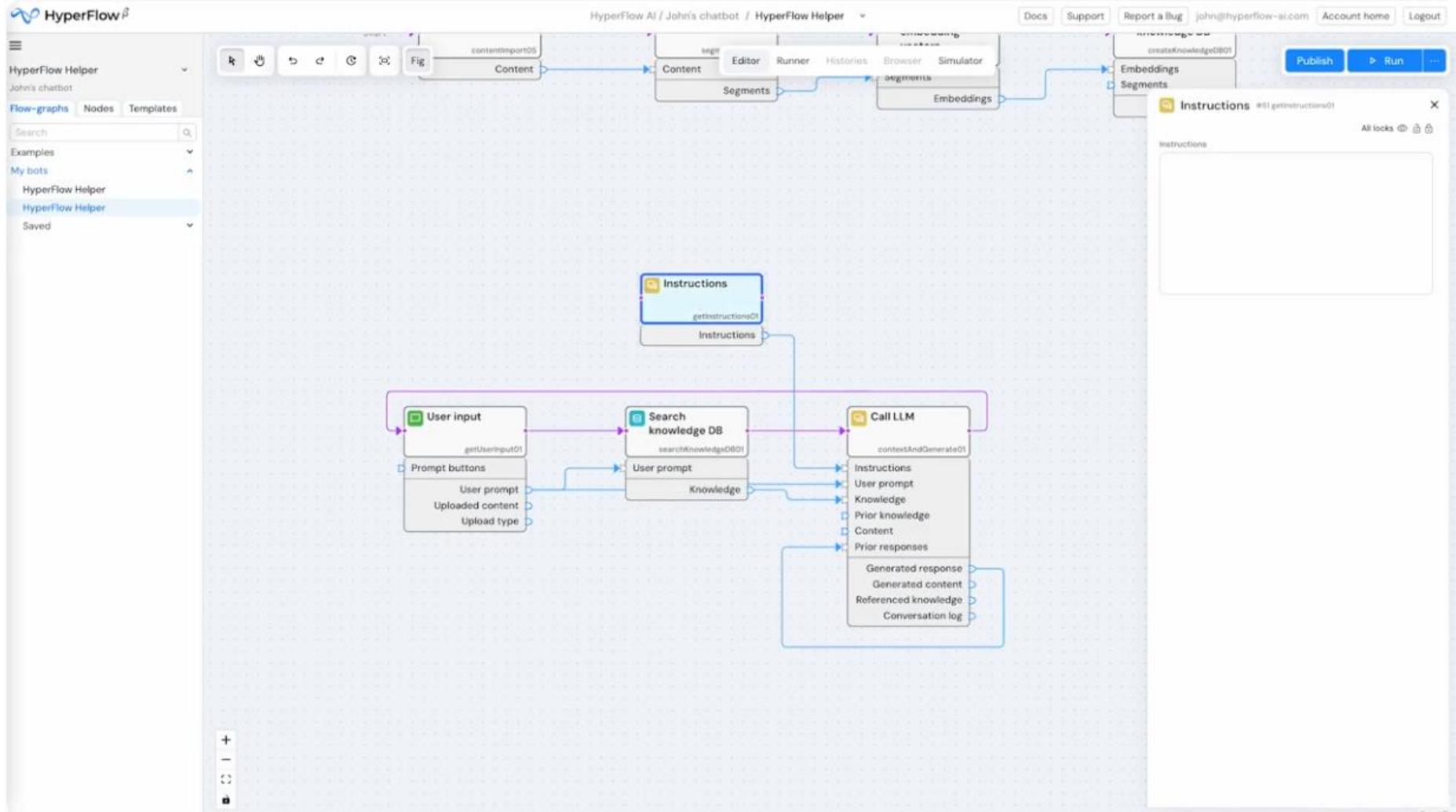
RAG 기반 지식 DB 및 챗봇 구축



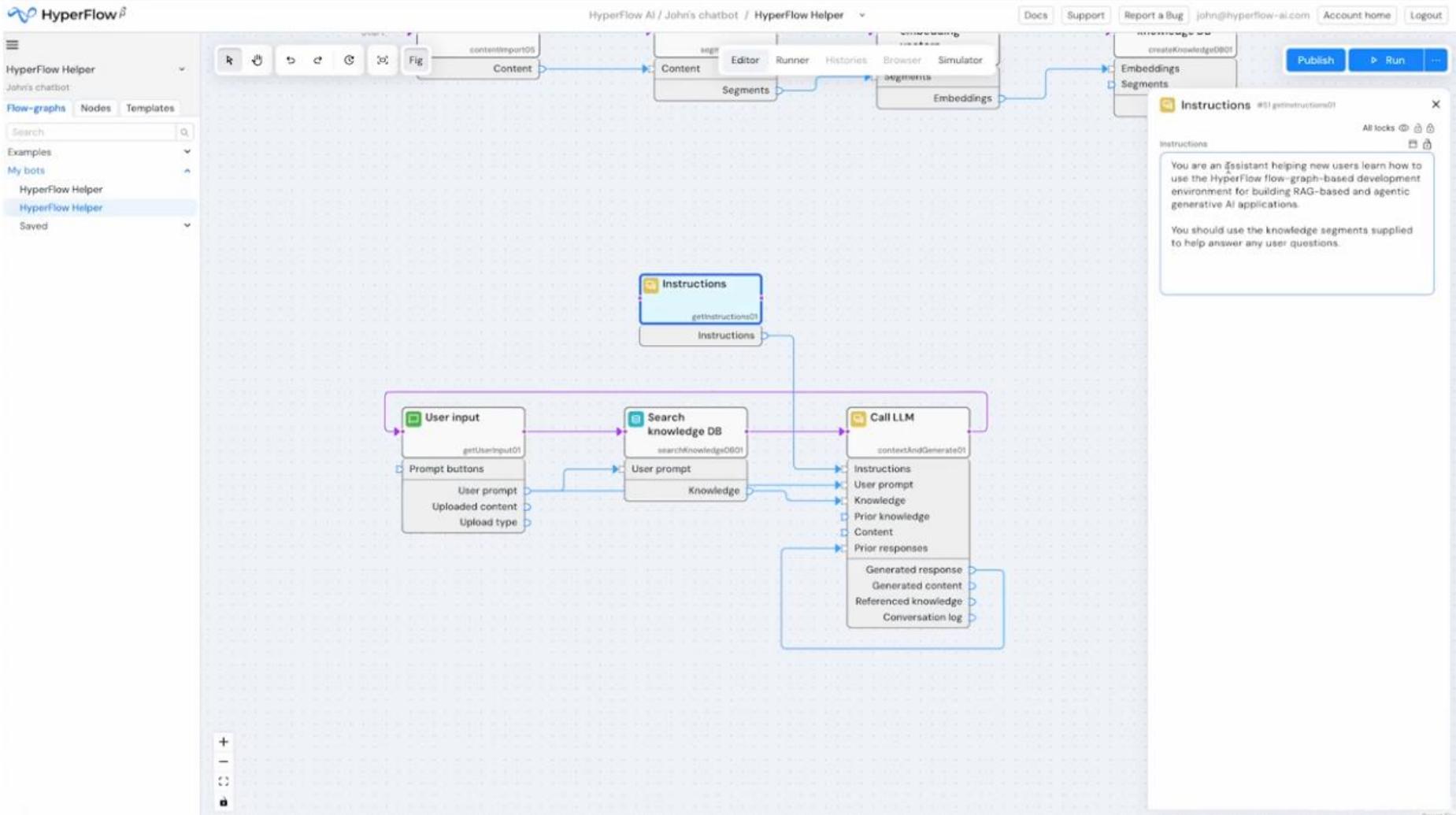
RAG 기반 지식 DB 및 챗봇 구축



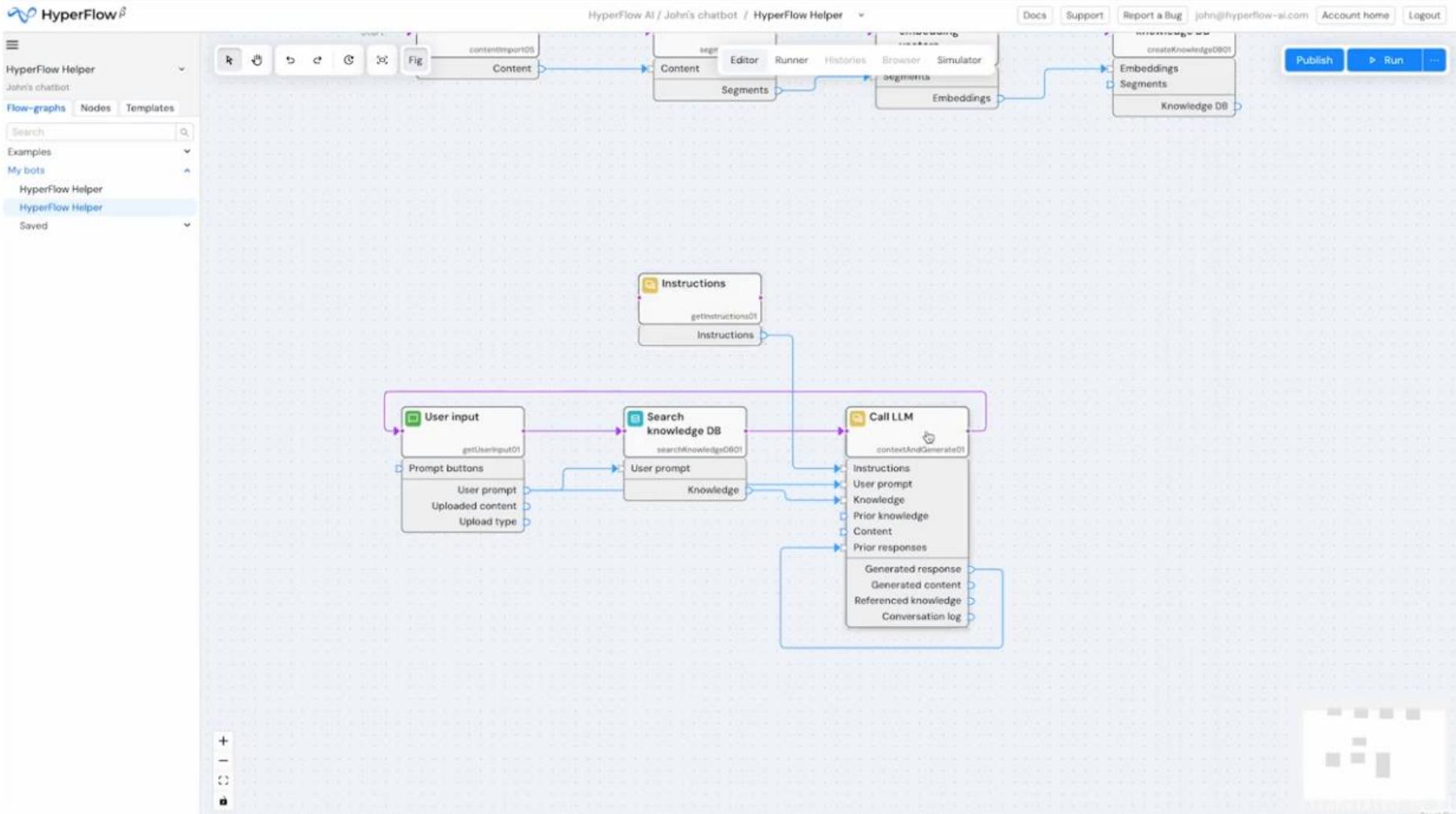
RAG 기반 지식 DB 및 챗봇 구축



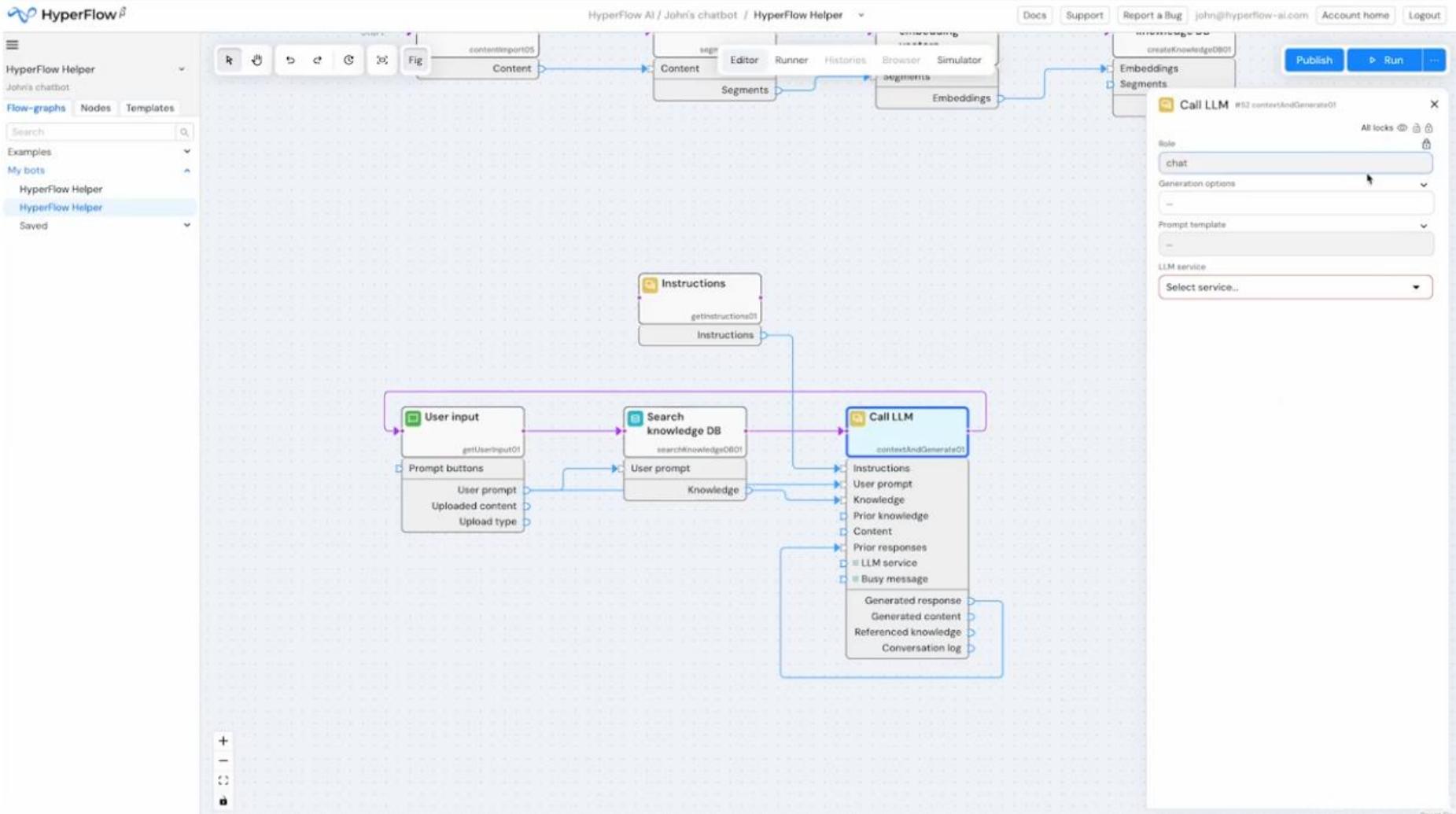
RAG 기반 지식 DB 및 챗봇 구축



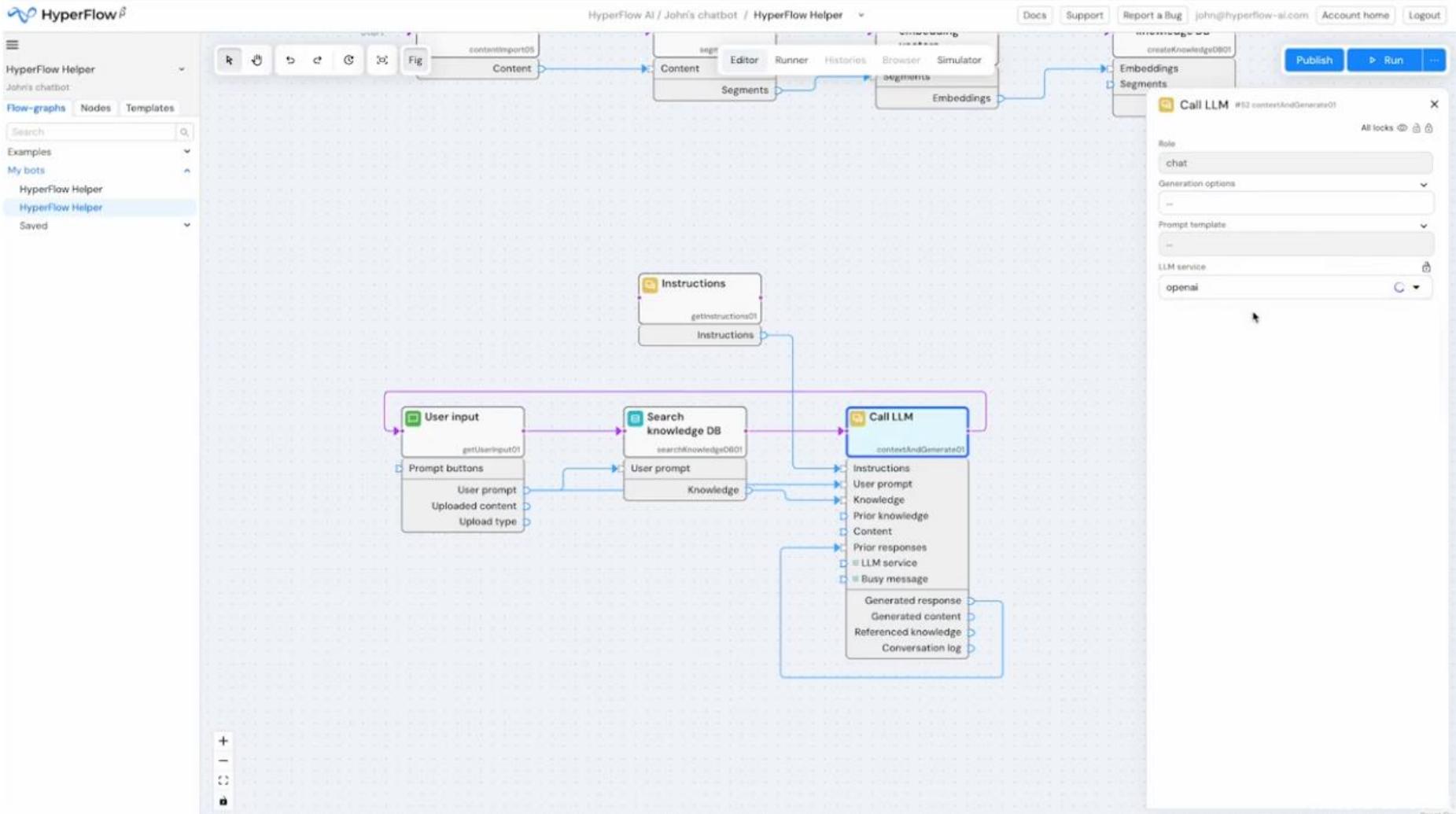
RAG 기반 지식 DB 및 챗봇 구축



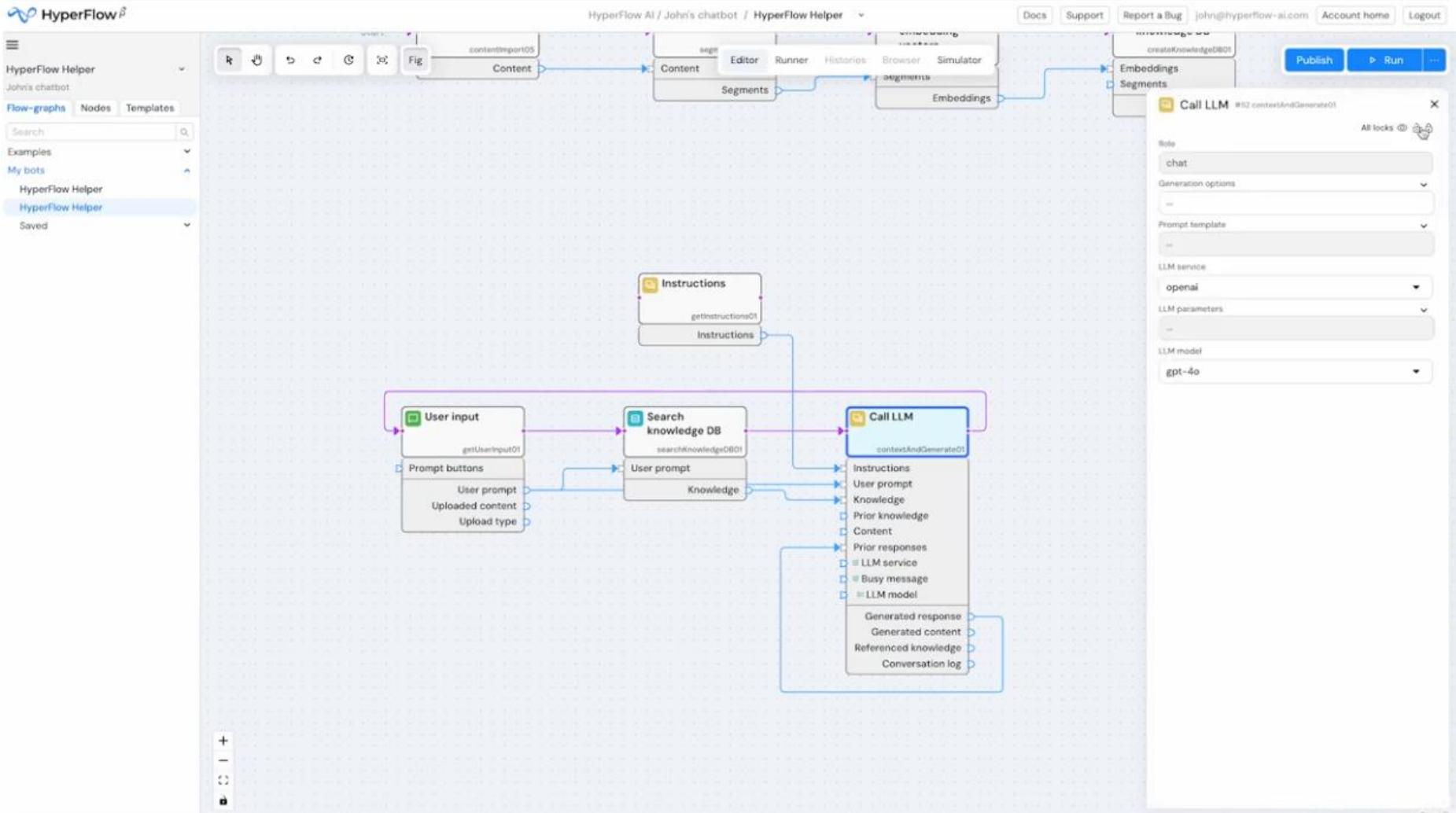
RAG 기반 지식 DB 및 챗봇 구축



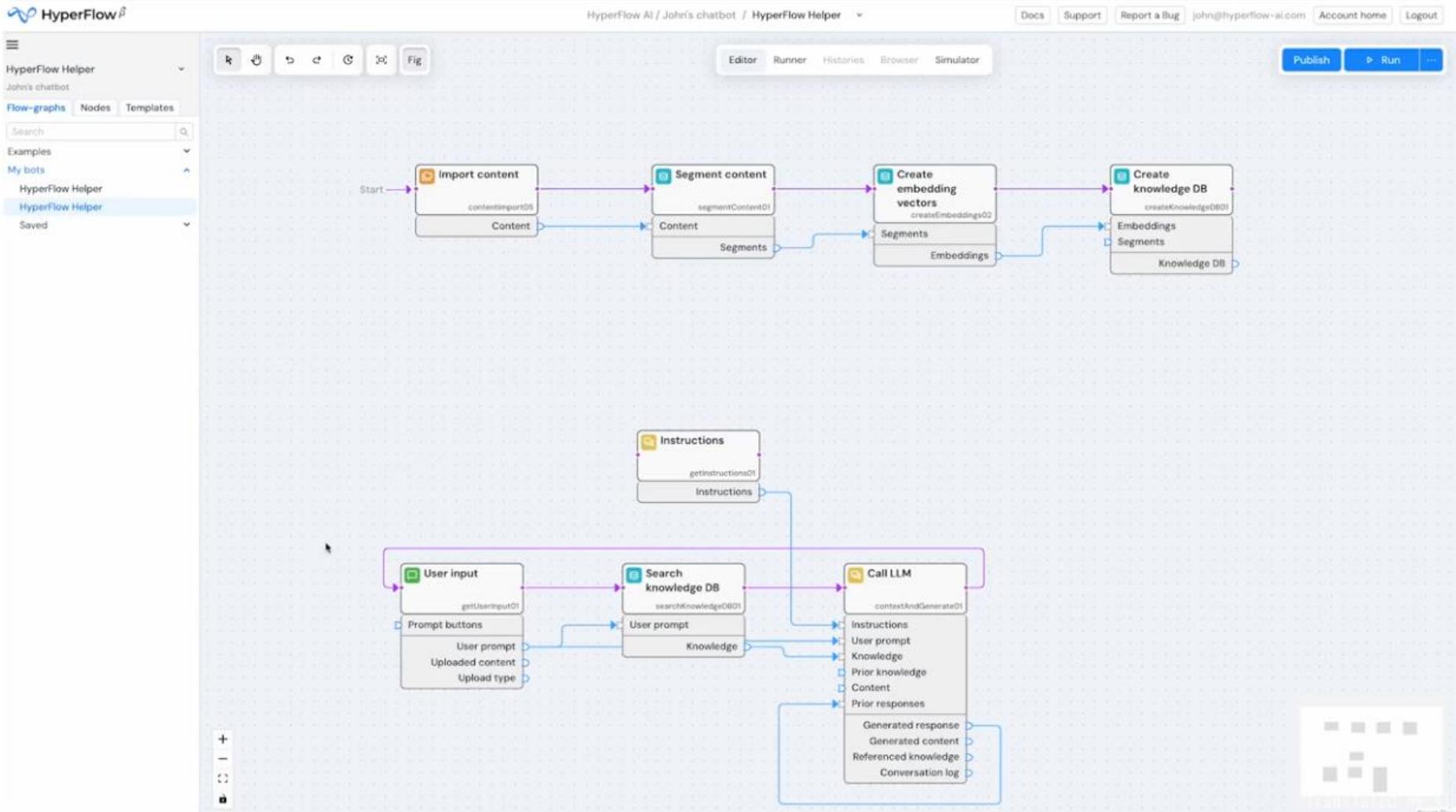
RAG 기반 지식 DB 및 챗봇 구축



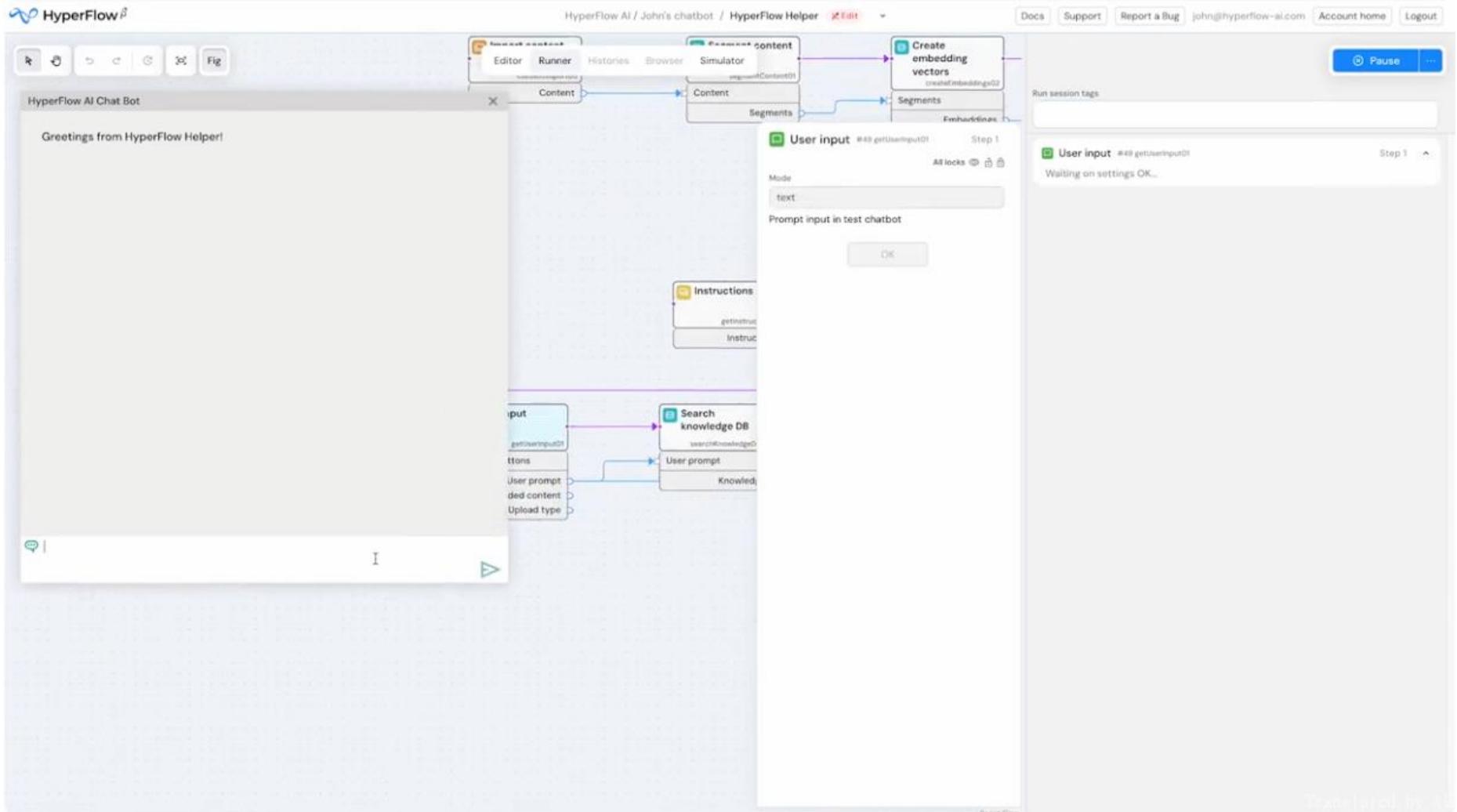
RAG 기반 지식 DB 및 챗봇 구축



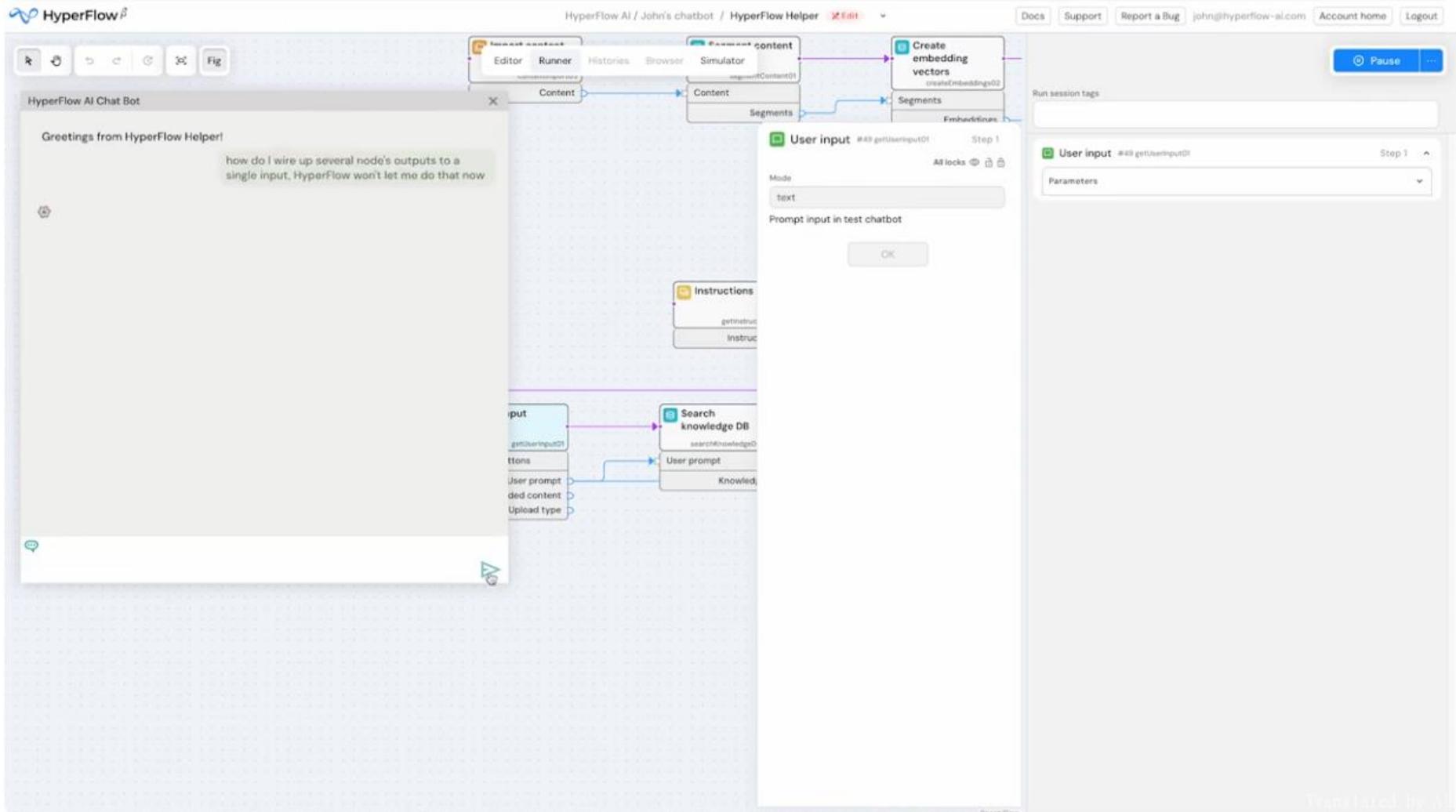
RAG 기반 지식 DB 및 챗봇 구축



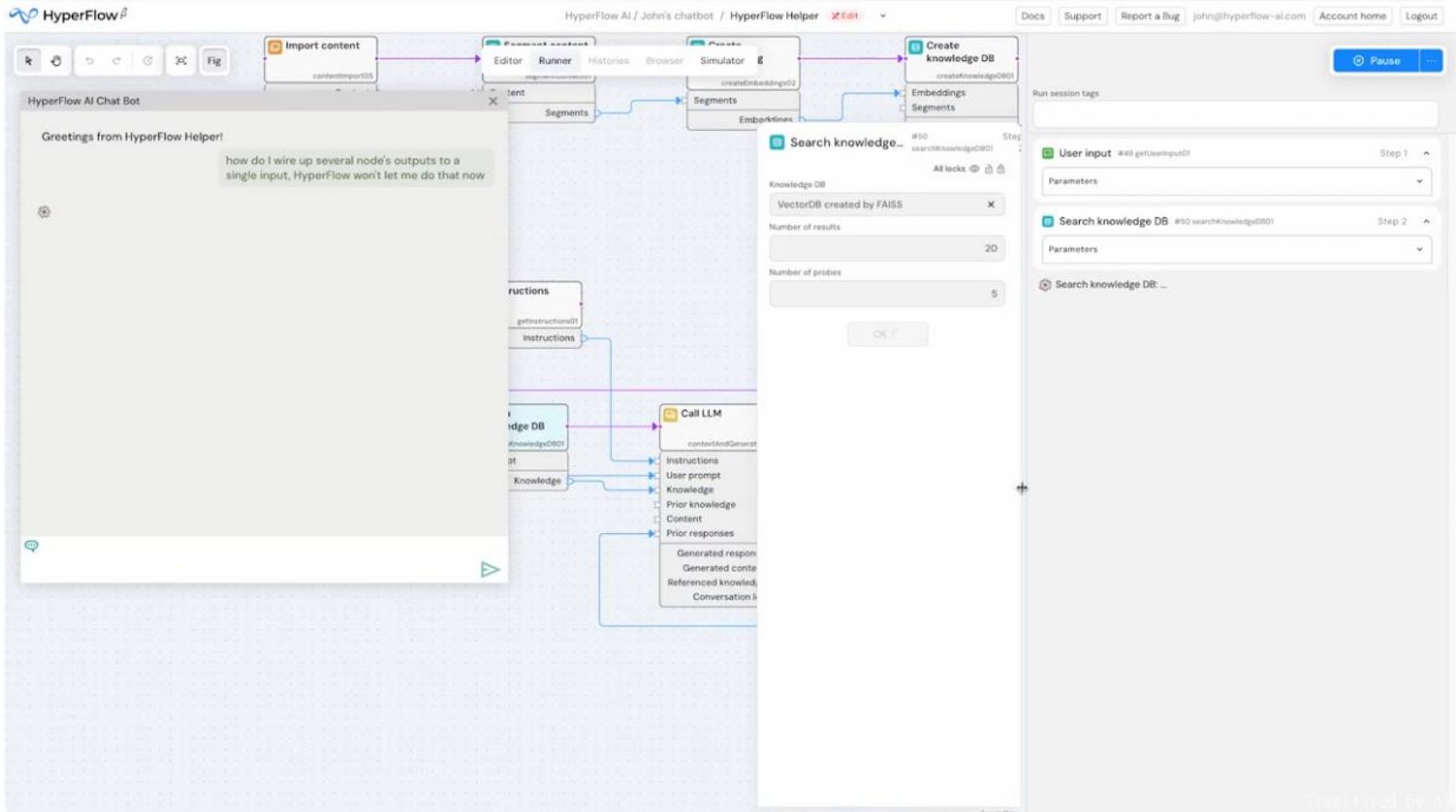
RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow ^β

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

Segment content → Create embeddings → Knowledge DB → Call LLM → Search knowledge DB → Instructions → Call LLM

Call LLM #52 contextAndGenerateQ1 Step 4

Role: chat
Generation options:
Prompt template:
LLM service: OpenAI

OK

Search knowledge DB #50 searchKnowledgeQ01 Step 2

Parameters:

Segmentation preview:

- #88 Nodes reference: Nodes Overview: Merge inputs: Accept a variable number of inputs ...
- #89 Merge Inputs: Merge Inputs Node: Data flow node group
- #90 Merge Inputs: Merge Inputs Node: Overview: The Merge Inputs node combines multiple ...
- #91 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Made with Drag image ...
- #92 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining d...
- #93 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining d...
- #94 Nodes reference: Nodes Overview: HyperFlow's capabilities reside largely in the nodes ...
- #95 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Workflow Patterns: Basic M...
- #96 Message to Host: Message to Host Node: Control flow node group
- #97 Merge Inputs: Merge Inputs Node: Data flow history: As a flow graph runs, each of a no...
- #98 Merge Inputs: Merge Inputs Node: Property| Type Description: ---> Input || any| P...|
- #99 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Key Features: Multiple data...
- #100 Chat Output: Wireable Inputs: Property| Type Description: ---> Input|| text/num...
- #101 Loop Controller: Loop Controller Node: Flow control node group
- #102 HyperFlow Beta Documentation / ... / Code Docs / User Input Made with Drag image to ...
- #103 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Data Type Selection Choic...
- #104 Chat Output: Chat Output Node: Description: This node enables merging and displayin...
- #105 User Input: User Input Node: Configuration Guidelines: Usage Examples: Text-only Mod...
- #106 Batch Content: Batch Content Node: Data flow node group
- #107 HyperFlow Beta Documentation: 4 more properties| Beta program welcome Release not ...

Metadata: Count: 30 Min Distance: 0.40580475330352783 Max Distance: 0.5299315462075454 Elapsed: 0.21s

Instructions #51 getInstructionsQ1 Step 3

Parameters:

Instructions:
You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications.
You should use the knowledge segments supplied to help answer any user questions.

Call LLM #52 contextAndGenerateQ1 Step 4

Waiting on settings OK...

React Flow

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow AI Chat Bot
Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

Segment content → Create Embedding 02 → Knowledge DB → Call LLM → Instructions → Call LLM → Call LLM

Call LLM #52 contextAndGenerate01 Step 4

- Role: chat
- Generation options:
- Prompt template:
- LLM service: OpenAI
- LLM parameters:
- LLM model: gpt-4o

Metadata: Count: 20, Min Distance: 0.40280475330352783, Max Distance: 0.529931543257644, Elapsed: 0.21s

Instructions #51 getInstructions01 Step 3

You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications.

You should use the knowledge segments supplied to help answer any user questions.

Call LLM #52 contextAndGenerate01 Step 4

Parameters

Composite prompt

Type	Content	Approx tokens	Source step
1. instructions	You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications.	46	
2. knowledge	--- **Knowledge Segments:** --- Segment ID: "000" - Content: Nodes Overview: Merge inputs: Accept a variable number of inputs merging them	1372	2
3. user	**User Query:** how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now	23	1

Call LLM: ...

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

HyperFlow AI Chat Bot

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID '009' for more details).

Instructions

getInstruction instruc

Input

getOverInput01

Buttons

User prompt

ded content

Upload type

Search knowledge DB

searchKnowledge0

User prompt

Knowned

content

Content

Segments

Create embedding vectors

createEmbedding02

Segments

Embedding

User input #49 get userInput01 Step 5

All locks

Mode: text

Prompt input in test chatbot

OK

Call LLM #52 content5AndGenerate01 Step 4

Parameters

Composite prompt

Type	Content	Approx. tokens	Source step
1. instructions	You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based agents;	46	
2. knowledge	... **Knowledge Segments** - ***Segment ID*** "000" - **Content** Nodes reference Nodes Overview Merge Inputs: Accept a variable number of inputs merging them	1372	2
3. user	**User Query:** how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now	23	1

Generated text

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID '009' for more details).

Metadata

Model: BH-4a-2024-08-06	Created: 2025-01-12 20:38	Elapsed: 3.63s	Prompt Tokens: 1471	Completion Tokens: 281
-------------------------	---------------------------	----------------	---------------------	------------------------

User input #49 get userInput01 Step 5

Waiting on settings OK...

React Flow



RAG 기반 지식 DB 및 챗봇 구축



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

HyperFlow AI Chat Bot

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Content → Content → Create embedding vectors → Segments → Segments → Embedding

User input → Step 5

Call LLM → Step 4

Instructions

Input → Search knowledge DB → User prompt

Metadata

Model: EFT-4a-2024-08-06 | Created: 2025-01-12 20:38 | Elapsed: 3.43s | Prompt Tokens: 1471 | Completion Tokens: 281

User input → Step 5

Waiting on settings OK...

```
graph LR; Content[Content] --> Create[Create embedding vectors]; Create --> Segments[Segments]; Segments --> Embedding[Embedding]; UserInput[User input] --> Step5[Step 5]; CallLLM[Call LLM] --> Step4[Step 4]; Instructions[Instructions] --> Input[Input]; Input --> SearchDB[Search knowledge DB]; SearchDB --> UserPrompt[User prompt];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Content → Content → Create embedding vectors → Embedding

Content → Segments → Segments

User input #49 getUserId01 Step 1

Mode: text

Prompt input in test chatbot

OK

Run session tags

User input #49 getUserId01 Step 2

Parameters

Search knowledge DB #50 searchKnowledgeDB01 Step 3

Parameters

Segmentation preview

#49 getUserId01
Nodes reference: Nodes Overview: Merge inputs:
Accept a variable number of inputs merging them into a single output.
Category: Data flow Internal name: mergeDataflows
title: Nodes reference url: https://docs.hyperflow-ai.com/Nodes-reference-158779882b5781a99ada0dd8230c3fb0d7pvs25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

#50 searchKnowledgeDB01
Nodes reference: Nodes Overview: Search knowledge DB:
Searches a knowledge database for relevant information based on user input.
Category: Data flow Internal name: searchKnowledgeDB
title: Nodes reference url: https://docs.hyperflow-ai.com/Nodes-reference-158779882b5781a99ada0dd8230c3fb0d7pvs25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

Instructions #51 getInstructions01 Step 4

Parameters

Instructions

You are an assistant helping new users learn how to use the HyperFlow flow...
search...based documentation environment for building AI-powered applications.

React Flow

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. **Add a Merge Inputs Node:** Drag and drop a Merge Inputs node to your workspace from the node library.
2. **Connect Your Nodes:** Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. **Configure the Node (Optional):** Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. **Attach to Target Node:** Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Instructions

getInstructions01

Input

Buttons

User prompt

ded content

Upload type

Search knowledge DB

searchKnowledgeDB01

User prompt

Known...

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

HyperFlow AI Chat Bot

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Instructions

User input #49 getUserInput01 Step 1

Mode: text

Prompt input in test chatbot

OK

User input #49 getUserInput01 Step 2

Parameters

Search knowledge DB #50 searchKnowledgeDB01 Step 2

Parameters

Segmentation preview

- aee d=05299315482575684
Nodes reference: Nodes Overview: Merge inputs:
Accept a variable number of inputs merging them into a single output.
Category: Data flow internal name: mergeDataflows
title: Nodes reference url: https://docs.hyperflow-ai.com/Nodes-reference-15877982b5781a99ada8d8290a3c19652f?pvs=25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team
- aee d=0526675224304192
Merge Inputs/Merge Inputs Node:
Data flow node group
title: Merge Inputs url: https://docs.hyperflow-ai.com/Merge-Inputs-c486de579c626402831b390a3c19652f?pvs=25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team
- + #002 Merge Inputs: Merge Inputs Node: Overview: The Merge Inputs node combines multiple...
- + #003 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Made with Drag Image...
- + #004 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Reference Made with Drag Image to reposition
- + #005 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining...
- + #006 Nodes reference: Nodes Overview: HyperFlow's capabilities reside largely in the nodes...
- + #007 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Workflow Patterns: Basic M...
- + #008 Message to Host: Message to Host Node: Control flow node group
- + #009 Merge Inputs: Merge Inputs Node: Data-flow history: As a flow graph runs, each of a no...
- + #010 Merge Inputs: Wireable Inputs: Property! Type! Description: --->--->---> Input [Any] Prop...
- + #011 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Key Features: Multiple data...
- + #012 Chart Output: Wireable Inputs: Property! Type! Description: --->--->---> input! text/num...
- + #013 Loop Controller: Loop Controller Node: Flow control node group
- + #014 HyperFlow Beta Documentation / ... / Code Docs / User Input Made with Drag Image to ...
- + #015 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Data Type Selection Choic...
- + #016 Chat Output: Chat Output Node: Description: This node enables merging and displayin...
- + #017 User Input: User Input Node: Configuration Guidelines: Usage Examples: Text-only Mod...
- + #018 Batch Content: Batch Content Node: Data flow node group
- + #019 HyperFlow Beta Documentation: 4 more properties Beta program welcome Release not...

Metadata

Count: 20	Min Distance: 0.40500475330352793	Max Distance: 0.8299315452875684	Avg: 0.21s
-----------	-----------------------------------	----------------------------------	------------

React Flow

```
graph LR; Content[Content] --> CreateEmbedding[Create embedding vectors]; CreateEmbedding --> Segments[Segments]; Segments --> UserInput[User input]; UserInput --> SearchKnowledge[Search knowledge DB]; SearchKnowledge --> Instructions[Instructions]
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

HyperFlow AI Chat Bot

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Instructions

getInstruction instruc

Input

getInput01 tions

User prompt ded content Upload type

Search knowledge DB

searchKnowledge

User prompt Knowned

Create embedding vectors

createEmbedding02

content

Content

Segments

Embedding

User input

#43 getUserInput01 Step 5

All locks

Mode text

Prompt input in test chatbot

OK

Run session tags

Segmentation preview

Nodes reference: Nodes Overview: Merge inputs:

Accept a variable number of inputs merging them into a single output. Category: Data flow Internal name: mergeDataflows

title: Nodes reference url: <https://docs.hyperflow-ai.com/Nodes-reference-158771982b0791a99daidd8230c3fbd7pvs+25> description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

- att: d:0:5266752243041992 Merge inputs: Merge Inputs Node:

Data flow node group

title: Merge inputs url: <https://docs.hyperflow-ai.com/Merge-Inputs-c46de579c62b402393b390a3c19652f?pvs+25> description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

- att: d:0:4964863348067424 Merge inputs: Merge Inputs Node: Overview:

The Merge Inputs node combines multiple input sources into a single output stream. It supports various data types and merge modes, allowing flexible combination of data flows within the graph.

This node is most-commonly used to allow multiple nodes to source an input data connector on another node. Input nodes, in the current release, are limited to single data source connections.

title: Merge inputs url: <https://docs.hyperflow-ai.com/Merge-Inputs-c46de579c62b402393b390a3c19652f?pvs+25> description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

+ #85 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Made with Drag Image ...
+ #84 HyperFlow Beta Documentation / Nodes reference Made with Drag Image to reposition
+ #85 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining d...
+ #86 Nodes reference: Nodes Overview: HyperFlow's capabilities reside largely in the nodes ...
+ #87 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Workflow Patterns: Basic M...
+ #88 Message to Host: Message to Host Node: Control flow node group
+ #89 Merge Inputs: Merge Inputs Node: Data-flow history: As a flow-graph runs, each of a no...
+ #88 Merge Inputs: Writable Inputs: Proprietary Type Description --->---> Input || any! Pr...
+ #81 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Key Features: Multiple data...
+ #83 Chart Output: Writable Inputs: Proprietary Type Description --->---> Input|| text/num...
+ #83 Loop Controller: Loop Controller Node: Flow control node group
+ #84 HyperFlow Beta Documentation / ... / Code Docs / User Input Made with Drag Image to ...
+ #85 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Data Type Selection Choos...
+ #84 Chat Output: Chat Output Node: Description: This node enables merging and displayin...
+ #87 User Input: User Input Node: Configuration Guidelines: Usage Examples: Text-only Mod...
+ #83 Batch Content: Batch Content Node: Data flow node group
+ #88 HyperFlow Beta Documentation: 4 more properties Beta program welcome Release net...

Metadata

Count: 30 Min Distance: 0.40590476330332783 Max Distance: 0.5299375452576484 Elapsed: 0.21s

React Flow

```
graph LR; A[User input] --> B[Create embedding vectors]; C[Content] --> D[Content]; D --> E[Create embedding vectors]; F[Segments] --> G[Segments]; G --> H[Create embedding vectors]; I[Search knowledge DB] --> J[Knowned]; K[Input] --> L[Search knowledge DB]; L --> J; M[User prompt] --> N[Knowned]; O[Content] --> P[Content]; P --> Q[Create embedding vectors]; R[Segments] --> S[Segments]; S --> T[Create embedding vectors]; U[User prompt] --> V[Knowned];
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Content Content Content

Create embedding vectors createEmbeddings02

Segments Segments Embedding

User input #43 getUserInput01 Step 5

All locks

Mode text

Prompt input in test chatbot

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Instructions

getInstruction instruc

Input

getUserInput01

Items

User prompt

ded content

Upload type

Search knowledge DB

searchKnowledge0

User prompt

Knowned

team

#02 d1048648653480674744 Merge Inputs: Merge Inputs Node: Overview: The Merge Inputs node combines multiple input sources into a single output stream. It supports various data types and merge modes, allowing flexible combination of data flows within the graph.

This node is most-commonly used to allow multiple nodes to source an input: data connector on another node. Input nodes, in the current release, are limited to single data source connections.

title: Merge Inputs url: https://docs.hyperflow-ai.com/Merge-Inputs-c46de579c82b4023931b390a3c19652f?pv=25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

#03 d1048665380477905273 Hyperflow Beta Documentation

/

/

/

Code Docs

/

Merge Inputs

Made with

Drag image to reposition

title: Merge Inputs url: https://docs.hyperflow-ai.com/Merge-Inputs-c46de579c82b4023931b390a3c19652f?pv=25 description: A new tool that blends your everyday work apps into one. It's the all-in-one workspace for you and your team

#044 hyperflow Beta Documentation / Nodes reference Made with Drag image to reposition

#055 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining d...

#066 Nodes reference: Nodes Overview: HyperFlow's capabilities reside largely in the nodes...

#077 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Workflow Patterns: Basic M...

#088 Message to Host: Message to Host Node: Control flow node group

#099 Merge Inputs: Merge Inputs Node: Data Flow history: As a flow graph runs, each of a no...

#100 Merge Inputs: Wireable Inputs: Propertyl Type Description --->---> Input | any! Pn...

#111 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Key Features: Multiple data...

#122 Chat Output: Wireable Inputs: Propertyl Type Description --->---> input! text/num...

#133 Loop Controller: Loop Controller Node: Flow control node group

#144 HyperFlow Beta Documentation / ... / Code Docs / User Input Made with Drag image to ...

#155 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Data Type Selection Choos...

#166 Chat Output: Chat Output Node: Description: This node enables merging and displayin...

#177 User Input: User Input Node: Configuration Guidelines: Usage Examples: Text-only Mod...

#188 Batch Content: Batch Content Node: Data flow node group

#199 HyperFlow Beta Documentation: 4 more properties Beta program welcome Release note...

Metadata

Count Min Distance Max Distance Elapsed

React Flow

```
graph LR; A[User input] --> B[Create embedding vectors]; C[Search knowledge DB]; D[User prompt]; E[Knowned]; F[Segments]; G[Segments]; H[Content]; I[Content]; J[Content]; K[Content]; L[Content]; M[Content]; N[Content]; O[Content]; P[Content]; Q[Content]; R[Content]; S[Content]; T[Content]; U[Content]; V[Content]; W[Content]; X[Content]; Y[Content]; Z[Content]; A --> B; B --> D; D --> C; C --> E; E --> F; F --> G; G --> H; H --> I; I --> J; J --> K; K --> L; L --> M; M --> N; N --> O; O --> P; P --> Q; Q --> R; R --> S; S --> T; T --> U; U --> V; V --> W; W --> X; X --> Y; Y --> Z; Z --> A
```

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Run session tags

Greetings from HyperFlow Helper!

how do I wire up several node's outputs to a single input, HyperFlow won't let me do that now

In HyperFlow, if you want to wire up several nodes' outputs to a single input, you should use the "Merge Inputs" node. This node is explicitly designed to combine multiple input sources into a single output stream. Here's how you can set it up:

1. Add a Merge Inputs Node: Drag and drop a Merge Inputs node to your workspace from the node library.
2. Connect Your Nodes: Wire the outputs of the nodes whose data you want to merge into the inputs of the Merge Inputs node. Since the Merge Inputs node can accept a variable number of inputs, you have the flexibility to attach multiple sources to it.
3. Configure the Node (Optional): Check and configure the merge modes or data types if necessary to ensure that the merge operation meets your requirements (Refer to Segment IDs "002" and "011" for configuration guidelines).
4. Attach to Target Node: Connect the output of the Merge Inputs node to the input port of the other node you want to receive the combined data. This newly combined stream will be accessible in the targeted node.

Using the Merge Inputs node allows you to overcome the limitation of single data source connections to an input in the current HyperFlow release. You can further control how to merge using options like interleave, merge history, and most-recent according to your data requirements (See Segment ID "009" for more details).

Content → Content → Create embedding vectors → Segments → Segments → Embedding → User input → Search knowledge DB → User prompt → Known → Instructions

Content → Content → Create embedding vectors → Segments → Segments → Embedding → User input → Search knowledge DB → User prompt → Known → Instructions

User input → Step 1

Search knowledge DB → Step 2

Segmentation preview

Instructions

Metadata

React Flow

HyperFlow AI Chat Bot

Editor Runner Histories Browser Simulator

Content

Content

content

createEmbeddings02

Segments

Segments

Embedding

User input

#01 getUserInput01

Mode: text

Prompt input in test chatbot

OK

Search knowledge DB

#02 searchknowledgeDB01

Parameters

Instructions

#03 getInstructions01

Parameters

Segmentation preview

#009 Nodes reference: Nodes Overview: Merge Inputs: Accept a variable number of inputs...
#010 Merge Inputs: Merge Inputs Node: Data flow node group
#011 Merge Inputs: Merge Inputs Node: Overview: The Merge Inputs node combines multiple...
#012 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Made with Drag image...
#013 HyperFlow Beta Documentation / ... / Code Docs / Merge Inputs Made with Drag image...
#014 HyperFlow Beta Documentation / ... / Nodes reference Made with Drag image to reposition...
#015 Merge Inputs: Merge Inputs Node: Configuration Guidelines: When to use: Combining d...
#016 Nodes reference: Nodes Overview: HyperFlow's capabilities reside largely in the nodes...
#017 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Workflow Patterns: Basic M...
#018 Message to Host: Message to Host Node: Control flow node group
#019 Merge Inputs: Merge Inputs Node: Data-flow history: As a fine-graph runs, each of a no...
#020 Merge Inputs: Merge Inputs Node: Data-flow history: As a fine-graph runs, each of a no...
#021 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Key Features: Multiple data...
#022 Chat Output: Wireable Inputs: Property|Type|Description: ----- Input ||| any|Num|
#023 Loop Controller: Loop Controller Node: Flow control node group
#014 HyperFlow Beta Documentation / ... / Code Docs / User Input Made with Drag image to ...
#015 Merge Inputs: Merge Inputs Node: Configuration Guidelines: Data Type Selection Chec...
#016 Chat Output: Chat Output Node: Description: This node enables merging and displayin...
#017 User Input: User Input Node: Configuration Guidelines: Usage Examples: Text-only Mod...
#018 Batch Content: Batch Content Node: Data flow node group
#019 HyperFlow Beta Documentation: 4 more properties Beta program welcome Release not...

Metadata

Count: 30 Min Distance: 0.409804793303952793 Max Distance: 0.8299910462675684 Elapsed: 0.21s

Instructions

#01 getInstructions01

Parameters

Instructions

You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications.

You should use the knowledge segments supplied to help answer any user questions.

Call 1/14 #01 searchknowledgeDB01

Call 2/14 #02 getInstructions01

Call 3/14 #03 getUserInput01

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content Segment content Create

Content

Instructions

You are an assistant helping new users learn how to use the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications.

You should use the knowledge segments supplied to help answer any user questions.

Instructions #51 getInstructions()

All locks

Instructions

Start

User input

getUserInput()

Prompt buttons

User prompt

Knowledge

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

React Flow

RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Helper

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Import content Segment content Create

Instructions

You are a helpful and informative assistant who guides users on using the HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications. You have access to a knowledge base containing information about HyperFlow.

When answering a user's question, adhere to the following guidelines:

- **Utilize Provided Knowledge:** Prioritize the provided knowledge segments to answer the user's questions about HyperFlow.
- **Concise Instructions:** Provide clear and brief step-by-step instructions.
- **UI Guidance:** Whenever relevant, include the location of features or tools within the HyperFlow user interface (e.g., menus, toolbars, dialog boxes).
- **Suggest Related Topics:** Offer recommendations for related topics or features that might be helpful to the user.
- **Insufficient Information:** If the provided knowledge segments do not contain enough information to answer the user's query, respond with "Sorry, I do not have enough information to answer that."
- **Reference Formatting:** When referring to knowledge segments, use the term "references" instead of "segments."
- **Markdown Formatting:** If the response contains tables or lists or equations, you must output the response in raw markdown, including \$\$ around equations, and \$ around inline math formulas, etc.
- **Output Knowledge Segment IDs:** At the end of your response, include a JSON structure listing the IDs of the knowledge segments used to answer the query. For example:
```json{ "segment\_ids": ["id1", "id2"] }```

User input

Prompt buttons

User prompt

Knowledge

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

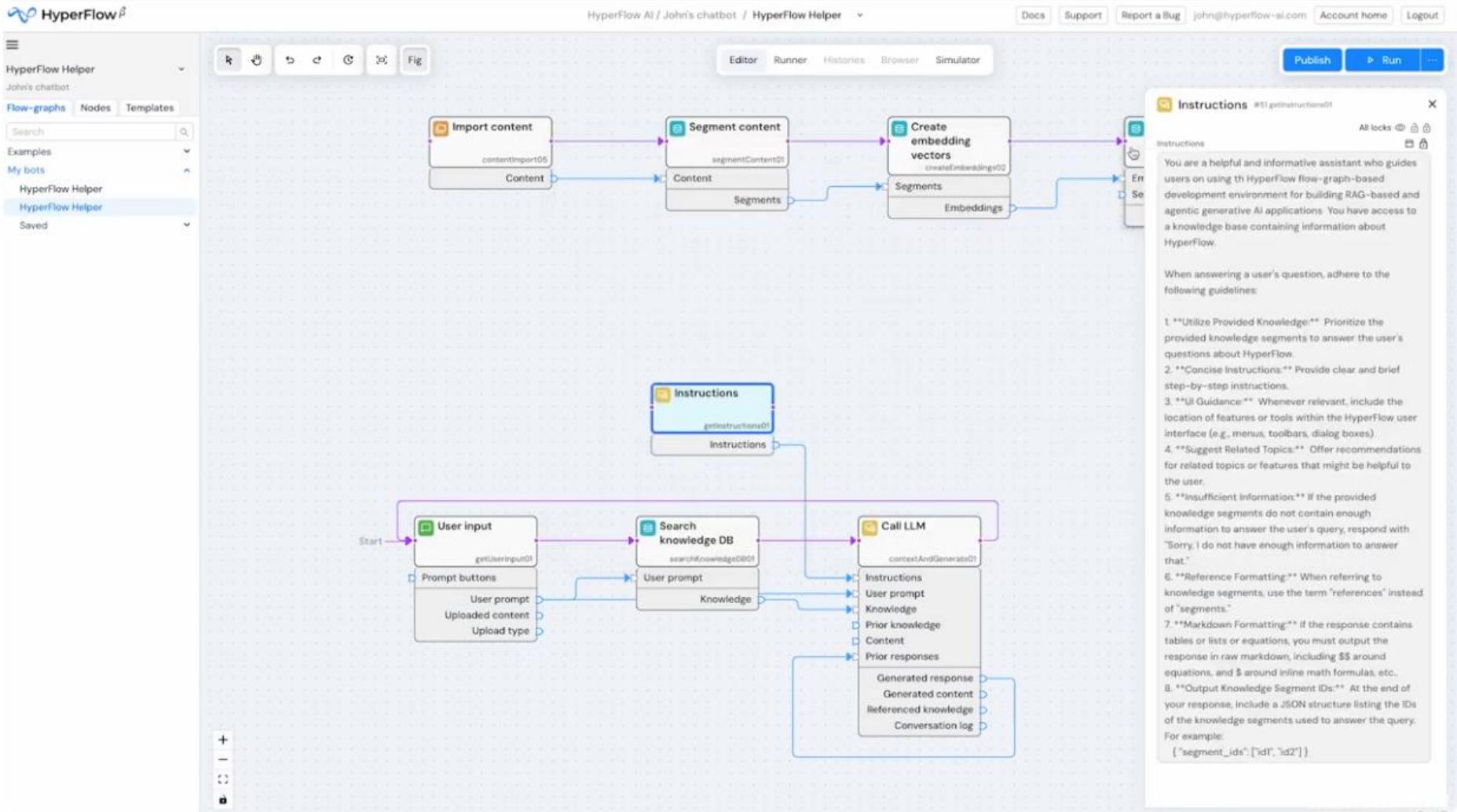
Generated response

Generated content

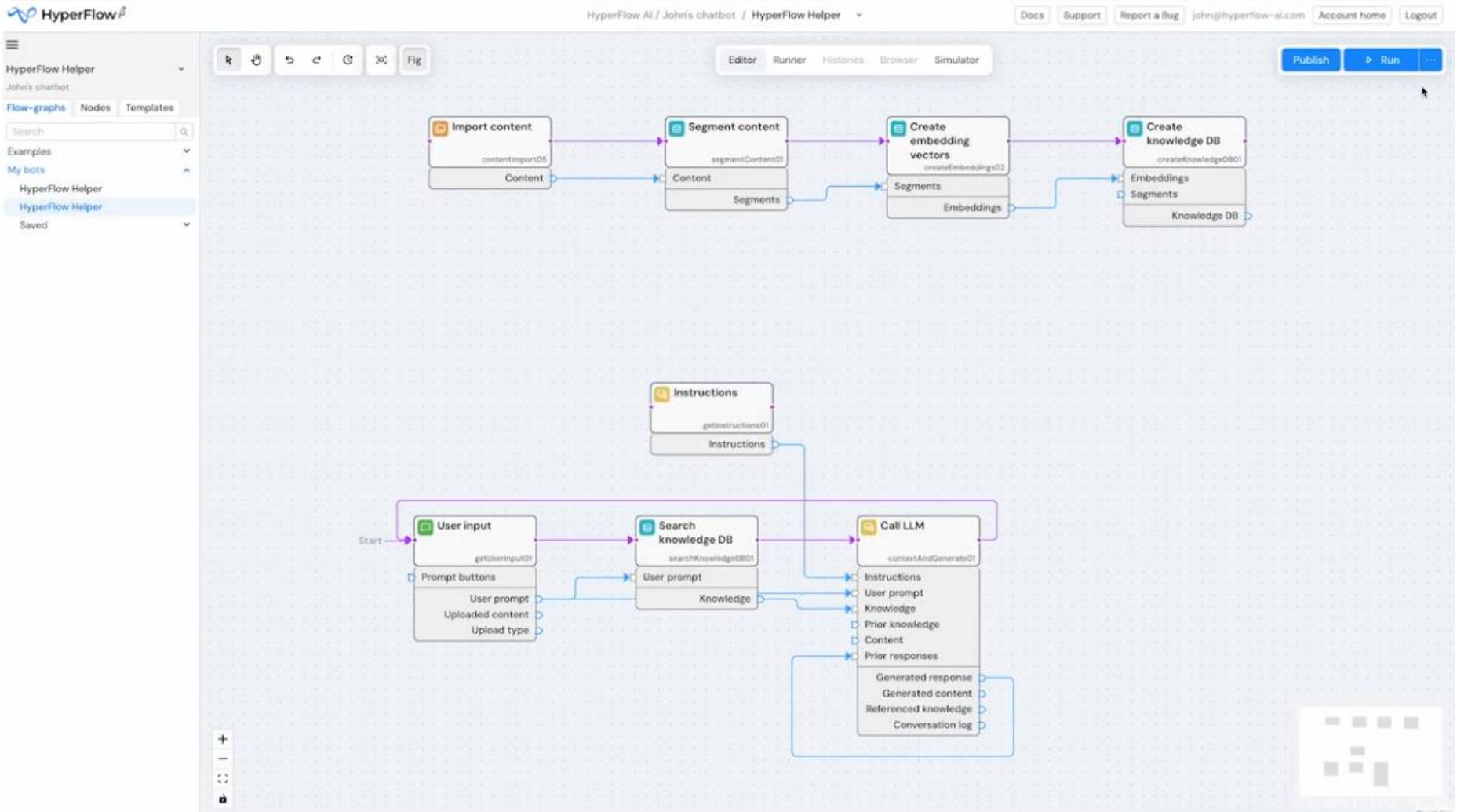
Referenced knowledge

Conversation log

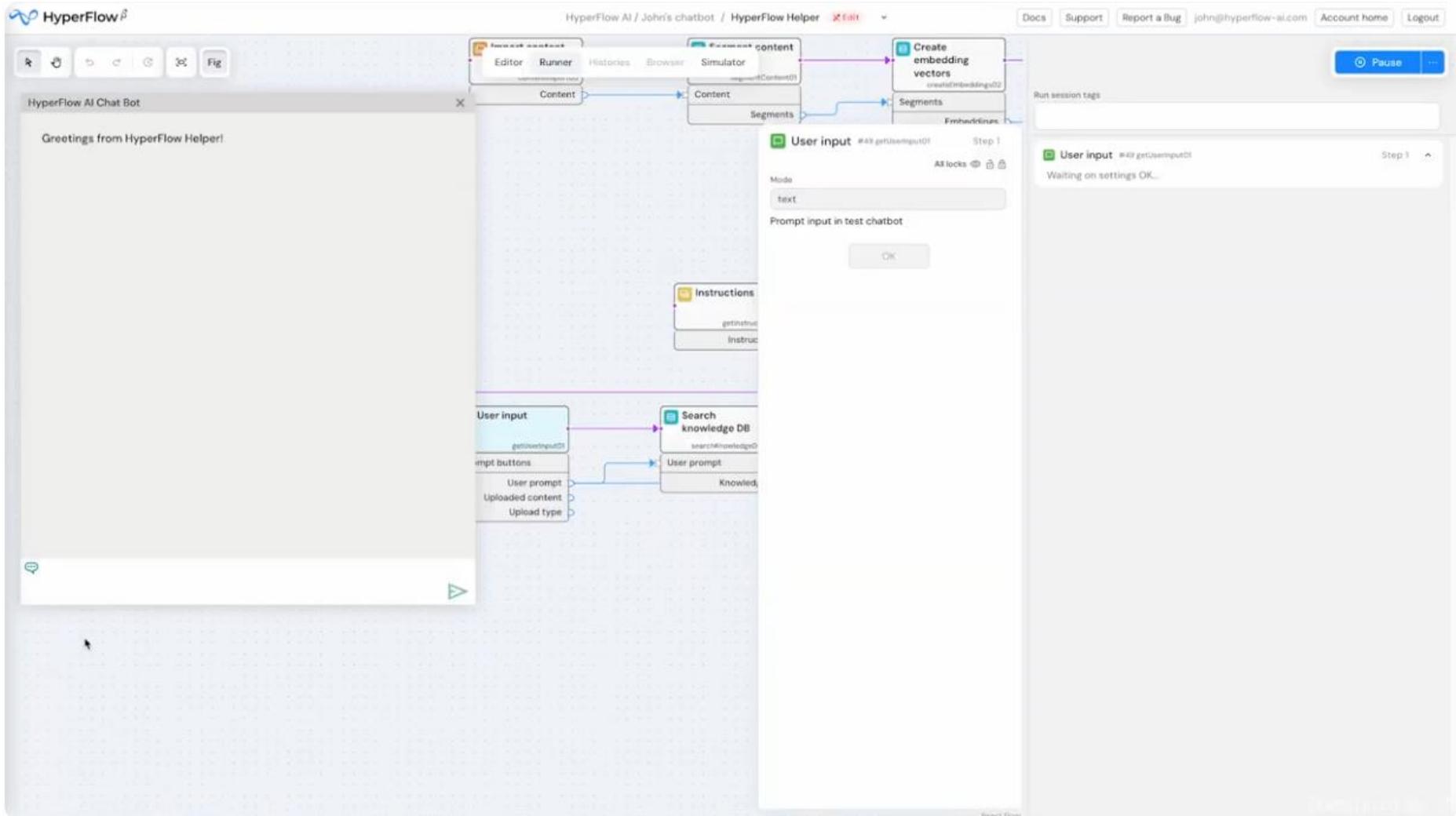
# RAG 기반 지식 DB 및 챗봇 구축



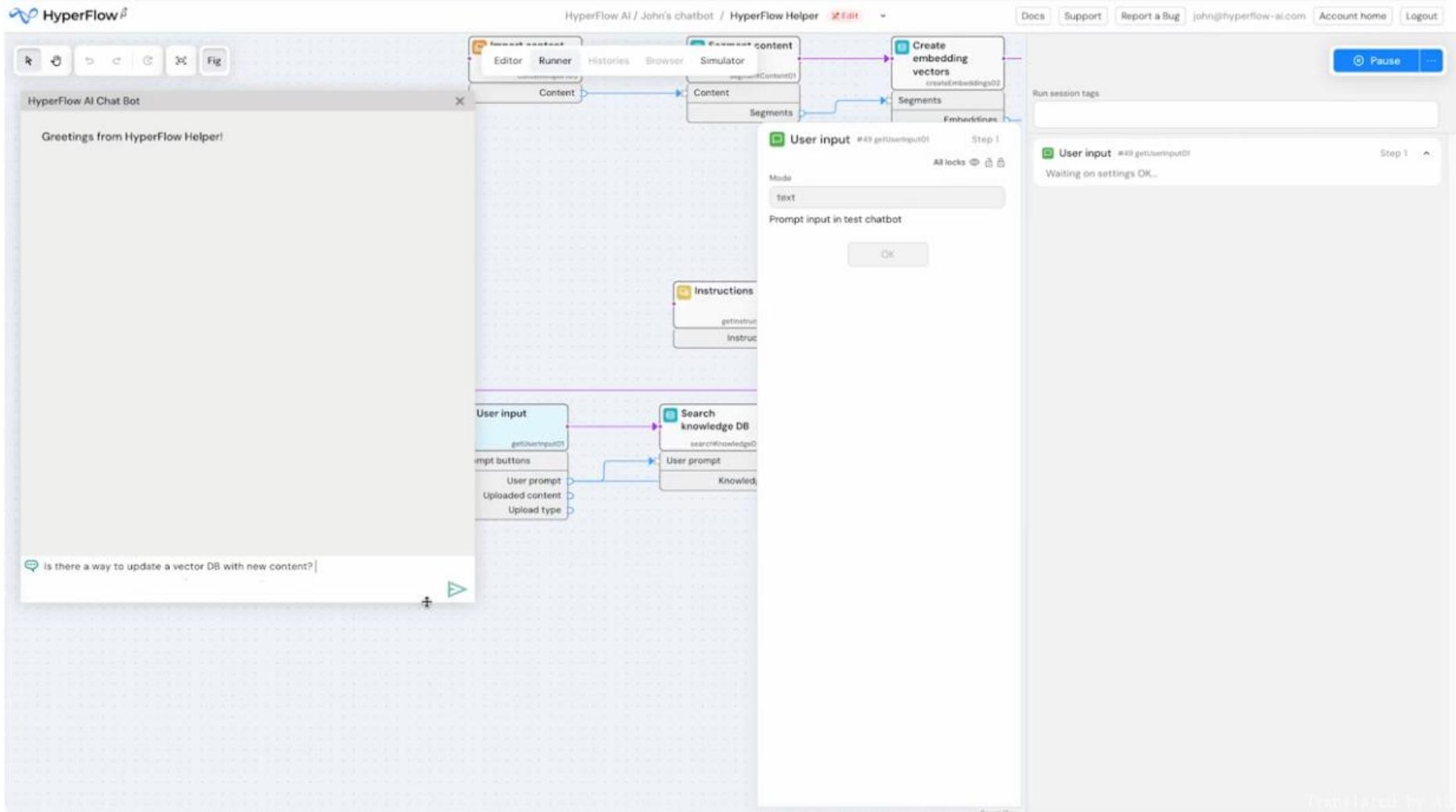
# RAG 기반 지식 DB 및 챗봇 구축



# RAG 기반 지식 DB 및 챗봇 구축



# RAG 기반 지식 DB 및 챗봇 구축



# RAG 기반 지식 DB 및 챗봇 구축



HyperFlow AI Chat Bot

Greetings from HyperFlow Helper!

Is there a way to update a vector DB with new content?

Segment content

Create embedding vectors

Segments

Embeddings

Instructions

All locks

Instructions

You are a helpful and informative assistant who guides users on using th HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications. You have access to a knowledge base containing information about Hyperflow.

When answering a user's question, adhere to the following guidelines:

- \*\*Utilize Provided Knowledge:\*\* Prioritize the provided knowledge segments to answer the user's questions about HyperFlow.
- \*\*Concise Instructions:\*\* Provide clear and brief step-by-step instructions.
- \*\*UI Guidance:\*\* Whenever relevant, include the location of features or tools within the HyperFlow user interface (e.g., menus, toolbars, dialog boxes).
- \*\*Suggest Related Topics:\*\* Offer recommendations for related topics or features that might be helpful to the user.
- \*\*Insufficient Information:\*\* If the provided knowledge segments do not contain enough information to answer the user's query, respond with "Sorry, I do not have enough information to answer that."
- \*\*Reference Formatting:\*\* When referring to knowledge segments, use the term "references" instead of "segments."
- \*\*Markdown Formatting:\*\* If the response contains tables or lists or equations, you must output the response in raw markdown, including \$\$ around equations, and \$ around inline math formulas, etc.
- \*\*Output Knowledge Segment IDs:\*\* At the end of your response, include a JSON structure listing the IDs of the knowledge segments used to answer the query. For example:

Call LLM

ch  
ledge DB

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Run session tags

Segmentation preview

Metadata

Count: 20 Min Distance: 0.3935467600822448 Max Distance: 0.5105642080307007 Elapsed: 0.21s

Instructions

Parameters

Instructions

You are a helpful and informative assistant who guides users on using th HyperFlow flow-graph-based development environment for building RAG-based and agentic generative AI applications. You have access to a knowledge base containing information about Hyperflow.

When answering a user's question, adhere to the following guidelines:

- \*\*Utilize Provided Knowledge:\*\* Prioritize the provided knowledge segments to answer the user's questions about HyperFlow.
- \*\*Concise Instructions:\*\* Provide clear and brief step-by-step instructions.
- \*\*UI Guidance:\*\* Whenever relevant, include the location of features or tools within the HyperFlow user interface (e.g., menus, toolbars, dialog boxes).

Instructions:

# RAG 기반 지식 DB 및 챗봇 구축



**HyperFlow**

HyperFlow AI / John's chatbot / HyperFlow Helper

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

Run session tags

Composite prompt

Type	Content	Approx tokens	Source step
instructions	You are a helpful and informative assistant who guides users on using the HyperFlow flow-graph-based development environment for building RAG-based and ... "Knowledge Segments" -> Segment ID: "000" -> "Content Annotation Batch Content Call LLM Chat Output	316	1
knowledge	... "User Query": "Is there a way to update a vector DB with new content?"	1469	2
user	"User Query": "Is there a way to update a vector DB with new content?"	14	1

Generated text

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

References:

Real-time Knowledge Injector

User input

Content

Create embedding vectors

Segments

Embeddings

Instructions

Search knowledge DB

User prompt

Knowned

User input

genUserInput01

impt buttons

User prompt

Uploaded content

Upload type

Mode: text

Prompt input in test chatbot

OK

React Flow

Waiting on settings OK...

# RAG 기반 지식 DB 및 챗봇 구축



**HyperFlow** beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

HyperFlow AI Chat Bot

Is there a way to update a vector DB with new content?

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the **Real-time Knowledge Injector** node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

References:

Real-time Knowledge Injector

Instructions

User input

Content

Create embedding vectors

Segments

Embedding

User input

Mode: text

Prompt input in test chatbot

OK

Composite prompt

Type	Content	Approx. tokens	Source step
1. instructions	You are a helpful and informative assistant who guides users on using the HyperFlow flow-graph-based development environment for building RAG-based and ... "Knowledge Segments" -> Segment ID: "000" -> "Content" -> Visualize Vector Database: Menu Add Content Annotation Batch Content Call LLM Chat Output	316	1
2. knowledge	... "User Query": "Is there a way to update a vector DB with new content?"	1469	2
3. user	"User Query": "Is there a way to update a vector DB with new content?"	14	1

Generated text

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the **Real-time Knowledge Injector** node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

{"segment\_ids": ["013", "015", "017"]}

References

Real-time Knowledge Injector

Metadata

Model: ERT-4a-2024-08-06 | Created: 2025-01-12 20:39 | Elapsed: 3.21s | Prompt Tokens: 1838 | Completion Tokens: 281

User input #49 getUserId01

Waiting on settings OK...

React Flow

# RAG 기반 지식 DB 및 챗봇 구축



HyperFlow

HyperFlow AI / John's chatbot / HyperFlow Helper Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

HyperFlow AI Chat Bot

Is there a way to update a vector DB with new content?

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

References:

Real-time Knowledge Injector

Content

User input

Segments

Create embedding vectors

Embedding

Instructions

User input

Search knowledge DB

User prompt

Known

Composite prompt

Type Content Approx tokens Source step

1. instructions You are a helpful and informative assistant who guides users on using the hyperflow flow-graph-based development environment for building RAG-based and ... "Knowledge Segments" - "Segment ID" "000" - "Content" "Visualize Vector Database: Menu Add Content Annotation Batch Content Call LLM Chat Output 316 1

2. knowledge ... "Knowledge Segments" - "Segment ID" "000" - "Content" "Visualize Vector Database: Menu Add Content Annotation Batch Content Call LLM Chat Output 1469 2

3. user ... "User Query" "Is there a way to update a vector DB with new content?" 14 1

Generated text

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

{ "segment\_ids": [ "013", "015", "017" ] }

References

Real-time Knowledge Injector

Metadata

Model: ERT-4a-2024-08-06 Created: 2025-01-12 20:39 Elapsed: 3.21s Prompt Tokens: 1838 Completion Tokens: 261

User input #49 get userInput01

Waiting on settings OK...

React Flow

<https://docs.hyperflow-ai.com/real-time-Knowledge-Injector-832afedcbb5347b49411567590e69b73?pv=25>

# RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta Docu... / ... / Code Docs / Real-time Knowledge Injector

Made with



## Real-time Knowledge Injector

- ☰ Menu
- ➕ Add Content
- \* Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- ↔ Data Range
- 槔 Data Set
- ↳ Data Transform
- ✓ Define Tool
- Import content
- ➡ Instructions
- ↙ Key-value Store
- Launch Parameters
- 🔗 LLM Tool Agent
- ⟳ Loop Controller

### Real-time Knowledge Injector Node

Knowledge bases > Knowledge DBs node group

#### Overview

The Real-time Knowledge Injector node is a powerful component that enables dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution, often useful in advanced interactive chatbots and assistants. It provides flexible configuration options for both creating new vector databases and updating existing ones with additional knowledge.

#### Description

This node serves as a unified solution for knowledge database management, capable of processing various input types (text, segments, or embeddings) and integrating them into a knowledge database. It can either create a new knowledge database with specified parameters or inject entries into an existing database, making it ideal for both initial setup and continuous, real-time knowledge expansion.

#### Configuration Guidelines

- Knowledge Database Setup
  - Choose between creating new DB or using existing
  - Configure knowledge DB parameters if creating new
- Embedding Configuration

#### Table of Contents

- Real-time Knowledge Injector Node
  - Overview
  - Description
  - Configuration Guidelines
    - Workflow Patterns
    - Key Features
    - When to use
    - Best Practices
    - New Database Creation
    - Database Update
    - Continuous Learning
  - Parameters
  - Creation Specific Parameters
  - Wireable Inputs
  - Wireable Outputs



# RAG 기반 지식 DB 및 챗봇 구축



**HyperFlow** beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

HyperFlow AI Chat Bot

Is there a way to update a vector DB with new content?

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

References:

[Real-time Knowledge Injector](#)

I want to implement a long-term store for the conversations between a user and an LLM.

Content → Content → Create embedding vectors → Segments → Embedding → User input → Step 5

User input Mode: text

Prompt input in test chatbot

OK

Composite prompt

Type	Content	Approx tokens	Source step
1. instructions	You are a helpful and informative assistant who guides users on using the hyperflow flow-graph-based development environment for building RAG-based and ... "Knowledge Segments" -> Segment ID: "000" -> "Content" -> Visualize Vector Database: Menu Add Content Annotation Batch Content Call LLM Chat Output	316	1
2. knowledge	"User Query": "Is there a way to update a vector DB with new content?"	1469	2
3. user	"User Query": "Is there a way to update a vector DB with new content?"	14	1

Generated text

Yes, you can update a vector database with new content using the "Real-time Knowledge Injector" node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

{"segment\_ids": ["013", "015", "017"]}

References

References:

[Real-time Knowledge Injector](#)

Metadata

Model	Created	Elapsed	Prompt Tokens	Completion Tokens
Eft-4a-2024-08-06	2025-01-12 20:39	3.21s	1828	261

User input #49 getInput01

Waiting on settings OK...

React Flow

# RAG 기반 지식 DB 및 챗봇 구축



HyperFlow AI Chat Bot  
Yes, you can update a vector database with new content using the 'Real-time Knowledge Injector' node in HyperFlow. This powerful component allows for dynamic creation and management of knowledge databases by injecting new segments during flow-graph execution. Here are the steps:

- Configure the Real-time Knowledge Injector node:
  - This node can be used to update existing vector databases with additional knowledge.
  - It is important to validate against existing database parameters to ensure consistency.
  - The node accumulates new knowledge while maintaining the database's integrity.
- Connect Inputs to Outputs:
  - For continuous learning and progressive knowledge accumulation, connect the output of the knowledge injector back to the input.
- Monitor Database Growth:
  - Ensure careful monitoring of how the database grows and changes with each new update.

These configurations make the Real-time Knowledge Injector node ideal for applications such as advanced interactive chatbots and assistants.

References:

[Real-time Knowledge Injector](#)

I want to implement a long-term store for the conversations between a user and an LLM

HyperFlow AI / John's chatbot / HyperFlow Helper

Editor Runner Histories Browser Simulator

Content

Segments

vectors

createEmbeddings001

Create knowledge DB

createknowledge001

Instructions

#01 getInstructions01

All locks

Step 7

Run session tags

References

Real-time Knowledge Injector

Metadata

Created: 2024-04-06 2025-01-12 20:29 Updated: 32s Prompt Tokens: 1626 Completion Tokens: 301

User input #01 getUserId01 Step 5

Parameters

Search knowledge DB #02 searchKnowledge001 Step 6

Parameters

Segmentation preview

Call LLM Configuration Guidelines: When to use: Generating AI content...

Instructions: Configuration Guidelines: When to Use: Setting LLM be...

Call LLM Tool Agent: Configuration Guidelines: Best Practices: Use appropriate...

Call LLM: Configuration Guidelines: Best Practices: Use appropriate...

Call LLM: Wireable Inputs: Property Type Description: Instructional text; Sy...

Call LLM: Configuration Guidelines: Workflow Patterns: Basic Chat Flow...

Call LLM: Configuration Guidelines: Role Setting Choose appropriate ge...

Call LLM: Wireable Outputs: Property Type Description: Tool output...

Call LLM: Configuration Guidelines: Key Features: Role-based prompt co...

Key-value Store: Configuration Guidelines: Key Features: Persistent...

Call LLM: Parameters: Property Description: Request...

Call LLM: Configuration Guidelines: Key Features: Role Define the cha...

Call LLM: Configuration Guidelines: Overview: The Call LLM node is a powerful component that co...

Custom AI App Control API Reference: API Reference: Start a session. When to use: Star...

LLM Tool Agent: Wireable Inputs: Property Type Description: Instructional t...

Key-value Store: Configuration Guidelines: When to use: Storing...

Custom AI App Control API Reference: API Reference: Progress a session. Key Features:...

Call LLM: Configuration Structure: The generation options consist of three main e...

Call LLM: Configuration Guidelines: Role Setting Choose appropriate ge...

Count: 39 Min Distance: 0.313748167892239 Max Distance: 0.498763751923192 Step 7

Instructions #01 getInstructions01

Waiting on input...

# RAG 기반 지식 DB 및 챗봇 구축



**HyperFlow** beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Pause

HyperFlow AI Chat Bot

I want to implement a long-term store for the conversations between a user and an LLM.

To implement a long-term store for conversations between a user and an LLM in HyperFlow, you can use the **Key-value Store Node**. This node is suitable for storing conversation logs persistently. Here's how you can do it:

1. Add the Key-value Store Node:
  - Use the menu to locate and add the Key-value Store Node to your flow-graph.
2. Configure the Node for Conversation Storage:
  - Utilize the key-based data retrieval and persistent storage features to log conversations.
  - You can store multi-type data, ensuring both text-based interactions and associated metadata are recorded.
3. Connect Conversation Data:
  - Use wireable inputs on the LLM Tool Agent Node to capture conversation logs.
  - Route this data into your Key-value Store Node to ensure that all interactions are logged.
4. Define Keys for Efficient Retrieval:
  - Use unique session IDs or timestamps as keys for logging and retrieving conversation data efficiently.
5. Regular Maintenance:
  - Consider implementing a mechanism to manage or clean outdated entries.

User input #49 getUserId01 Step 9

All locks 0

Prompt input in test chatbot

OK

Instructions

getInstruction instruc

User input

getUserId01

Input buttons

User prompt

Uploaded content

Upload type

Search knowledge DB

searchKnowledge

User prompt

Knowned

Content

content

Segments

Segments

Create embedding vectors

createEmbeddings02

Embedding

Run session tags

user

\*\*User Query:\*\* I want to implement a long-term store for the conversations between a user and an LLM.

Generated text

To implement a long-term store for conversations between a user and an LLM in HyperFlow, you can use the **Key-value Store Node**. This node is suitable for storing conversation logs persistently. Here's how you can do it:

1. Add the Key-value Store Node:
  - Use the menu to locate and add the Key-value Store Node to your flow-graph.
2. Configure the Node for Conversation Storage:
  - Utilize the key-based data retrieval and persistent storage features to log conversations.
  - You can store multi-type data, ensuring both text-based interactions and associated metadata are recorded.
3. Connect Conversation Data:
  - Use wireable inputs on the LLM Tool Agent Node to capture conversation logs.
  - Route this data into your Key-value Store Node to ensure that all interactions are logged.
4. Define Keys for Efficient Retrieval:
  - Use unique session IDs or timestamps as keys for logging and retrieving conversation data efficiently.
5. Regular Maintenance:
  - Consider implementing a mechanism to manage or clean outdated entries to ensure the store remains efficient.

By adopting this setup, conversation records are kept independent of runtime instances, allowing you to maintain continuity across sessions or usage instances.

{ "segment\_ids": [ "010", "010" ] }

References

References:

Key-value Store

Metadata

Model: EKI-4x-2024-08-06 | Created: 2025-01-12 20:31 | Elapsed: 3.93s | Prompt Tokens: 3329 | Completion Tokens: 263

User input #49 getUserId01 Step 9

Waiting on settings OK...

React Flow

# RAG 기반 지식 DB 및 챗봇 구축



HyperFlow Beta Docu... / ... / Code Docs / Key-value Store Made with

# HyperFlow

## Key-value Store

Table of Contents

- Key-value Store Node
  - Overview
  - Description
    - Storing structured data as JSON Objects
  - Configuration Guidelines
    - Workflow Patterns
    - Key Features
    - When to use
    - Best Practices
  - Data Types
  - Parameters
  - Wireable Inputs
  - Wireable Outputs

☰ Menu

- Add Content
- \* Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- Data Range
- Data Set
- Data Transform
- Define Tool
- Import content
- Instructions
- Key-value Store
- Launch Parameters
- LLM Tool Agent
- Loop Controller

### Key-value Store Node

Data flow node group

#### Overview

The Key-value Store node provides persistent data storage capabilities within flow-graphs. It enables storing and retrieving any type of data using unique keys, functioning as a cold storage solution that persists within and across flow-graph executions and is shared between flow-graphs in the same HyperFlow project.

#### Description

This node implements a **key-value** storage system that can be accessed throughout the flow-graph. It supports basic operations like inserting, retrieving, and removing data entries using unique keys. The storage persists between runs, and so can be used to implement various forms of long-term memory for the flow-graphs within a project, and can be integrated with other nodes, particularly useful with Python code nodes.

#### Storing structured data as JSON Objects

One particularly powerful use of the **Key-value store** node is to store JSON Object values, which are complex objects like arrays, lists & dictionaries that can be represented as valid JSON strings. Several other node types in HyperFlow, such as Launch Parameters, Data Transform and Message to Host, produce or consume JSON object values, and being able to retain them in a key-value store enables very sophisticated work-flows to be implemented in HyperFlow.

#### Configuration Guidelines



# RAG 기반 지식 DB 및 챗봇 구축



**HyperFlow** Beta

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Docs Support Report a Bug john@hyperflow-ai.com Account home Logout

Content → Content → Create embedding vectors

User input → Step 9 → OK

Instructions → User input → Search knowledge DB

Generated text

To implement a long-term store for conversations between a user and an LLM in HyperFlow, you can use the **Key-value Store Node**. This node is suitable for storing conversation logs persistently. Here's how you can do it:

1. Add the Key-value Store Node:
  - Use the menu to locate and add the Key-value Store Node to your flow-graph.
2. Configure the Node for Conversation Storage:
  - Utilize the key-based data retrieval and persistent storage features to log conversations.
  - You can store multi-type data, ensuring both text-based interactions and associated metadata are recorded.
3. Connect Conversation Data:
  - Use wireable inputs on the LLM Tool Agent Node to capture conversation logs.
  - Route this data into your Key-value Store Node to ensure that all interactions are logged.
4. Define Keys for Efficient Retrieval:
  - Use unique session IDs or timestamps as keys for logging and retrieving conversation data efficiently.
5. Regular Maintenance:
  - Consider implementing a mechanism to manage or clean outdated entries to ensure the store remains efficient.

By adopting this setup, conversation records are kept independent of runtime instances, allowing you to maintain continuity across sessions or usage instances.

References:

[Key-value Store](#)

React Flow

Model: Eft-4a-2024-08-06 | Created: 2025-01-12 20:31 | Elapsed: 3.99s | Prompt Tokens: 3328 | Completion Tokens: 263

User input #49 getUserId01

Waiting on settings OK...

```
graph LR; Content[Content] --> Content2[Content]; Content2 --> Embedding[Create embedding vectors]; UserInput[User input] --> Step9[Step 9]; Step9 --> OK[OK]; Instructions[Instructions] --> UserInput; UserInput --> Search[Search knowledge DB]; UserInput --> Knowledge[Knowledge];
```

# Publishing a production flow-graph

## Publishing a production flow-graph

- Embeddable chatbot
- Custom AI app using the Control API

# Publishing a production flow-graph

- Versioning & release-management
- Pre-release testing
- Release activation
- Hot-release patches while still live

# 프로덕션 AI 앱을 위한 플로우그래프 배포



## Publishing a production flow-graph

- ▶ Embeddable chatbot
- ▶ Custom AI app using the Control API
- ▶ Versioning & release-management
- ▶ Pre-release testing
- ▶ Release activation
- ▶ Hot-release patches while still live

### 플로우그래프 제품 배포

- 내장형 챗봇
- 제어 API를 사용하는 맞춤형 AI 앱
- 버전 관리 및 릴리스 관리
- 출시 전 테스트
- 릴리스 활성화
- 아직 작동 중인 핫-릴리스 패치



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / --

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

Flow-graphs Nodes Templates

Search

John's chatbot

Demo bots Examples My bots

Flow-graphs

React File

The screenshot shows the HyperFlow AI platform interface. At the top, there's a navigation bar with the logo, user name 'John's chatbot', and links for Helper, Docs, Report a Bug, email (john@hyperflow-ai.com), Account home, and Logout. Below the navigation is a toolbar with icons for back, forward, search, and a 'Fig' button. A tab bar includes Editor, Runner, Histories, Browser, and Simulator. On the left, a sidebar lists 'John's chatbot' under 'Flow-graphs', along with 'Nodes' and 'Templates'. It also has sections for 'Demo bots', 'Examples', and 'My bots', each with a dropdown arrow. A large central area is labeled 'Flow-graphs' and contains a blank white canvas. At the bottom left, there are zoom and orientation control buttons (+, -, ⌂, ⌂). At the bottom right, there's a 'React File' button.

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / --

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator

John's chatbot

Flow-graphs Nodes Templates

Search

Demo bots Examples My bots

React Flow

The screenshot shows the HyperFlow AI platform interface. At the top, there's a navigation bar with the title "HyperFlow AI / John's chatbot", user information (john@hyperflow-ai.com), and links for Helper, Docs, Report a Bug, Account home, and Logout. Below the navigation is a toolbar with icons for Editor, Runner, Histories, Browser, and Simulator. On the left, a sidebar displays "John's chatbot" and tabs for Flow-graphs, Nodes, and Templates. It also includes sections for Demo bots, Examples, and My bots, along with a search bar and a "React Flow" button at the bottom right.

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow <sup>β</sup>

HyperFlow AI / John's chatbot / --

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

Flow-graphs Nodes Templates help

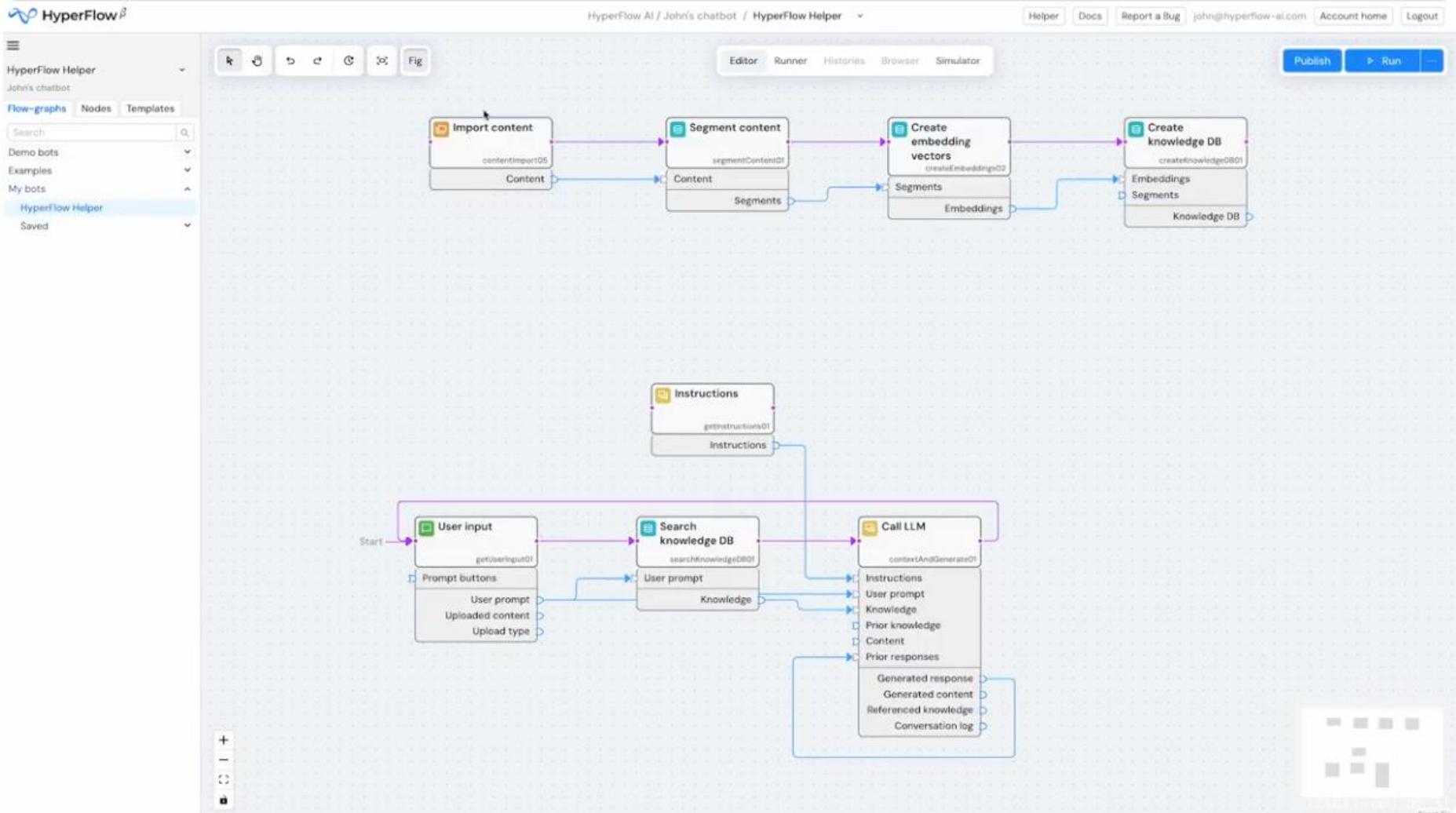
My bots / HyperFlow Helper

My bots.Saved / HyperFlow Helper KB build...

React File

The screenshot shows the HyperFlow AI platform interface. At the top, there's a navigation bar with the logo, user info (john@hyperflow-ai.com), and account links. Below it is a toolbar with various icons. The main area has tabs for 'Flow-graphs', 'Nodes', and 'Templates', with 'Flow-graphs' currently selected. A search bar is also present. On the left, a sidebar lists saved bots like 'My bots / HyperFlow Helper' and 'My bots.Saved / HyperFlow Helper KB build...'. The central workspace is a large, empty area for building flow graphs, with a vertical toolbar on the left edge containing zoom and orientation controls.

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Documentation

Made with

## HyperFlow Beta Documentation

⌄ 5 more properties

- [Beta program welcome](#)
- [Release notes](#)
- [hyperflow-ai.com/hyperflow - the HyperFlow IDE](#)
- [HyperFlow-AI community support on Discord](#)
- [HyperFlow-AI community support on KakaoTalk](#)
- [HyperFlow-AI tutorial video channel on Youtube](#)

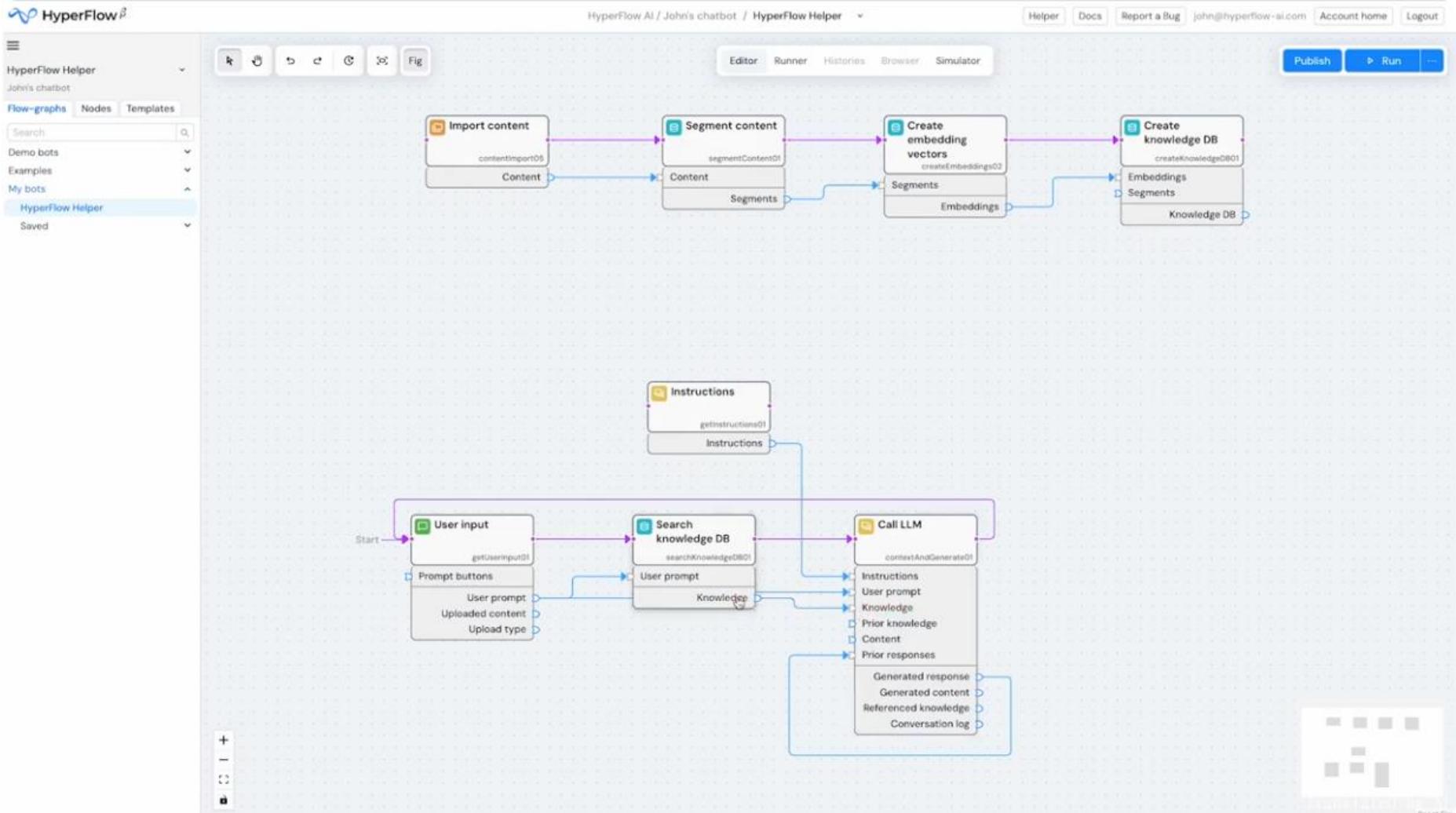
---

- [Nodes reference](#)
- [Services reference](#)
- [Data types](#)
- [Custom AI App Control API Reference](#)

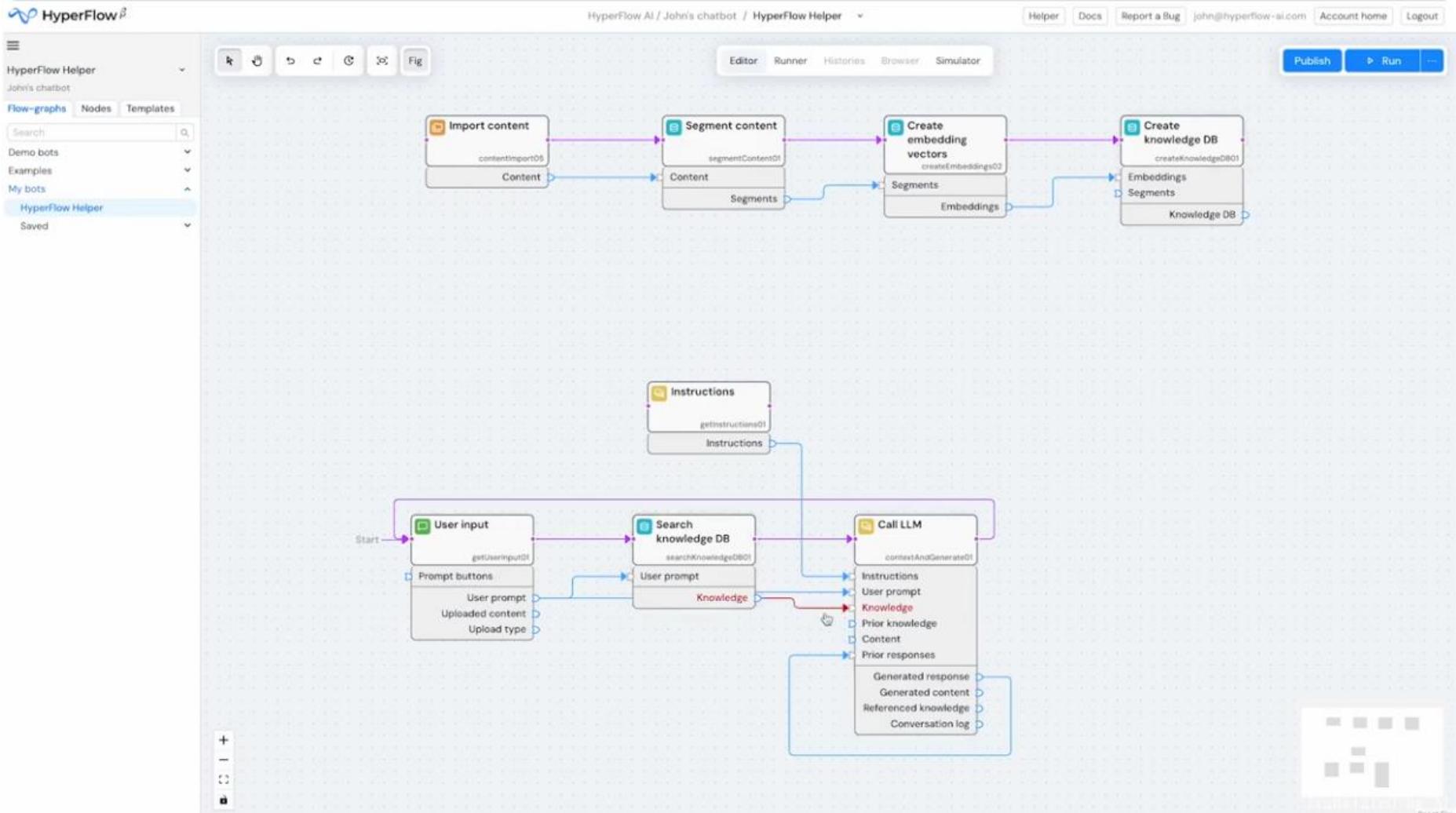
I



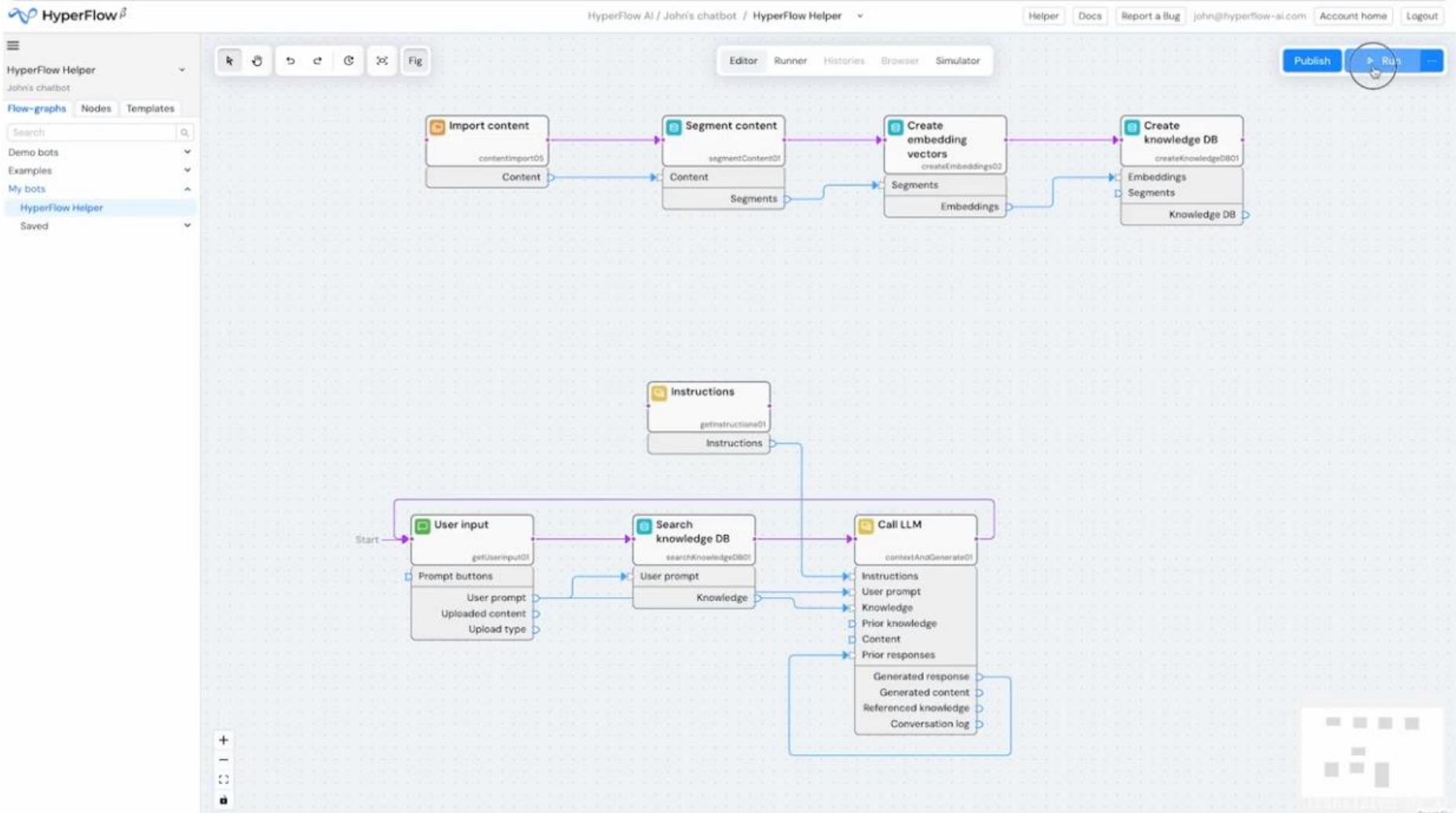
# 프로덕션 AI 앱을 위한 플로우그래프 배포



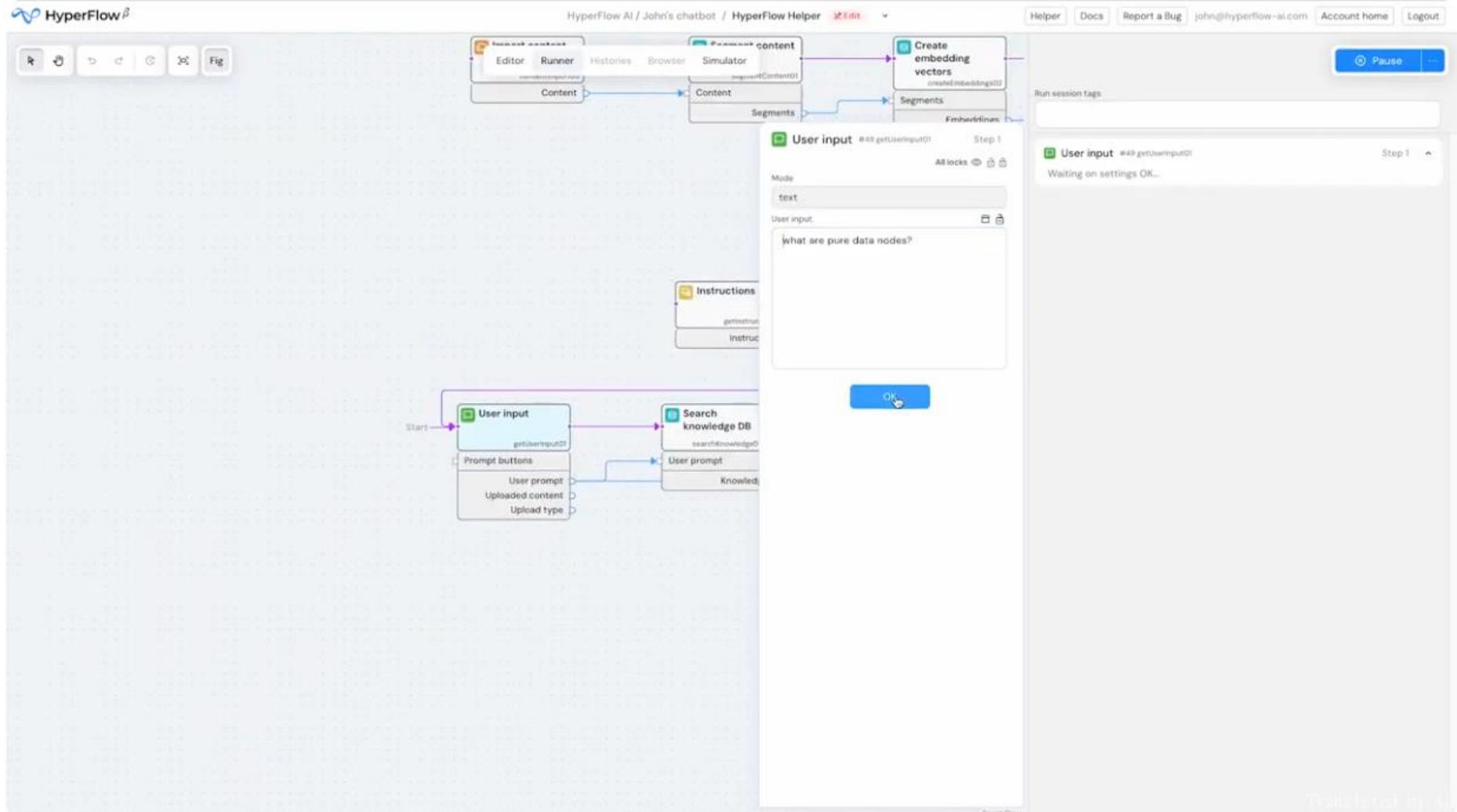
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow <sup>β</sup>

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Pause

Run session tags

Instructions:  
3. \*\*UI Guidance:\*\* Whenever relevant, include the location of features or tools within the HyperFlow user interface (e.g., menus, toolbars, dialog boxes).

Call LLM #62 contentNodeGenerate01 Step 4

Parameters

Composite prompt

Type Content

- Instructions You are a helpful and informative assistant who guides users on using the HyperFlow based development environment for building RAG-based and agentic generative AI. You have access to a knowledge base containing information about HyperFlow. When a user asks a question, you can search the knowledge base and provide an answer.
- knowledge ... \*\*Knowledge Segments\*\* - \*\*Segment ID\*\* "000" - \*\*Content\*\* Pure Data Nodes support sophisticated data flow - Nodes can run on demand when data nodes don't require process flow connections - System automatically manages segments.
- user \*\*User Query\*\* what are pure data nodes?

Generated text

Pure Data Nodes in HyperFlow refer to nodes that operate without requiring process flow connections. These nodes are triggered and run on demand whenever data is needed in a workflow. This approach simplifies wiring for data-driven workflows by allowing the system to automatically manage the execution order. Pure Data Nodes provide flexibility by supporting recursive data flow patterns, enabling sophisticated data flow management within HyperFlow. Examples include nodes like "Instructions" that run only when their data is pulled in the flow, rather than explicitly defining their process execution order.

For more in-depth understanding, you might want to explore the "Data Flow Configuration" and "Flow-Graph Components and Construction" topics in the HyperFlow documentation, which detail how data flows and nodes are arranged within a flow-graph.

```
{"segment_ids": ["000", "001"]}
```

References

References:

Metadata

Model: Eft-4a-2024-08-04	Created: 2025-05-21 15:07	Elapsed: 3.5ds	Prompt Tokens: 3218	Completion Tokens: 161
--------------------------	---------------------------	----------------	---------------------	------------------------

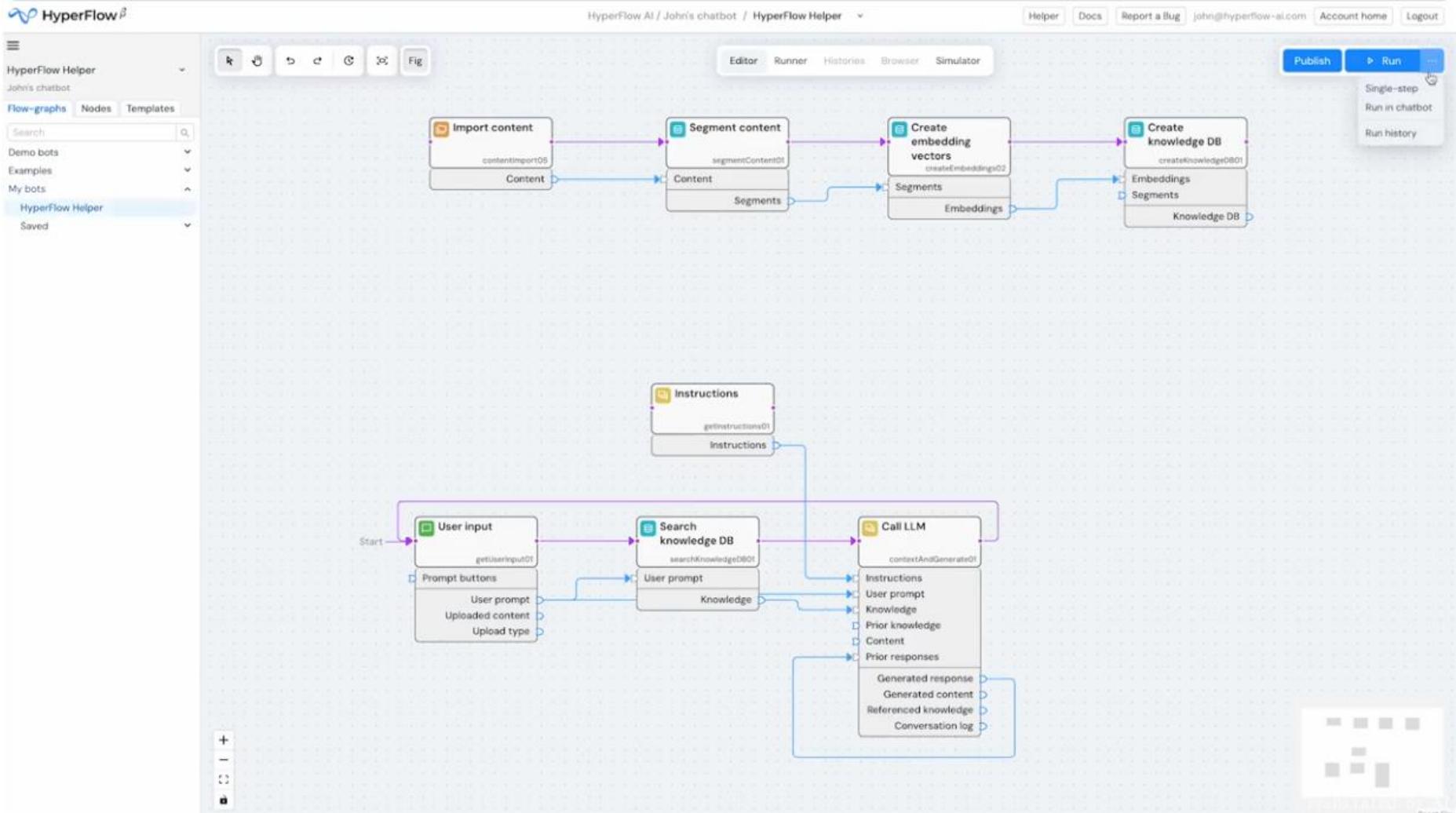
User input #49 get userInput01 Step 5

Waiting on settings OK...

React Flow

```
graph LR; Content[Content] --> ContentNode[Content]; Content --> Segments[Segments]; Segments --> Embedding[Create embedding vectors]; UserInput[User input #49 get userInput01] --> SearchKnowledge[Search knowledge DB]; UserInput --> UserPrompt[User prompt]; UserInput --> UploadContent[Uploaded content]; UserInput --> UploadType[Upload type];
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Pause

Run session tags

User input #49 getUserId01 Step 1

All locks OK

Prompt input in test chatbot

OK

Instructions

getInstruc

Instruc

User input #48 getUserId01 Step 1

Waiting on settings OK...

what are pure data nodes?

▶

Translated by AI

```
graph LR; Content[Content] --> Segments[Segments]; Content --> Create[Create embedding vectors]; Segments --> UserInput1[User input #49 getUserId01]; Create --> UserInput1; UserInput1 --> Instructions[Instructions]; UserInput1 --> UserInput2[User input #48 getUserId01]; Instructions --> UserInput2; UserInput2 --> Search[Search knowledge DB]; UserInput2 --> UserPrompt[User prompt]; Search --> UserPrompt;
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper ✗ Edit

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Pause

Run session tags:

User input #48 getUserId01 Step 1

All locks ⊕ ⊖ ⊖

Mode: text

Prompt input in test chatbot

OK

User input #49 getUserId01 Step 3

Waiting on settings OK...

Greetings from HyperFlow Helper!

what is the Merge Inputs node used for?

what is the Merge Inputs node used for?

React Flow

```
graph LR; Content1[Content] --> Create[Create embedding vectors]; Create --> UserInput1[User input]; UserInput1 --> Instructions[Instructions]; UserInput1 --> UserInput2[User input]; UserInput2 --> Search[Search knowledge DB]; Search --> UserPrompt[User prompt]
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



**HyperFlow <sup>β</sup>**

HyperFlow AI / John's chatbot / HyperFlow Helper ✗ Edit

Helper Docs Report a Bug john@hyperflow.ai.com Account home Logout

Greetings from HyperFlow Helper!

what is the Merge Inputs node used for?

The Merge Inputs node is used to combine multiple input sources into a single output stream within a flow-graph. It supports various data types and provides flexible merge modes, allowing you to efficiently manage and combine data flows between nodes. Some of its specific uses include combining data streams, merging historical data, and interleaving sequences. This node is particularly useful when you want multiple nodes to source an input data connector on another node, as input nodes are limited to single data source connections in the current release.

For additional guidance on using the Merge Inputs node effectively, consider topics such as configuration guidelines and examples of merge modes.

References:

- Merge Inputs

User input #49 get userInput() Step 5

Mode: text

Prompt input in test chatbot OK

Call LLM #52 contentAndGenerate() Step 4

Parameters

Composite prompt

Type	Content	Approx. tokens	Source step
1. instructions	You are a helpful and informative assistant who guides users on using the HyperFlow flow-graph-based development environment for building AI-based applications.	316	3
2. knowledge	--- "Knowledge Segments": [{"segment_id": "000", "content": "Content"}, {"segment_id": "001", "content": "Content"}, {"segment_id": "010", "content": "Content"}, {"segment_id": "011", "content": "Content"}, {"segment_id": "012", "content": "Content"}], "Merge Inputs Node #49 Overviews The Merge Inputs Node combines multiple input sources into a single output stream."	2347	2
3. user	"User Query": "what is the Merge Inputs node used for?"	9	1

Generated text

The Merge Inputs node is used to combine multiple input sources into a single output stream within a flow-graph. It supports various data types and provides flexible merge modes, allowing you to efficiently manage and combine data flows between nodes. Some of its specific uses include combining data streams, merging historical data, and interleaving sequences. This node is particularly useful when you want multiple nodes to source an input data connector on another node, as input nodes are limited to single data source connections in the current release.

For additional guidance on using the Merge Inputs node effectively, consider topics such as configuration guidelines and examples of merge modes.

```
{"segment_ids": ["000", "001", "011", "013"]}
```

References

References:  
Merge Inputs

Metadata

Model	Created	Elapsed	Prompt Tokens	Completion Tokens
Eft-4a-2024-08-06	2025-01-21T05:08	3.48s	870	141

User input #49 get userInput() Step 5

Waiting on settings OK...

# 프로덕션 AI 앱을 위한 플로우그래프 배포



 HyperFlow

<https://hyperflow-ai.com/chabotservice>

English ▾

## HyperFlow AI Beta Launch

Announcing HyperFlow AI, the advanced, no-code development environment for agentic and RAG-based generative AI applications.

[Learn more and register for access! >](#)

## We create **AI Chatbots**

AI Chatbots, tailored to your needs,  
built with HyperFlow AI, Mirinae's next-gen AI platform

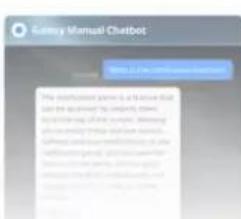
[Contact us](#)

Try these chatbots below!

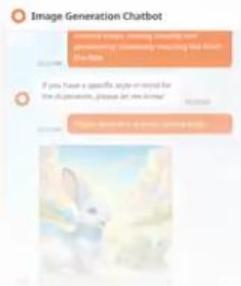
### Insurance Policy Chatbot



### Rainy Month Chatbot



### Image Generation Chatbot



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow

<https://hyperflow-ai.com/chabotservice>

English

Uploaded document

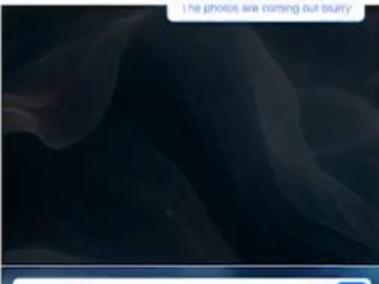
Galaxy24-manual-Korean PDF

Here are some questions you may ask!

What is the notification panel function?

How do I enter data?

The photos are coming out blurry.



Chatbot prompt



## Image Generation

With just an idea for an illustration, the chatbot automatically generates stunning webtoon panels in your desired style. Simply enter a plot and the number of panels you need, and watch as your imagined scenes come to life instantly. Customize the style freely to create your own unique webtoon!

Create images from your own story

A quick rabbit and a slow tortoise find themselves in a race. Confident in its speed, the rabbit becomes overconfident and takes a nap midway. Meanwhile, the tortoise keeps moving steadily and persistently, ultimately reaching the finish line first.

## Image Generation Chatbot

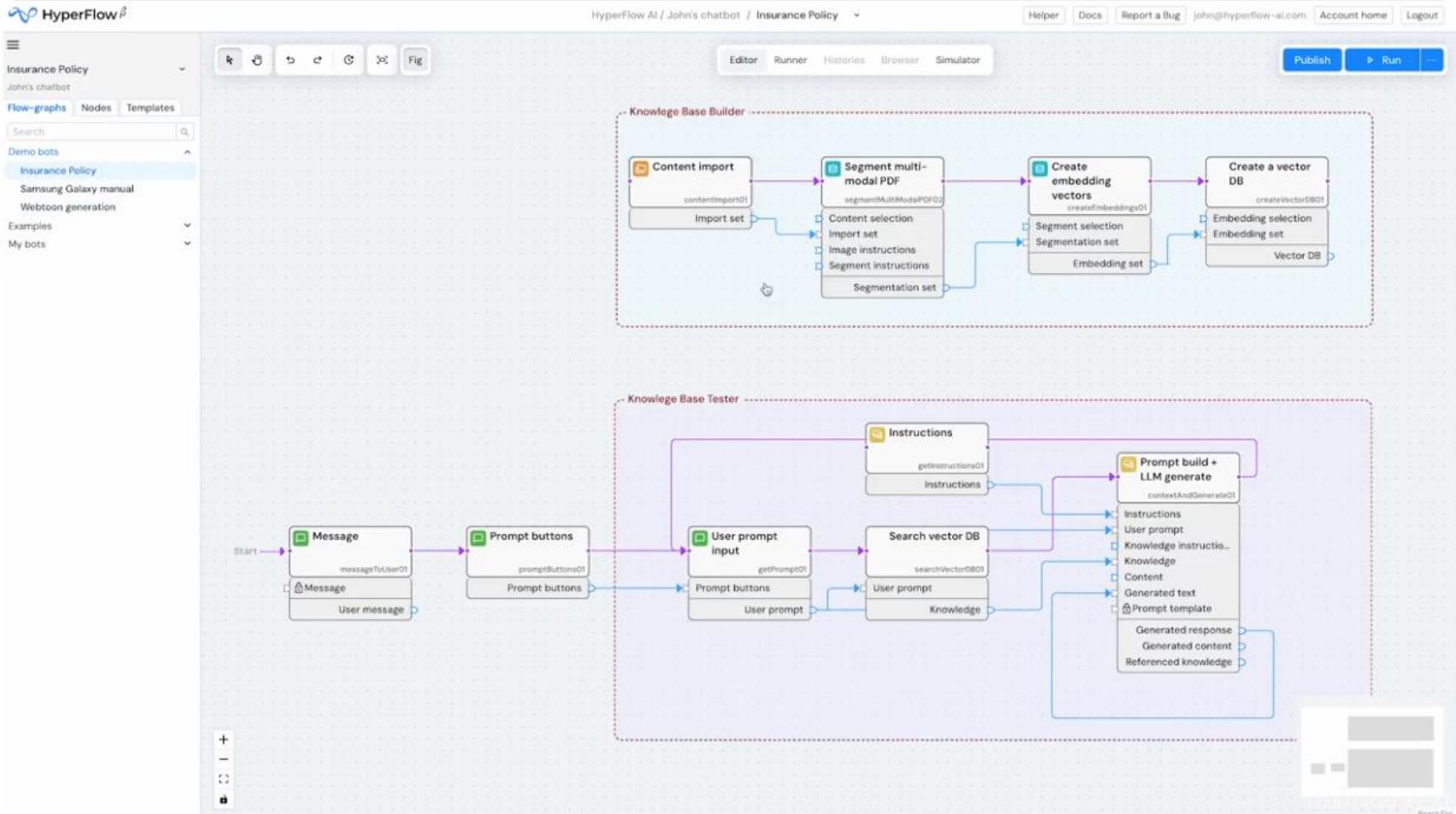
the woman throughout the images, ensuring her attire, hair color, and demeanor are uniform across the various contexts within the bustling harbor. This consistency is crucial as she engages with the surroundings, capturing memories in this iconic location, reflecting a joyful, relaxed day in Sydney.



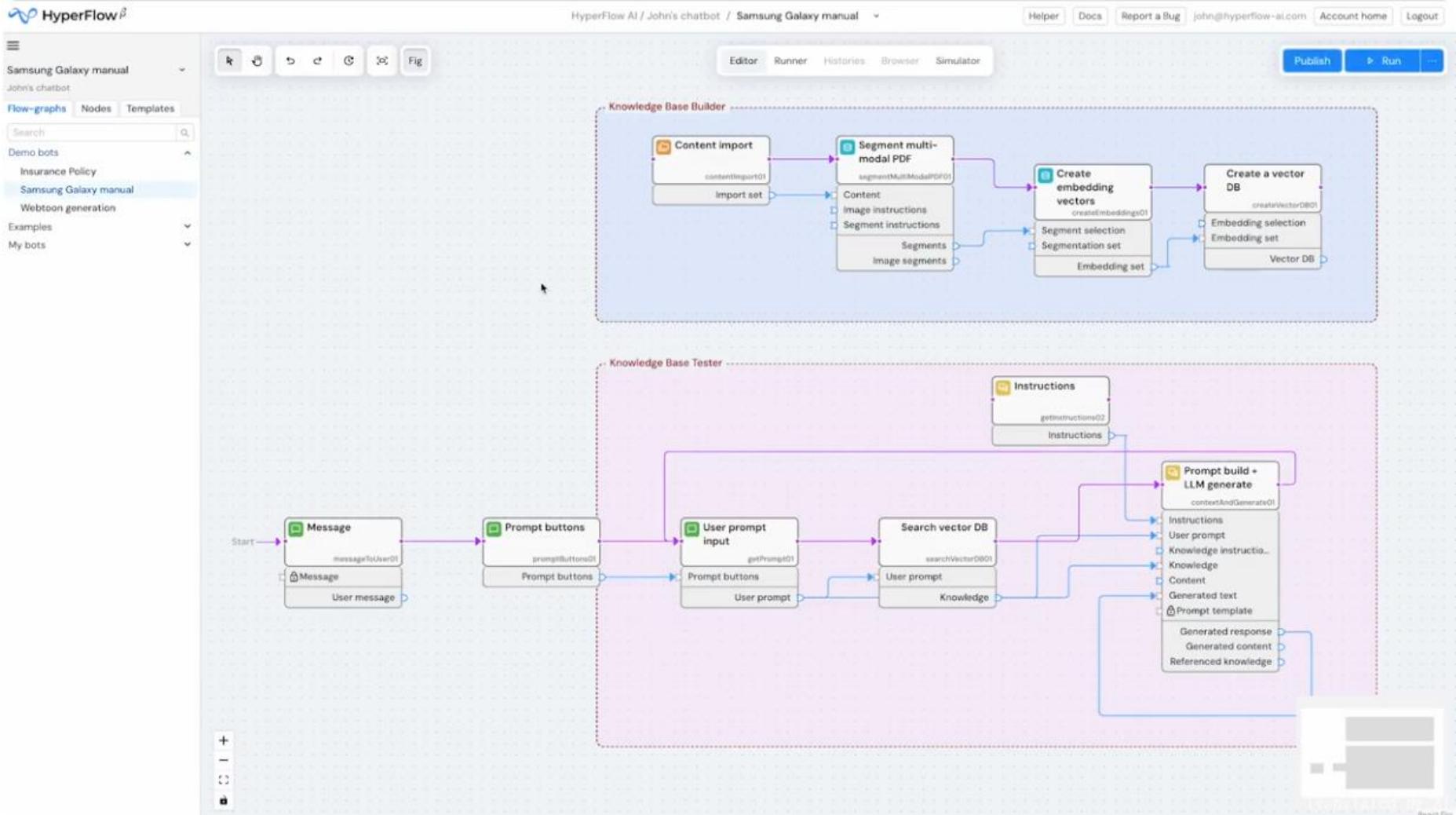
Chatbot prompt



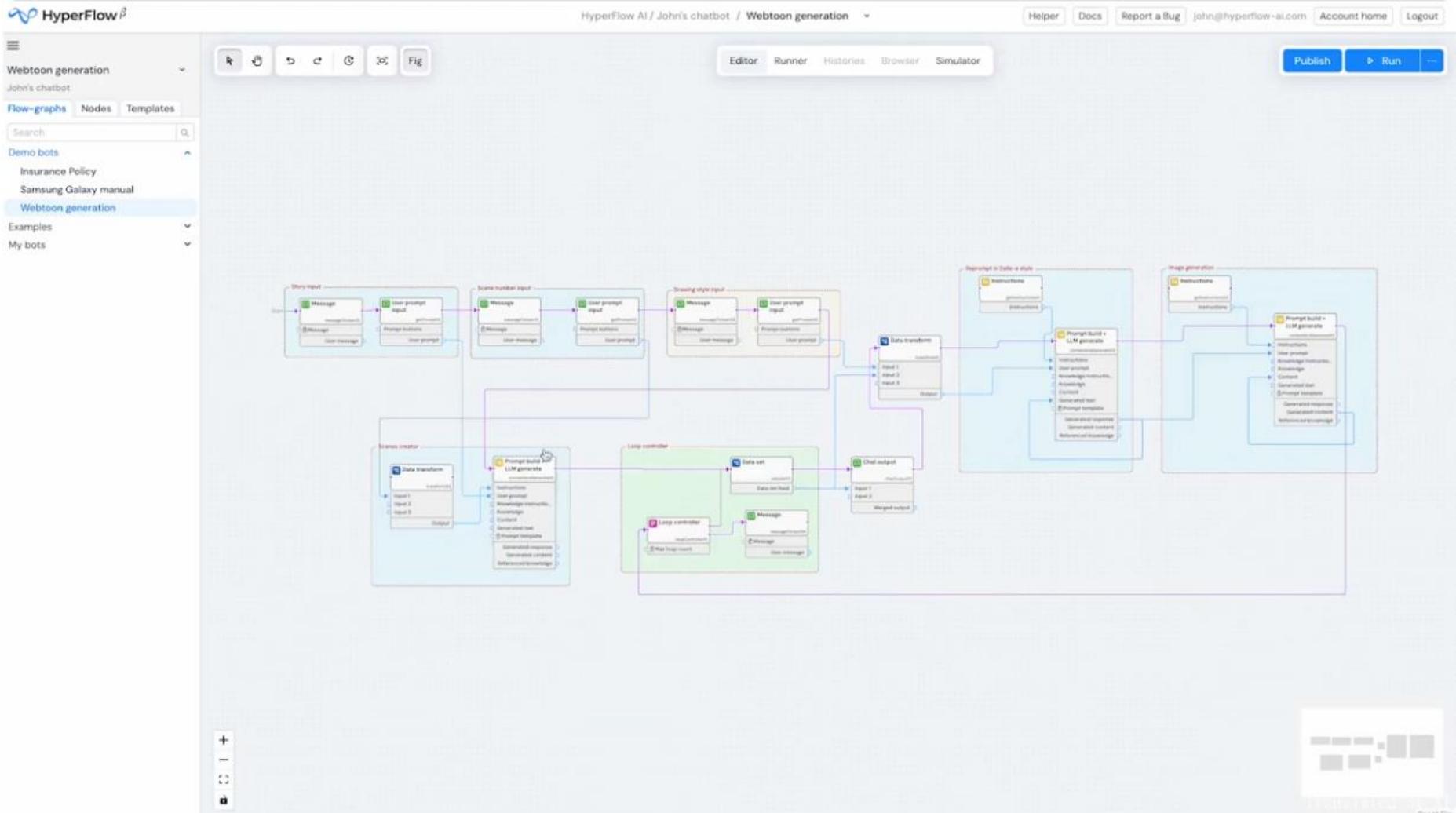
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow

English

## Galaxy interactive manual

This chatbot was trained on the Samsung Galaxy 24+ manual, providing precise answers to user questions. It learned the 203-page manual in Korean, including text and images, and uses Retrieval-Augmented Generation (RAG) technology with advanced LLMs to provide conversational help to users in any language.

### Uploaded document

Galaxy24-manual-Korean PDF

### Here are some questions you may ask!

What is the notification panel function?

How do I enter data?

The photos are coming out blurry.

## Galaxy Manual Chatbot

What can I help you with? You can choose from the options or ask any related question of your own:

What is the notification panel function

How do I enter data

The photos are coming out blurry

The photos are coming out blurry

Blurry photos can result from several factors. Here are some tips to improve your picture quality:

1. **Stability:** Make sure to hold your phone steady while taking the photo. Camera shake can cause blurriness, especially in lower light conditions. If you're capturing an image in a dim setting, using a tripod can help stabilize your phone.

2. **Focus:** Ensure that you're focusing on your subject properly before capturing the photo. If you're too close to the subject, autofocus may

Chatbot prompt



## Image Generation

With just an idea for an illustration, the chatbot automatically generates stunning webtoon panels in your desired style. Simply enter a plot and the number of panels you need, and watch as your imagined scenes come to life instantly. Customize the style freely to create your own unique webtoon!

### Create Images from your own story!

A quick rabbit and a slow tortoise find themselves in a race. Confident in its speed, the rabbit becomes overconfident and takes a nap midway. Meanwhile, the tortoise keeps moving steadily and persistently, ultimately reaching the finish line first.

## Image Generation Chatbot

middle-aged woman with curly auburn hair, wearing a light, flowing white sundress adorned with subtle blue floral patterns. She has a warm smile and is wearing large, oversized sunglasses, along with a wide-brimmed straw hat to shield herself from the sun. She is standing on the edge of the harbor with a camera in hand, capturing the picturesque landscape.

Around her, illustrate a blend of individuals including a group of friends laughing, a family with children playing, and a couple sitting on a bench, sharing ice cream cones. The scene is alive with vibrant energy, demonstrating the melting atmosphere of Circular Oasis. The importance lies in maintaining the consistent representation of the woman throughout the images, ensuring her attire, hair color, and demeanor are uniform across the various contexts within the bustling harbor. This consistency is crucial as she engages with the surroundings, capturing memories in this

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Documentation

Made with

## HyperFlow Beta Documentation

▼ 5 more properties

- [Beta program welcome](#)
- [Release notes](#)
- [hyperflow-ai.com/hyperflow - the HyperFlow IDE](#)
- [HyperFlow-AI community support on Discord](#)
- [HyperFlow-AI community support on KakaoTalk](#)
- [HyperFlow-AI tutorial video channel on Youtube](#)

---

- [Nodes reference](#)
- [Services reference](#)
- [Data types](#)
- [Custom AI App Control API Reference](#)

I



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Doc... / Custom AI App Control API Referenc...

Made with

Headers  
Body  
Response  
Code examples  
Responses examples  
Progress a session  
Key Features  
When to use  
Best Practices  
Headers  
Body  
Code examples  
Responses examples

File Upload Process  
Step 1: File Upload Endpoint  
Headers  
Request Body  
Response Format  
Code examples  
Responses examples  
Step 2: Using the Uploaded File  
Headers  
Request Body  
Response Format  
Code examples  
Responses examples  
App launch context  
Launch Parameters node  
Embedded chatbot URL Format  
Example Usage

request interaction, and so the client-side app in all cases except those awaiting input from the user, will immediately make another Control Progress call to wait for any newly-available output or requests from the flow-graph and deliver it to the user in a timely manner.

## General call forms

Both calls are `POST` HTTP calls, sending data in JSON structures in the POST body and receiving responses also in JSON form. The detailed definition of these structures is covered below.

The two calls each require a key field in the POST body, as follows

- **Control Start** requires the 24 character internal identifier of the flow-graph to be started in the `flowGraphID` POST body field. The correct form of this call is given in the **API Calls** button popup in the flow-graph's entry in the Published Flow-graph Manager UI. It can only reference *published* flow-graphs, attempting to launch unpublished flow-graphs will fail.
- A key entry in the response from a successful **Control Start** call is the `sessionId` field, which uniquely identifies this particular run of the flow-graph.
- **Control Progress** requires the `sessionId` returned from the Control Start call to be supplied in the POST body of all subsequent Control Progress calls to distinguish this user's interaction with the flow-graph.

## Understanding API responses

All API calls will return JSON with both a `success` and a `data` property. Exceptions to this will be specified in the documentation.

You should always attempt to JSON decode the response, then use the success property to determine if the API call succeeded.

```
JSON
{
 "success": true,
 "data": {...},
 "message": {...}
}
```

Copy

The `data` property of errors will include an `message` and an `error_code` to help you determine what went wrong. Non-zero error codes should never change so you can rely on them to make programming flow choices if necessary.

```
{
 "success": false,
 "message": {...},
 "cause": {...}
}
```

## Data field

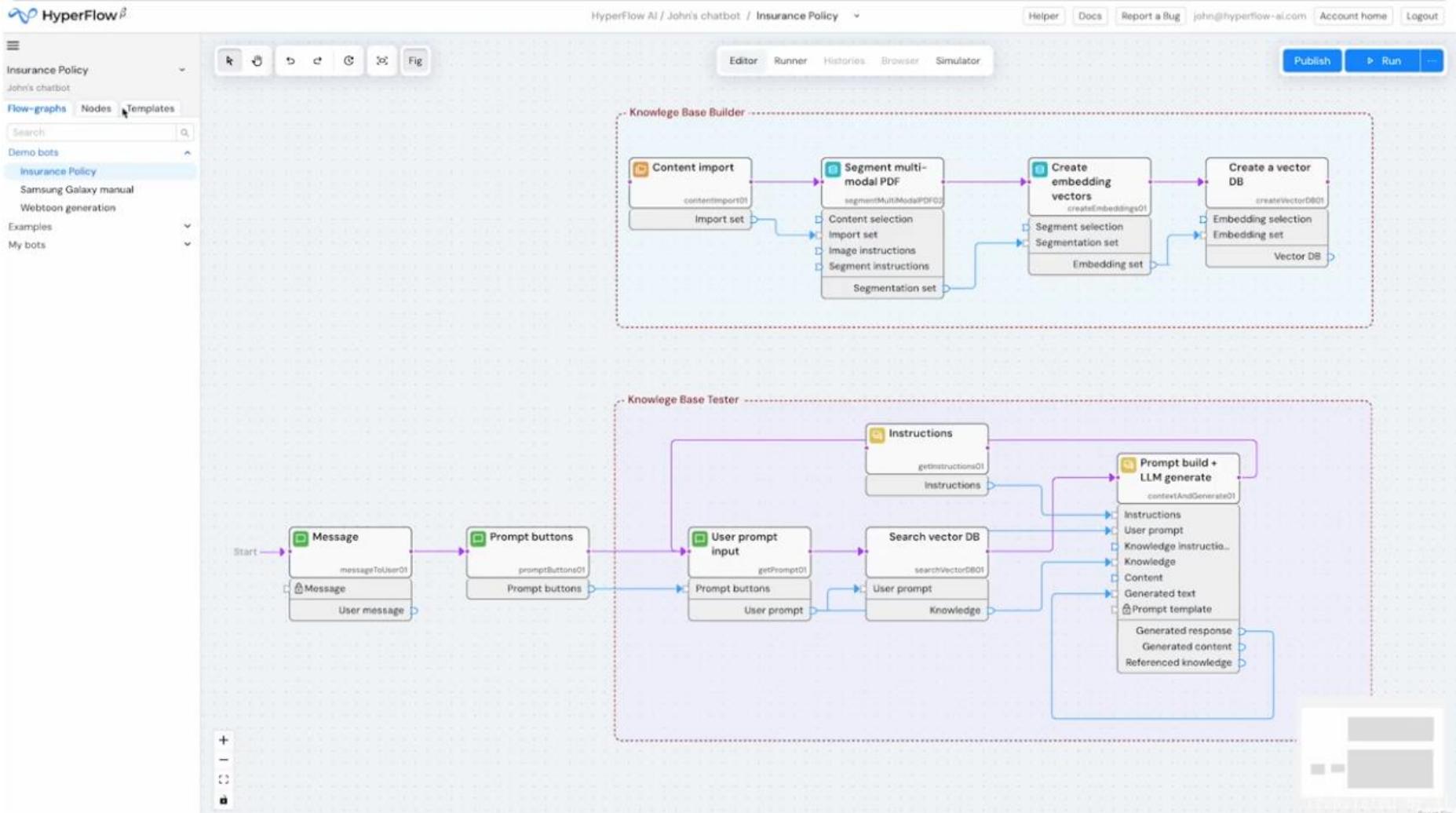
# 프로덕션 AI 앱을 위한 플로우그래프 배포



```
return <div>Unsupported media type</div>;
};

return (
 <>
 {(title || titleIconSrc) && (
 <TitleBar className="bot-title-bar" ref={titleBarRef} bgSrc={titleBgSrc}>
 <Title className="bot-title">
 {titleIconSrc && <TitleIcon className="bot-title-icon" src={titleIconSrc} />}
 {title}
 </Title>
 </TitleBar>
)}
 <ChatBody className="bot-body" ref={chatBodyRef} bgSrc={bodyBgSrc}>
 <ChatHistory ref={chatHistoryRef}>
 {({headerIconSrc || headerTitle) && (
 <Header className="bot-header">
 {headerIconSrc && <HeaderIcon className="bot-header-icon" src={headerIconSrc} />}
 {headerTitle && <HeaderTitle className="bot-header-title">{headerTitle}</HeaderTitle>}
 {headerSubtitle && <HeaderSubtitle className="bot-header-subtitle">{headerSubtitle}</HeaderSubtitle>}
 </Header>
)}
 {chatHistory.map((h, i) => (
 <React.Fragment key={i}>
 {h.type === "generator" ? (
 <Response className="bot-gen-text">
 {responseIconSrc && <ResponseIcon className="bot-response-icon" src={responseIconSrc} />}
 <MarkdownRenderer className="bot-gen-text-markdown" content={cleanup(h.text)} />
 </Response>
) : h.type === "user" ? (
 <UserPrompt className="bot-prior-prompt">{h.text}</UserPrompt>
) : h.type === "branch" ? (
 <BranchChoice className="bot-branch-choice">
 {branchSelectedMessage}
 {h.text}
 </BranchChoice>
) : h.type === "fileUpload" ? (
 <FileUpload className="bot-file-upload">
 <FileName className="bot-file-name">
 {attachedFileMessage}
 {h.filename}
 </FileName>
 {renderMedia(h)}
 </FileUpload>
) : h.type === "message" ? (
 <Message className="bot-message">
 {responseIconSrc && <ResponseIcon className="bot-response-icon" src={responseIconSrc} />}
 {/**MessageText className="bot-message-text">{h.text}</MessageText>*/}
 <MarkdownRenderer className="bot-message-text bot-message-text-markdown" content={cleanup(h.text)} />
 </Message>
) : h.type === "exception" ? (
 <Exception className="bot-error">
 {exceptionIconSrc && <ResponseIcon className="bot-exception-icon" src={exceptionIconSrc} />}
 <ExceptionText className="bot-error-text">Sorry, chatbot error: {h.exception.exception}</ExceptionText>
 </Exception>
)
 </React.Fragment>
)}
 </ChatHistory>
 </ChatBody>
</div>
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Editor Runner Histories Browser Simulator Publish Run

HyperFlow Helper

John's chatbot

Flow-graphs Nodes Templates

Search

Demo bots Examples My bots

HyperFlow Helper

Flow-graphs > My bots

Saved

Import content

contentImport01

Content

Segment content

segmentContent01

Content

Segments

Create emb vec

emb vec

Segment

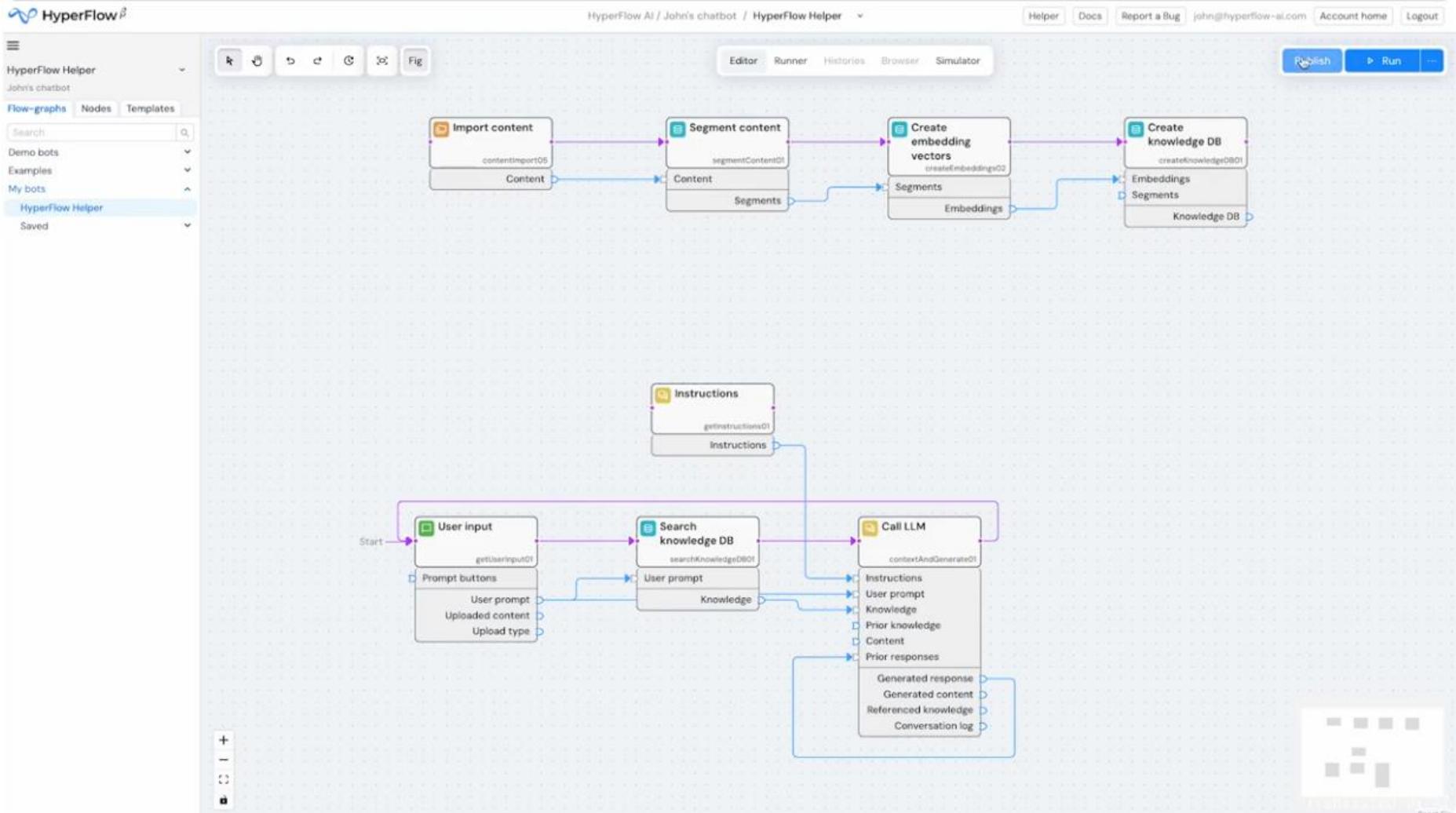
Instructions

getInstructions01

React File

```
graph LR; Import[Import content] -- contentImport01 --> Segment[Segment content]; Import -- Content --> Segment; Segment -- segmentContent01 --> Create[Create emb vec]; Segment -- Segments --> Import; Create -- emb vec --> Segment; Instructions[Instructions] -- getInstructions01 --> React[React File]
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow beta

HyperFlow Helper

John's chatbot

Flow-graphs Nodes Templates

Import content

contentImport01

Content

Validate & confirm flow-graph publish

Checking node #47 Create knowledge DB...

Node #45: Import content (contentImport01)

Tags: Unlocked  
Page or site URL: Unlocked  
Maximum pages: Unlocked

Cancel Confirm

create embedding vectors

createEmbedding01

embeds

Embeddings Segments

Knowledge DB

Instructions

getInstructions01

User input

getUserInput01

Prompt buttons

User prompt

Uploaded content

Upload type

Search knowledge DB

searchKnowledgeDB01

User prompt

Knowledge

Call LLM

contextAndGenerate01

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

Referenced knowledge

Conversation log

Start

Translated by AI

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Helper

HyperFlow Helper / Flow-graphs / HyperFlow Helper

Validate & confirm flow-graph publish

Check finished with issues.

Node #45: Import content (contentImport0\$)  
Tags: Unlocked  
Page or site URL: Unlocked  
Maximum pages: Unlocked

Node #48: Segment content (segmentContent0\$)  
Pattern: Unlocked  
Overlap: Unlocked  
Tags: Unlocked

Cancel Confirm

HyperFlow Helper

Flow-graphs Nodes Templates

Import content

contentImport0\$

content

contentpoints

Content

create embedding vectors

createEmbeddingVectors0\$

Events

Embeddings

Create knowledge DB

createKnowledgeDB0\$

Embeddings

Segments

Knowledge DB

Instructions

getInstructions0\$

Instructions

User input

getUserInput0\$

Prompt buttons

User prompt

Uploaded content

Upload type

Search knowledge DB

searchKnowledgeDB0\$

User prompt

Knowledge

Call LLM

connectAndGenerate0\$

Instructions

User prompt

Knowledge

Prior knowledge

Content

Prior responses

Generated response

Generated content

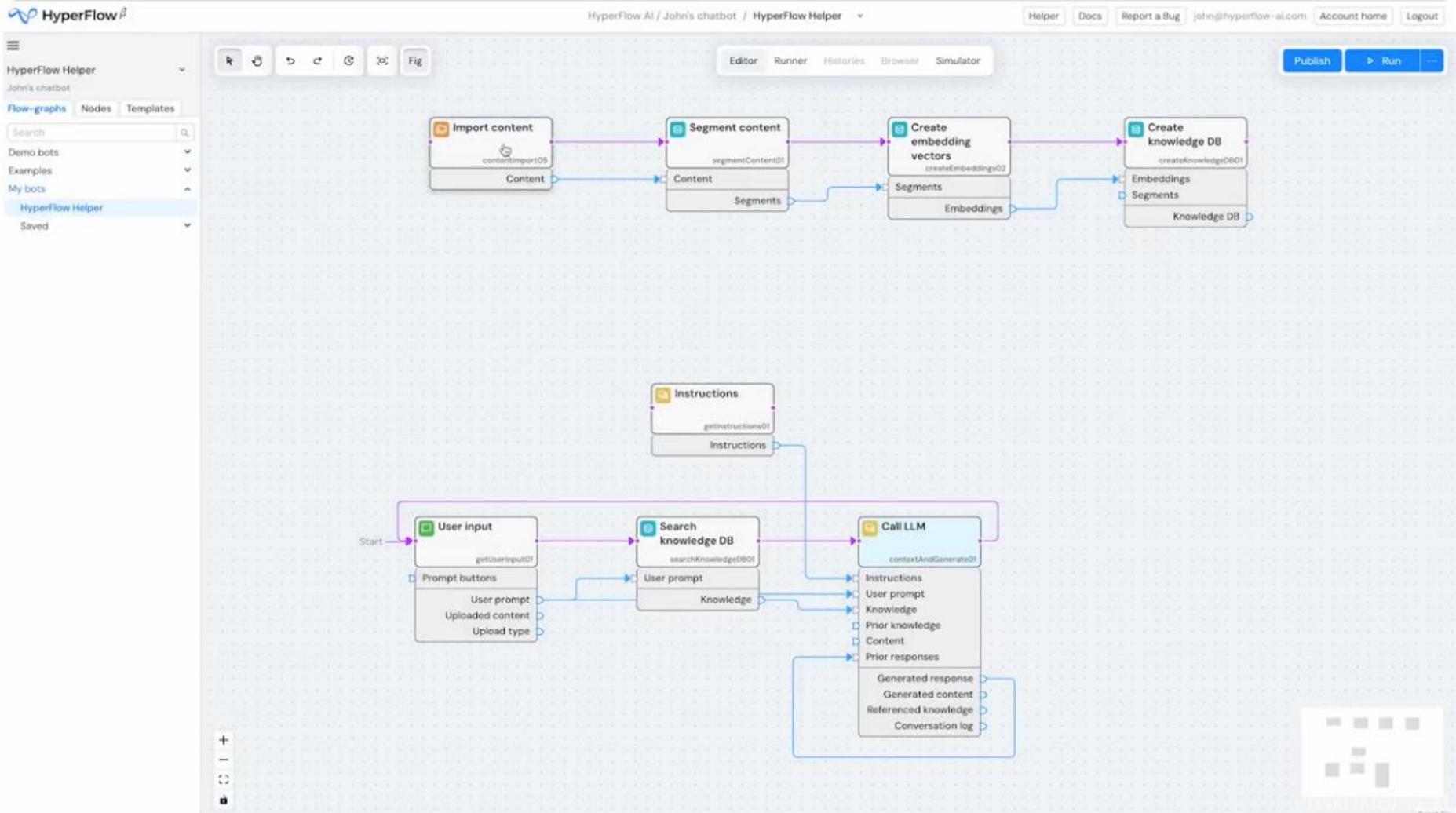
Referenced knowledge

Conversation log

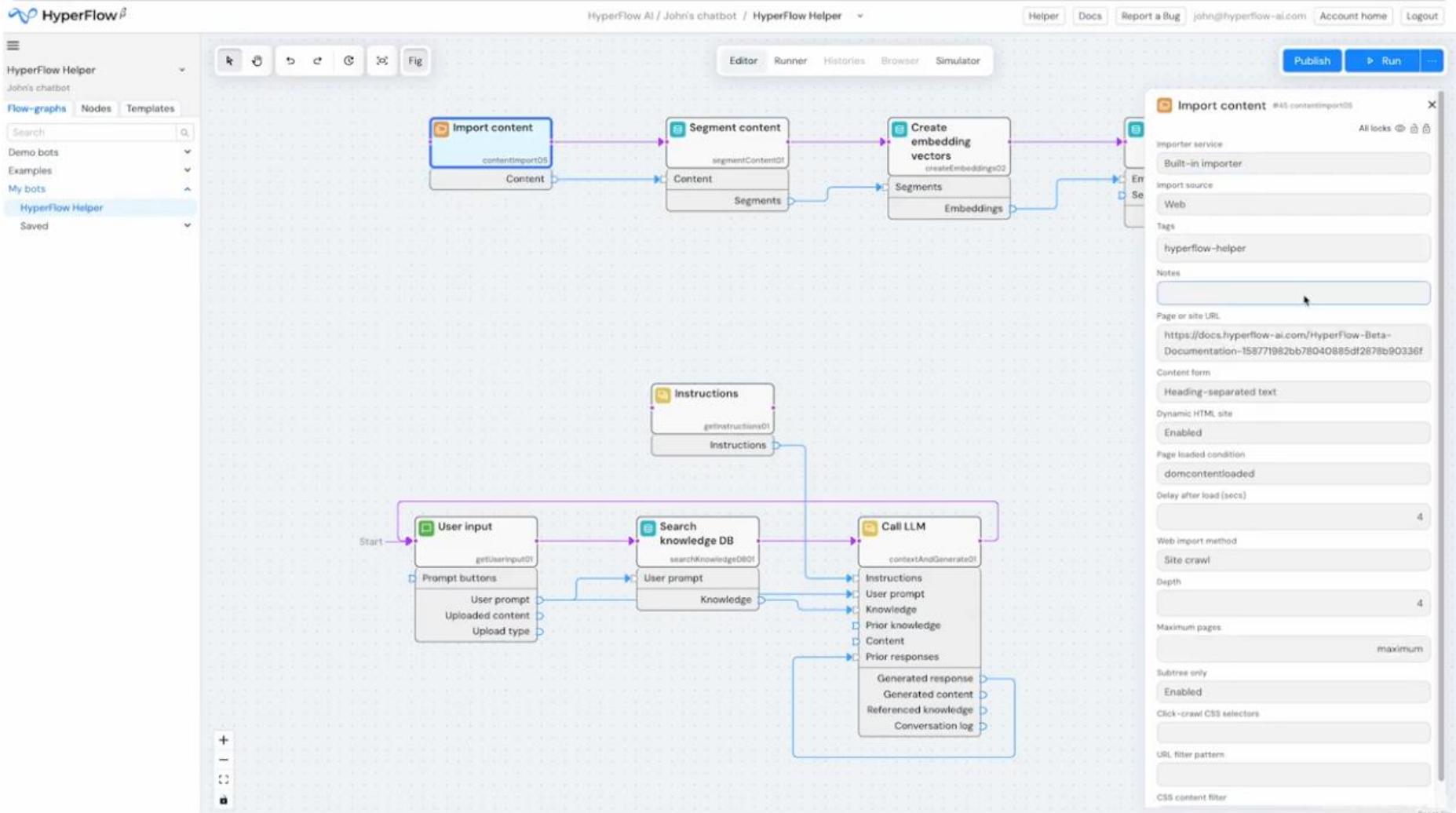
Start

Translated by AI

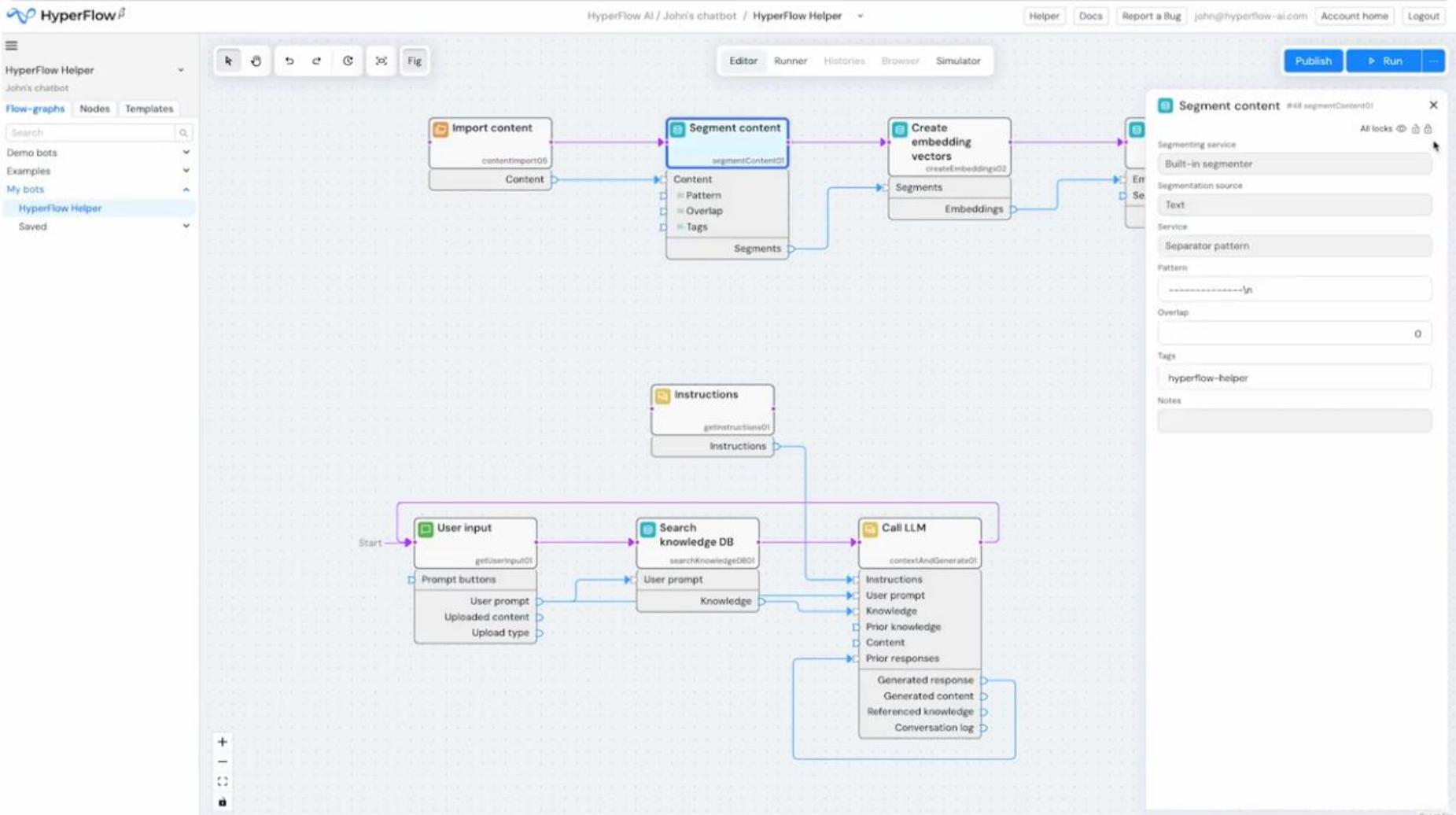
# 프로덕션 AI 앱을 위한 플로우그래프 배포



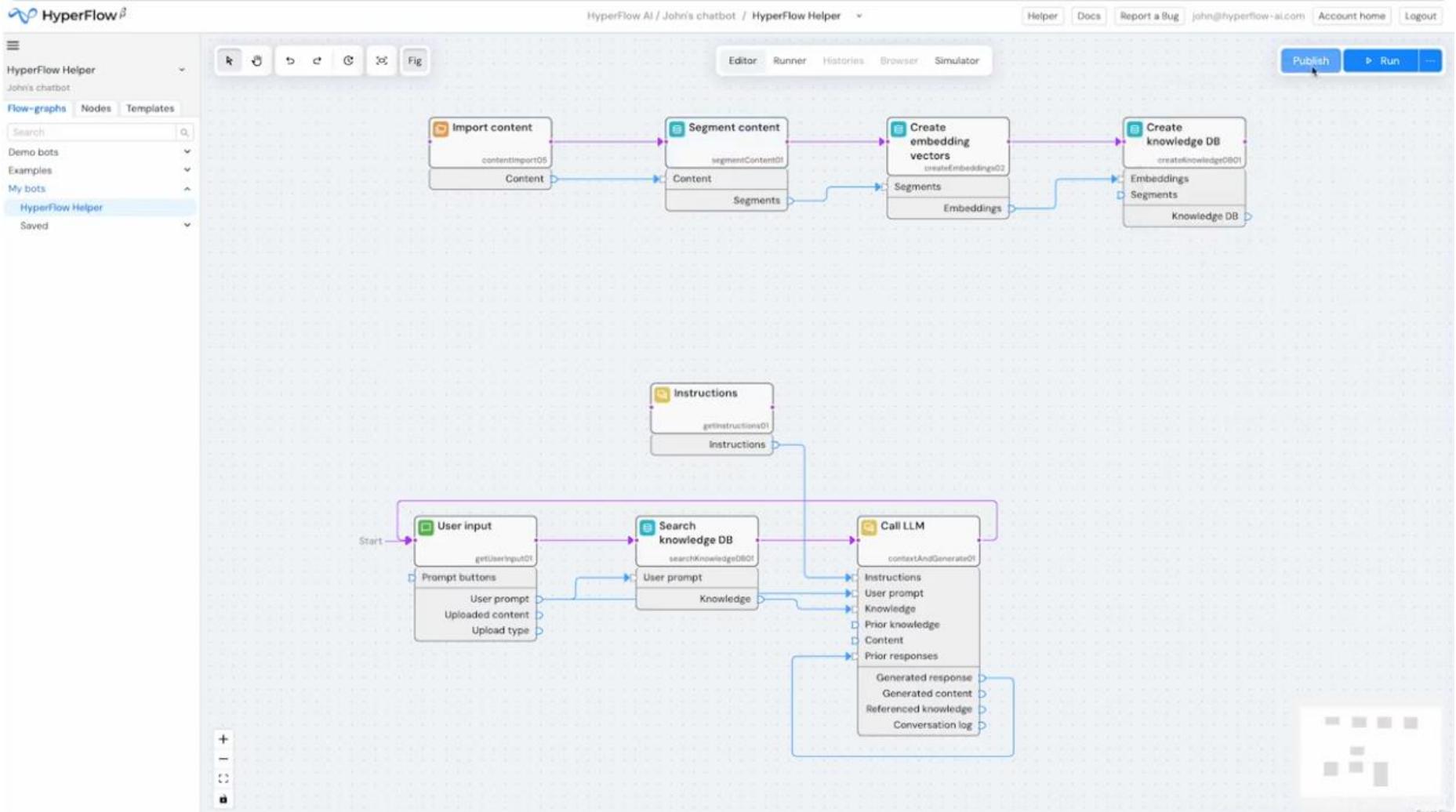
# 프로덕션 AI 앱을 위한 플로우그래프 배포



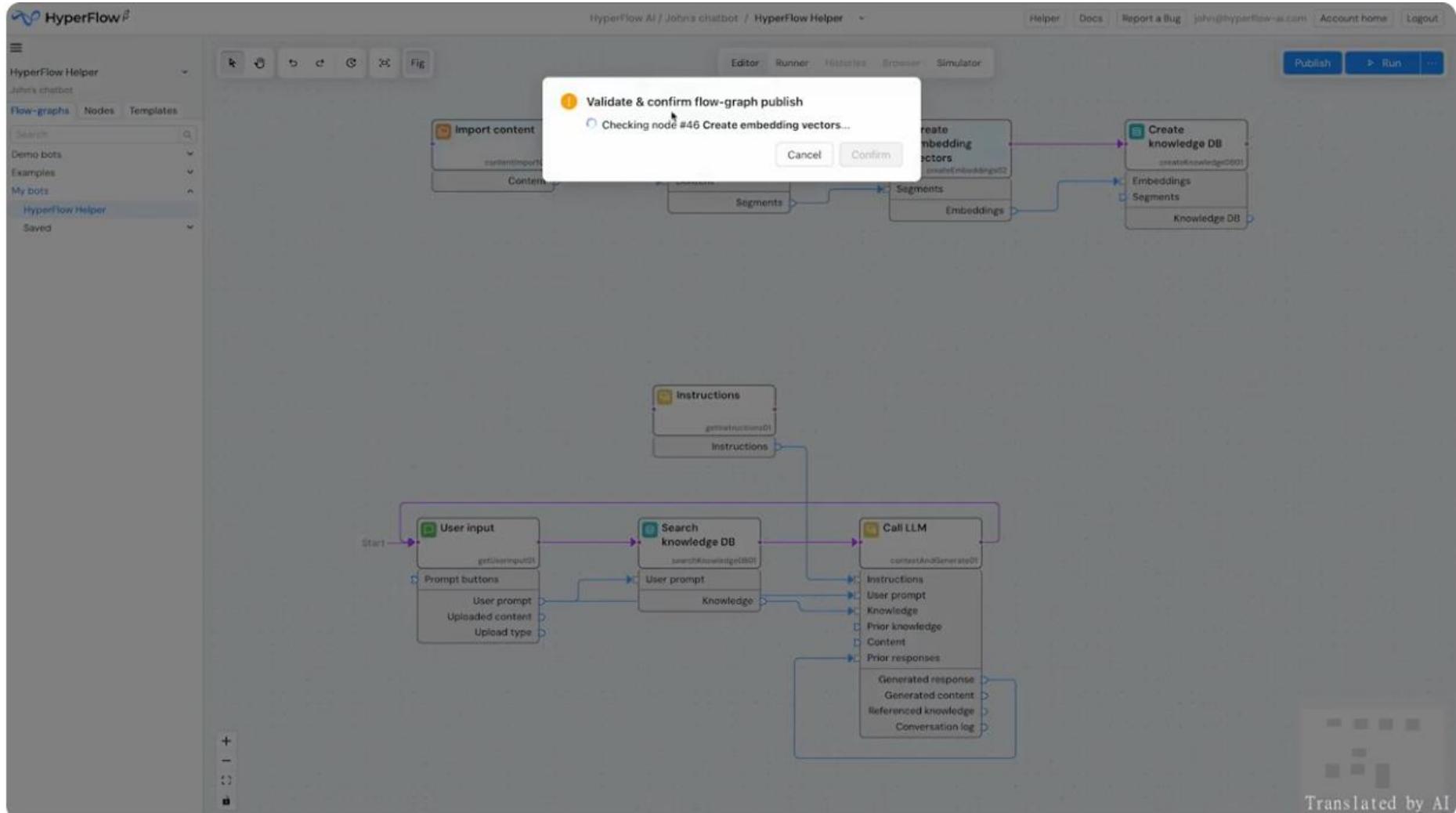
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow 

HyperFlow Helper / John's chatbot / HyperFlow Helper

Editor Runner History Browser Simulator Publish Run

Validate & confirm flow-graph publish  
Check finished OK, please confirm publishing.

Cancel Confirm

Import content → Content → Segments → Embedding vectors → Segments → Create knowledge DB → Knowledge DB

Flow-graphs Nodes Templates

Search Demo bots Examples My bots HyperFlow Helper Saved

Translated by AI

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	<a href="#">External test</a> API calls Embed code	<a href="#">Activate</a> <small>...</small>

[Bot mods →](#)  
[+ Create Project](#)

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)

**Published flow-graphs**

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	<a href="#">External test</a> <small>0</small> API calls Embed code	<a href="#">Activate</a> <small>...</small>

[Bot mods →](#)  
[+ Create Project](#)



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)  
**Published flow-graphs**  

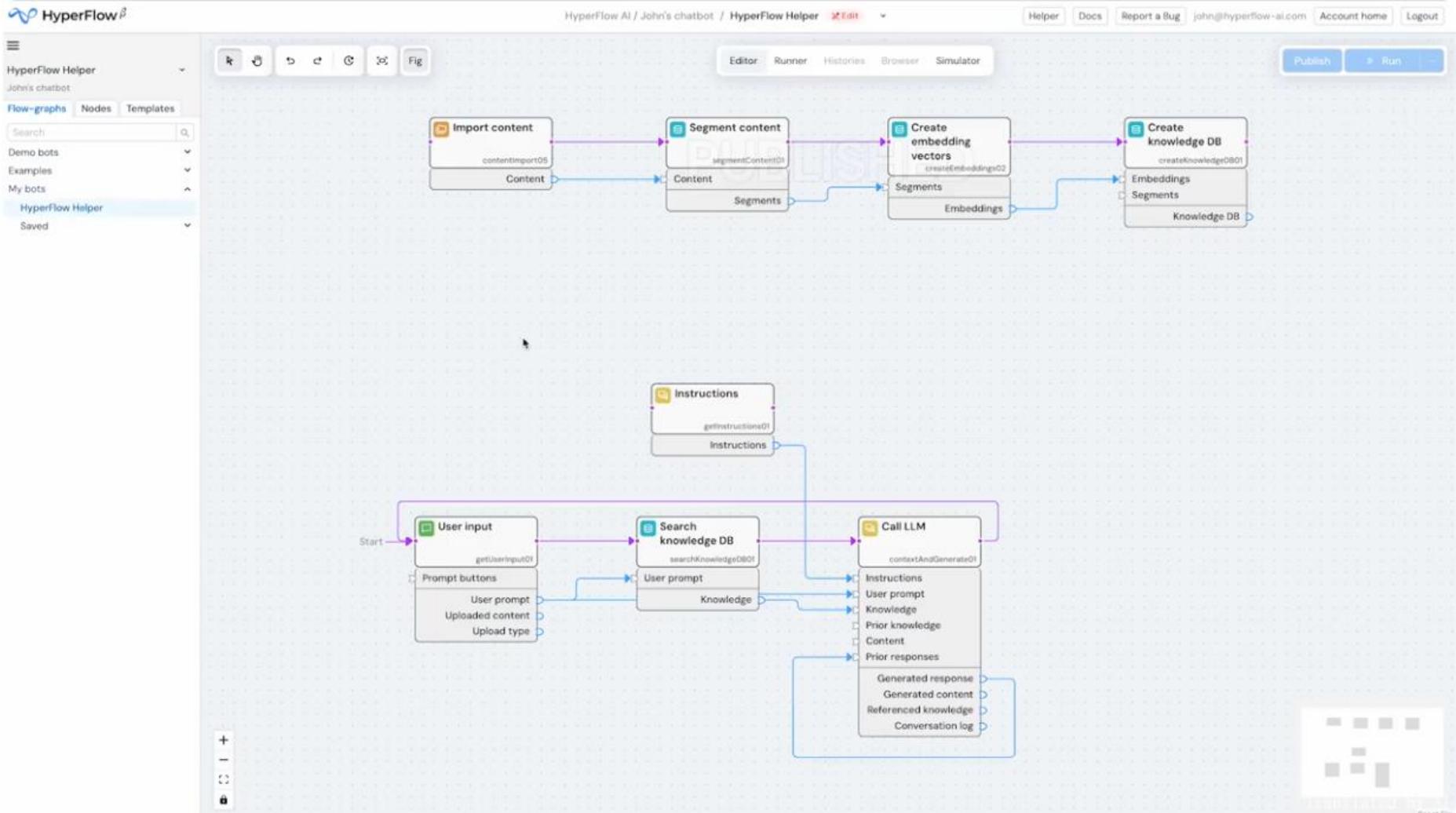
Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	<a href="#">External test</a> API calls Embed code	<a href="#">Activate</a> <small>[...]</small>

[Bot mode →](#)  
[+ Create Project](#)

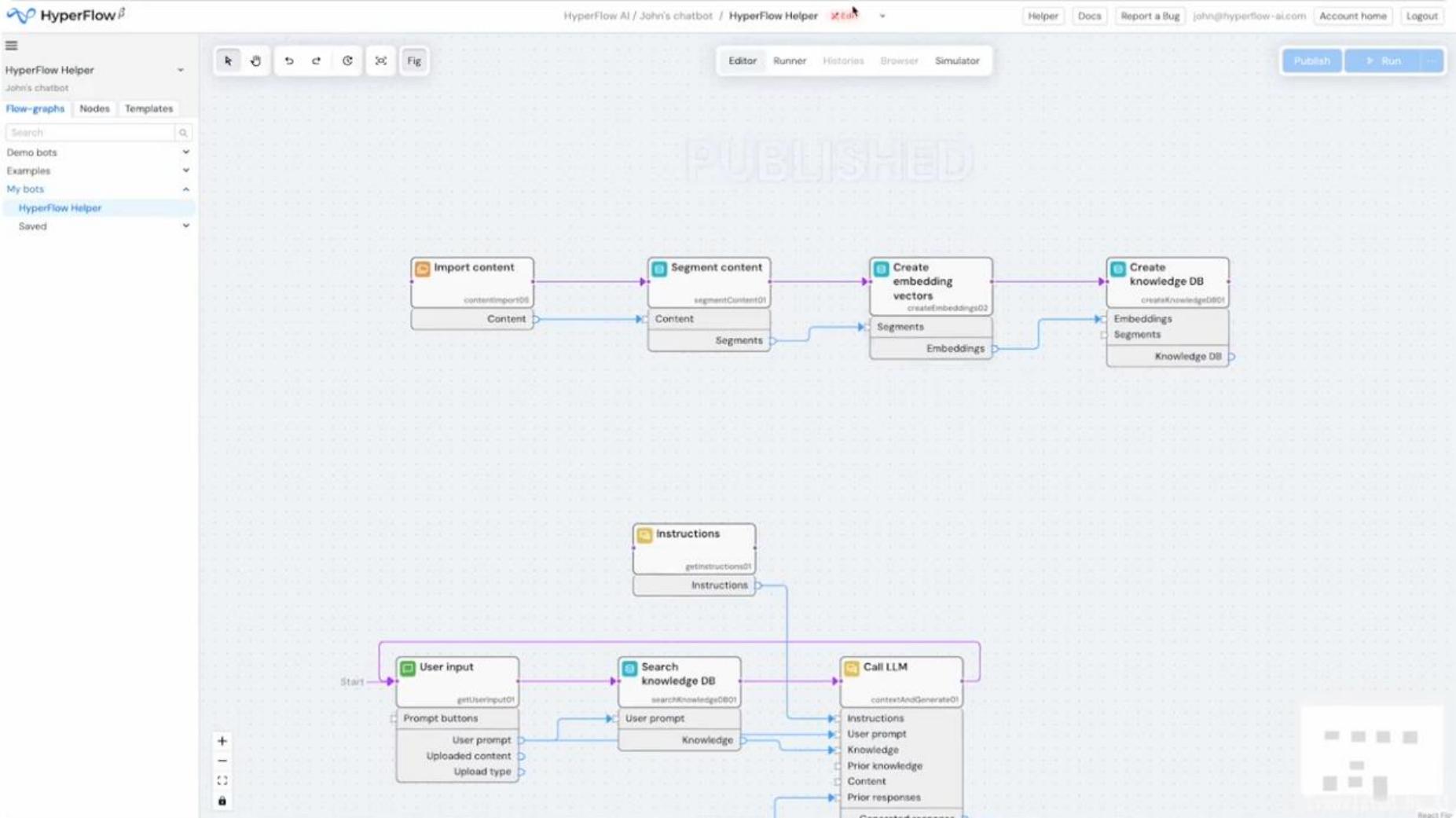
Test ↻  
Pause  
Archive  
Enter maintenance

A screenshot of the HyperFlow AI web application. At the top, there's a navigation bar with links for Helper, Docs, Report a Bug, email (john@hyperflow-ai.com), Account home, and Logout. Below that is a 'Create Organization' button. The main area is titled 'Organizations' and shows one organization named 'HyperFlow AI' owned by 'john@hyperflow-ai.com'. Under 'Members', it lists 'john@hyperflow-ai.com' as the owner with a delete icon. There's a button to '+ Invite member'. Under 'Projects', it shows a single project named 'John's chatbot' with a link to its details. The project details include a table with columns for Flow-graph, Status (HyperFlow Helper, ready), Published (2025-01-21 15:13), Links and code (with External test, API calls, Embed code buttons), and Actions (Activate, more options). Below the table are buttons for Bot mode and + Create Project. A context menu is open over the Actions button, showing options: Test (with a reload icon), Pause, Archive, and Enter maintenance.

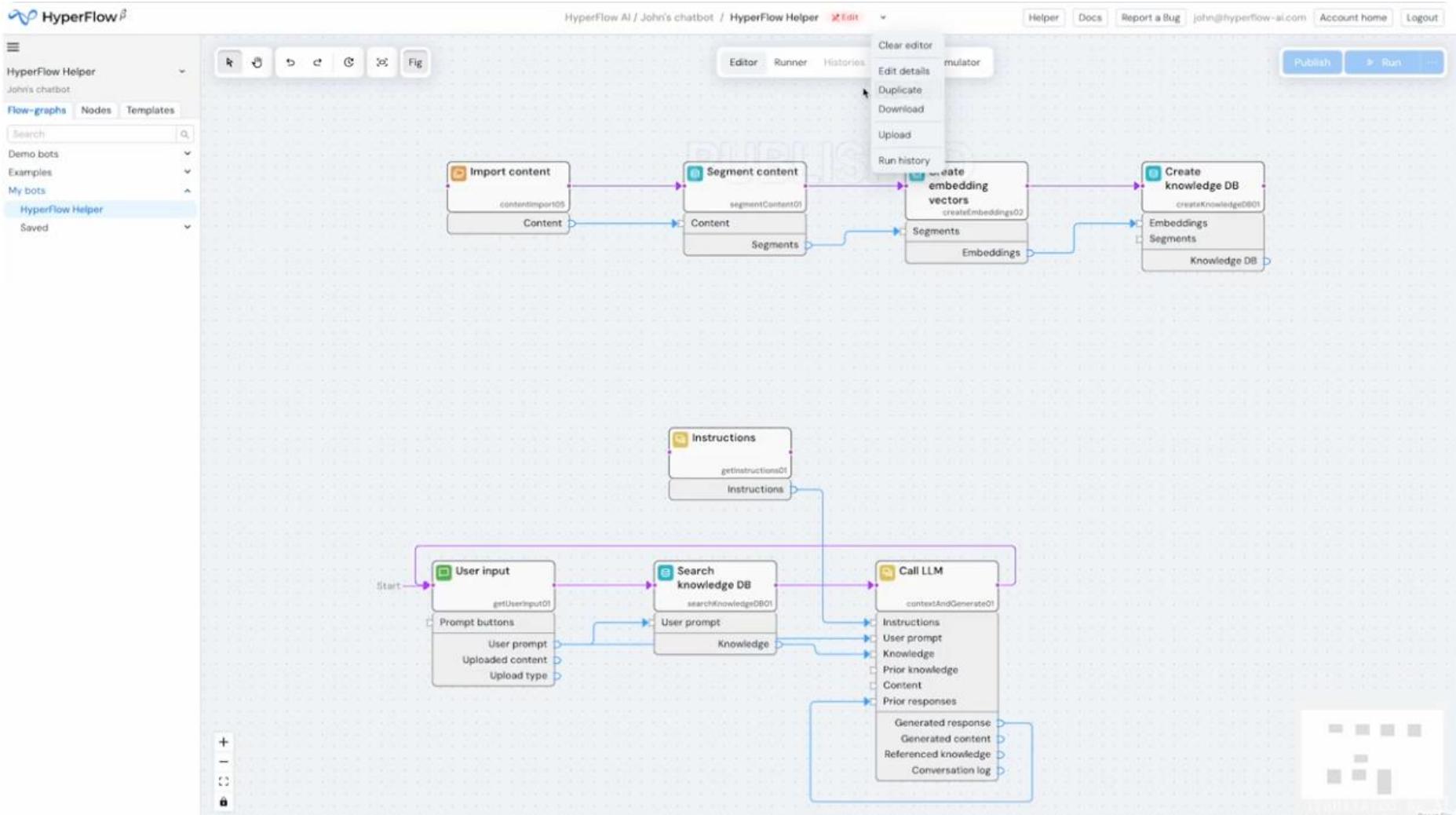
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**

John's chatbot →

Published flow-graphs	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	<a href="#">External test</a> API calls Embed code	<a href="#">Activate</a> <small>...</small>

[Bot mods →](#)  
[+ Create Project](#)

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
+ Invite member

**Projects**

John's chatbot →

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	External test API calls Embed code	Activate <small>...</small>

Bot mods →

+ Create Project

Actions menu (visible for the 'HyperFlow Helper' project):  
Test  
Pause  
Archive  
Enter maintenance

A screenshot of the HyperFlow AI web application. At the top, there's a navigation bar with links for Helper, Docs, Report a Bug, email (john@hyperflow-ai.com), Account home, and Logout. Below this is a 'Create Organization' button. The main area is titled 'Organizations' and shows a single organization named 'HyperFlow AI' with 'owner' status. Under 'Members', it lists 'john@hyperflow-ai.com: owner'. There's a '+ Invite member' button. The 'Projects' section shows a single project named 'John's chatbot' with a link to its details. The project details card includes tabs for 'Flow-graph' (HyperFlow Helper, ready, published 2025-01-21 15:13), 'Status' (External test, API calls, Embed code), and 'Actions' (Activate, Test, Pause, Archive, Enter maintenance). Below the project card is a '+ Create Project' button. A mouse cursor is hovering over the 'Actions' dropdown menu for the project card.

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow<sup>β</sup>

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)  
**Published flow-graphs**  

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	ready	2025-01-21 15:13	<a href="#">External tests</a> API calls Embed code	<a href="#">Activate</a> <span>[...]</span>

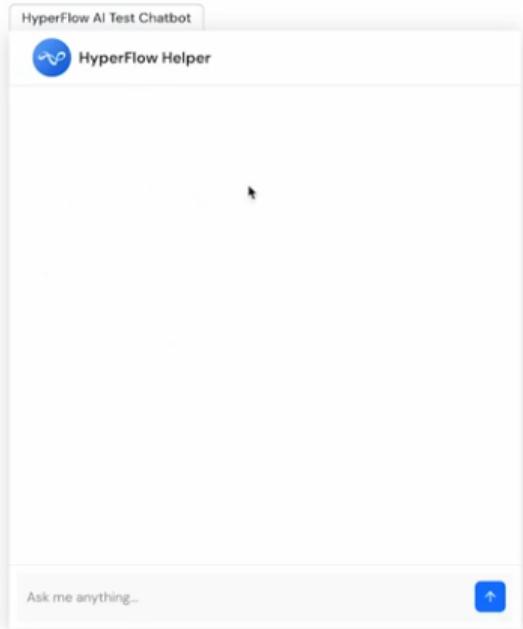
  
[Bot mods →](#)  
[+ Create Project](#)

<https://hyperflow-ai.com/testbot?n=6783a05a8226911e0afb0c7b>

# 프로덕션 AI 앱을 위한 플로우그래프 배포



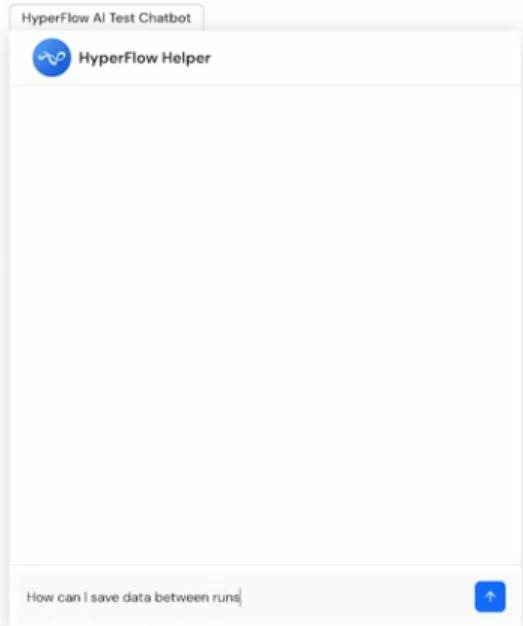
HyperFlow<sup>β</sup>



# 프로덕션 AI 앱을 위한 플로우그래프 배포



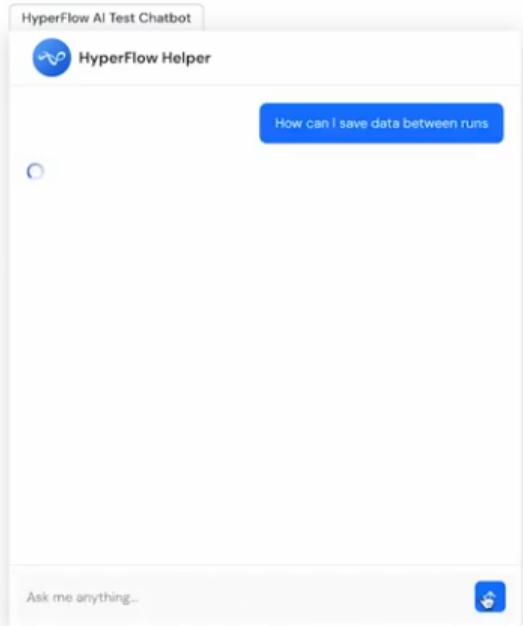
HyperFlow<sup>β</sup>



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow<sup>β</sup>



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow<sup>β</sup>

HyperFlow AI Test Chatbot

HyperFlow Helper

How can I save data between runs?

To save data between runs in HyperFlow, you can use the Key-value Store node. This node offers persistent storage capabilities, allowing you to store data that will be retained across multiple executions of your flow-graph.

**Steps to Save Data Using Key-value Store Node:**

- 1. Add a Key-value Store Node:**
  - In your flow-graph, locate the toolbar or node palette area.
  - Drag and drop the Key-value Store node onto your flow-graph workspace.
- 2. Configure the Node:**
  - Access the properties or configuration panel of the Key-value Store node.
  - Set the keys you want to use to store your data. Each piece of data should have a unique key.
- 3. Integrate with Other Nodes:**

Ask me anything...

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)  
**Published flow-graphs**  

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	testing	2025-01-21 15:13	<a href="#">External test</a> API calls Embed code	<a href="#">Activate</a> <small>[...]</small>

  
[Bot mods →](#)  
[+ Create Project](#)



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

**Embed code**

Example embed-code for a chatbot popup window for **HyperFlow Helper**

Note that the query-parameters, **param1** and **param2**, are optional and can be used to inject user-specific context into the running flow-graph. If supplied, they can be accessed as a JSON-encoded string using a **Launch parameters** flow-graph node.

```
<div class="chatbot">
 <iframe src="https://hyperflow-ai.com/bot?fg=6783a05a8225911e0afb0c7b¶m1=xxxxx¶m2=yyyy"></iframe>
</div>
```

**HyperFlow AI** owner

**Members**  
john@hyperflow-a  
+ Invite member

**Projects**  
John's chatbot →

**Published flow-graph**

Flow-graph

HyperFlow Helper testing 2025-01-21 15:13 External test API calls Embed code Activate

Bot mods →

+ Create Project



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Docu... / ... / Code Docs / Launch Parameters Made with

## Launch Parameters

☰ Menu

- Add Content
- Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- Data Range
- Data Set
- Data Transform
- Define Tool
- Import content
- Instructions
- Key-value Store
- Launch Parameters
- LLM Tool Agent
- Loop Controller

### Launch Parameters

Data flow node group

Launch parameters

Launch parameters

#### Overview

The Launch Parameters node makes available the `query_parameters` present on a HyperFlow Control API url, such as those used to launch HyperFlow embedded chatbots. Its output connector supplies the set of launch URL query parameters as a `JSON object` that can be accessed throughout the flow, allowing for dynamic parameter passing and flow configuration from the control API URLs.

This is particularly useful for passing user or context specific information to a production flow-graph running as an interactive, user-facing chatbot or assistant.

#### Description

This node extracts HyperFlow Control API URL query parameters and makes them available as a structured JSON object within the flow. It's particularly useful for configuring flows through URL parameters, enabling dynamic behavior based on how the flow is launched, and passing initial parameters to the application.

#### Table of Contents

- Launch Parameters
  - Overview
  - Description
  - Configuration Guidelines
  - Key Features
  - When to use
  - Best Practices
  - URL Parameter Format
  - Parameter Processing
  - Example usage
- Parameters
- Wireable Outputs



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Docu... / ... / Code Docs / Launch Parameters

☰ Menu

- Add Content
- Annotation
- Batch Content
- Call LLM
- Chat Output
- Choose a Path
- Create Embedding
- Create Knowledge DB
- Data Range
- Data Set
- Data Transform
- Define Tool
- Import content
- Instructions
- Key-value Store
- Launch Parameters
- LLM Tool Agent
- Loop Controller
- Match & Branch
- Match & Message
- Match & Prompt Button
- Match & Select
- Merge Inputs
- Message
- Message to Host
- Nested Loop Data
- Prompt Buttons
- Re-rank Search
- Real-time Knowledge Injector
- Retrieval Source
- Search Knowledge DB

## Launch Parameters

Data flow node group

### Overview

The Launch Parameters node makes available the `query_params` present on a HyperFlow Control API url, such as those used to launch HyperFlow embedded chatbots. Its output connector supplies the set of launch URL query parameters as a `JSON object` that can be accessed throughout the flow, allowing for dynamic parameter passing and flow configuration from the control API URLs.

This is particularly useful for passing user or context specific information to a production flow-graph running as an interactive, user-facing chatbot or assistant.

### Description

This node extracts HyperFlow Control API URL query parameters and makes them available as a structured JSON object within the flow. It's particularly useful for configuring flows through URL parameters, enabling dynamic behavior based on how the flow is launched, and passing initial parameters to the application.

### Configuration Guidelines

- Default Parameters**
  - Set default JSON object structure
  - Define fallback values
  - Configure required parameters
- Parameter Validation**
  - Validate parameter types
  - Handle missing parameters
  - Set parameter constraints

### Workflow Patterns

- Basic Parameter Flow

Table of Contents

- Launch Parameters
  - Overview
  - Description
  - Configuration Guidelines
    - Key Features
    - When to use
    - Best Practices
  - URL Parameter Format
  - Parameter Processing
  - Example usage
- Parameters
- Wireable Outputs

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

**Embed code**

Example embed-code for a chatbot popup window for **HyperFlow Helper**

Note that the query-parameters, **param1** and **param2**, are optional and can be used to inject user-specific context into the running flow-graph. If supplied, they can be accessed as a JSON-encoded string using a **Launch parameters** flow-graph node.

```
<div class="chatbot">
 <iframe src="https://hyperflow-ai.com/bot?fg=6783a05a8225911e0afb0c7b¶m1=xxxxx¶m2=yyyy"></iframe>
</div>
```

**HyperFlow AI**

Members

john@hyperflow-a

+ Invite member

Projects

John's chatbot →

Published flow-graph

Flow-graph

HyperFlow Helper testing 2025-01-21 15:13 External test API calls Embed code Activate ...

Bot mods →

+ Create Project

owner

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)  
**Published flow-graphs**  

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	testing	2025-01-21 15:13	<a href="#">External test</a> <small>API calls</small> Embed code	<a href="#">Activate</a> <small>...</small>

  
[Bot mods →](#)  
[+ Create Project](#)

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Helper Example API calls

Example Control Start call to start a session with the flow-graph HyperFlow Helper

Note that the `queryParams` field is optional and can be used to inject session-specific context into the running flow-graph. If supplied, they can be accessed as a JSON object using a `Launch parameters` flow-graph node.

```
const response = await axios.post('https://hyperflow-ai.com/api/flowgraph/control/start', {
 flowGraphID: "6783a05a8225911e0afb0c7b",
 queryParams: {
 param1: "xxxxx",
 param2: "yyyy",
 }
});
```

Example Control Progress call to continue a session with the flow-graph HyperFlow Helper

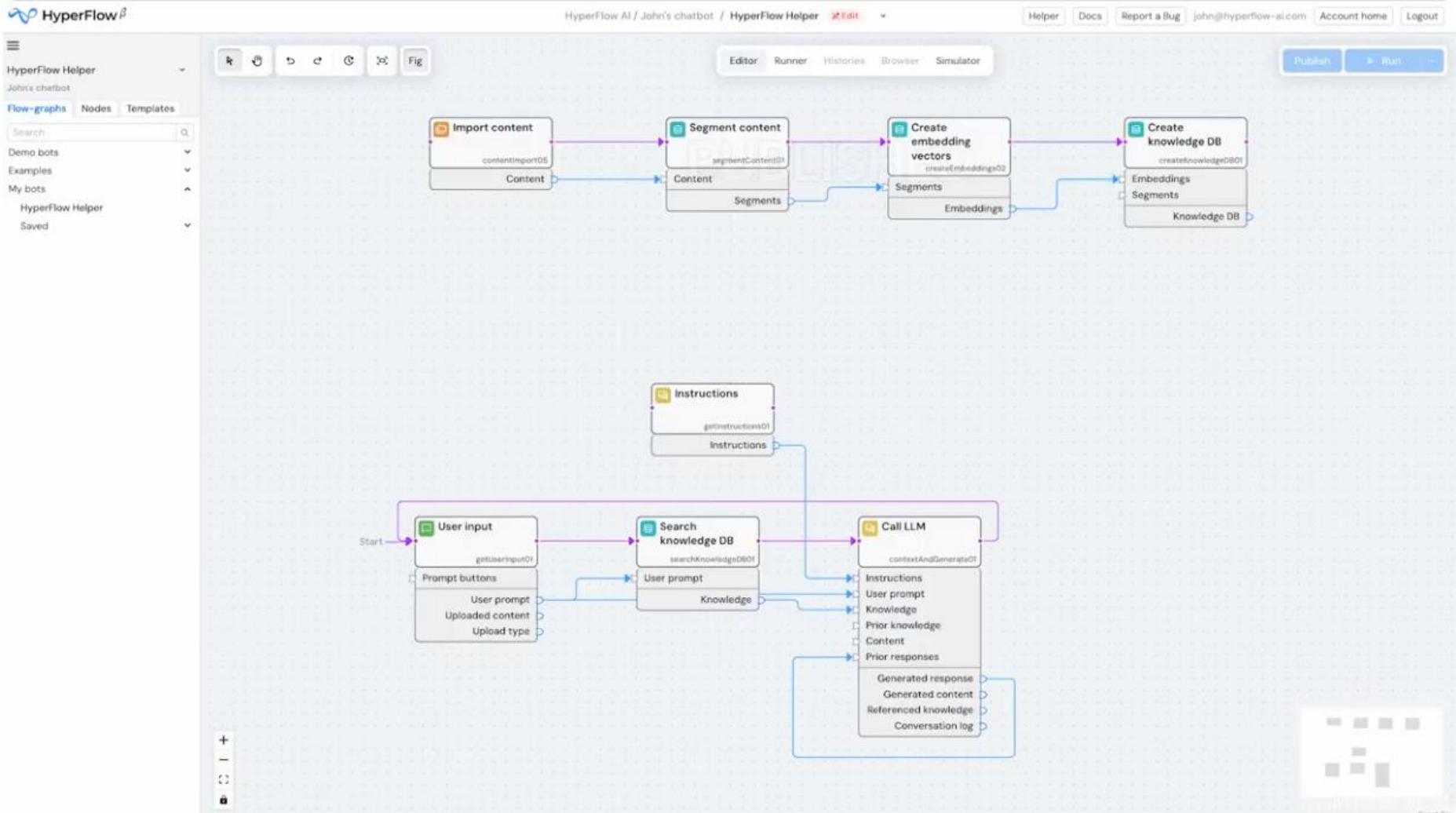
Note that the `sessionId` field required and comes from the response of the Control Start call.

```
+ i const response = await axios.post('https://hyperflow-ai.com/api/flowgraph/control/progress', {
 sessionId: "ssssss",
 currentStepIndex: nnn,
 progress: { ... }
});
```

HyperFlow Helper testing 2025-01-21 15:13 External test API calls Embed code Activate

Bot mode → + Create Project

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Beta Docu... / Custom AI App Control API Reference... Made with

# HyperFlow

## Custom AI App Control API Reference

⌄ 5 more properties

Here is all the information you need to interact with your newly published flow-graphs in any custom chatbots or other AI apps you might build. The HyperFlow Control API is a classic HTTP RESTful API, the end-point URLs for which are available in the Published Flow-graph Manager section of the Project Manager page.

HyperFlow AI also provides ready-made interactive chatbots that you can embed in your host applications or web-sites, and these chatbots also use this Control API. See the [Embed code](#) for these in the published flow-graph manager.

Finally, there is a [Chatbot Simulator](#), available via the Simulator mode button in the HyperFlow IDE that can be attached to a ready-to-publish flow-graph for testing that shows a detailed log of the API calls between flow-graph and AI app, which can be helpful in understanding the Control API in use.

Introduction  
Getting Started  
Publishing your production-ready flow-graph  
Control API Sequencing  
General call forms  
Understanding API responses  
Data field  
Responses  
Start

## Introduction

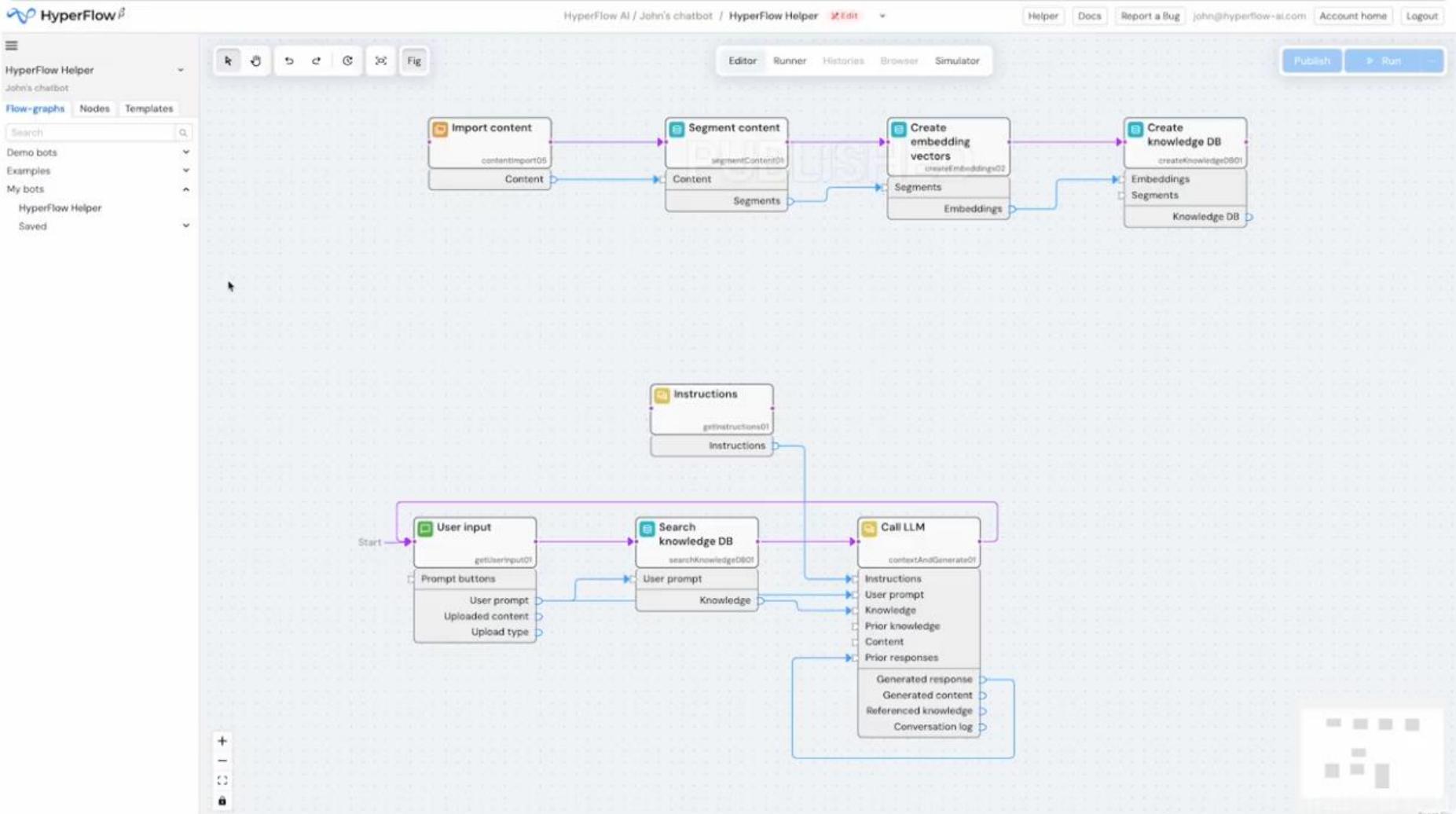
The Control API provides comprehensive access to a published flow-graphs functionality, allowing it to drive and interact with custom chatbots, AI assistants and other AI applications you might choose to develop to provide custom user-interfaces to your HyperFlow apps.

This documentation covers the available API calls and their returned properties. If you have any questions, please reach visit the HyperFlow Discord support server or email [team@hyperflow-ai.com](mailto:team@hyperflow-ai.com).

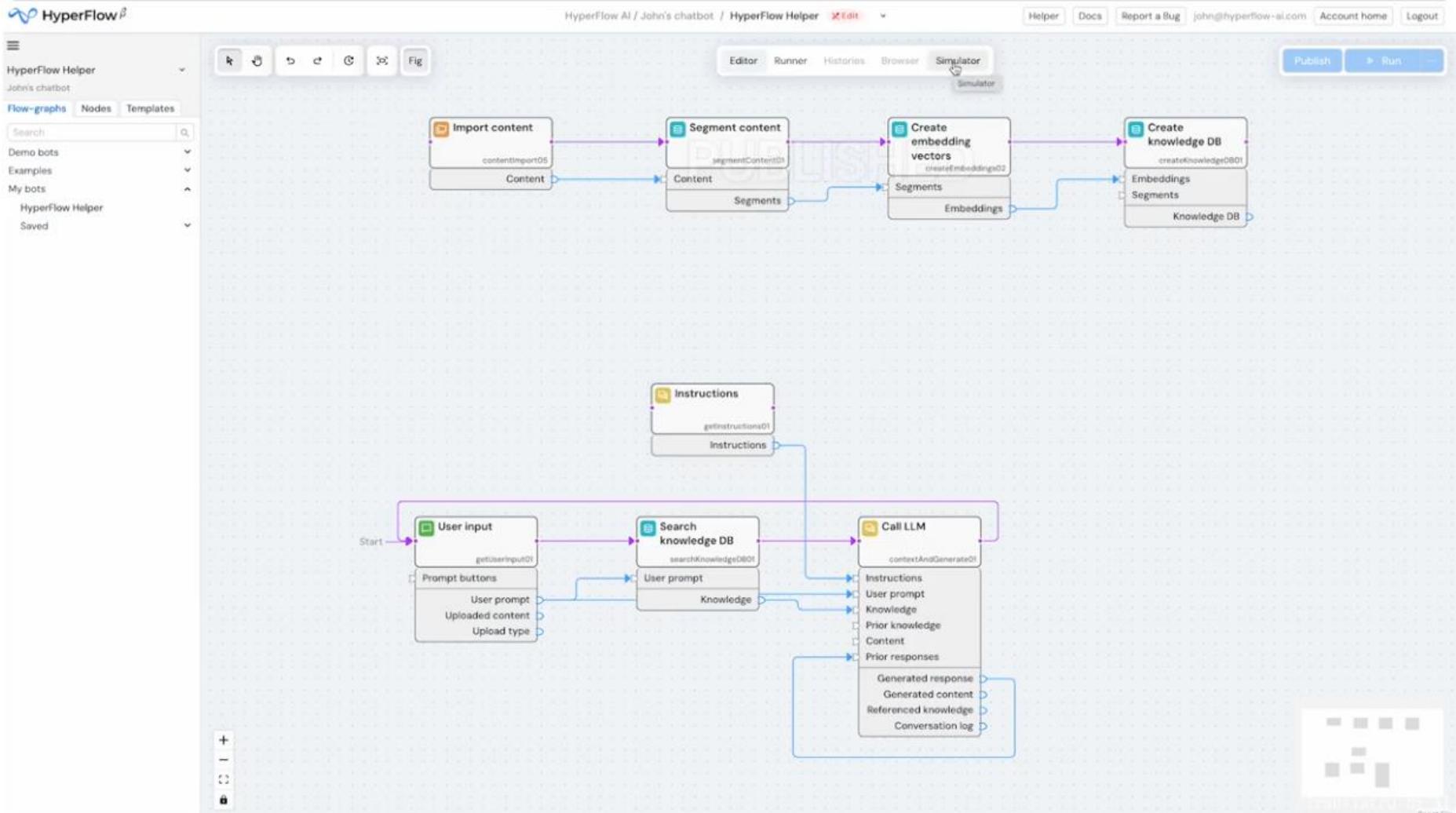
## Getting Started



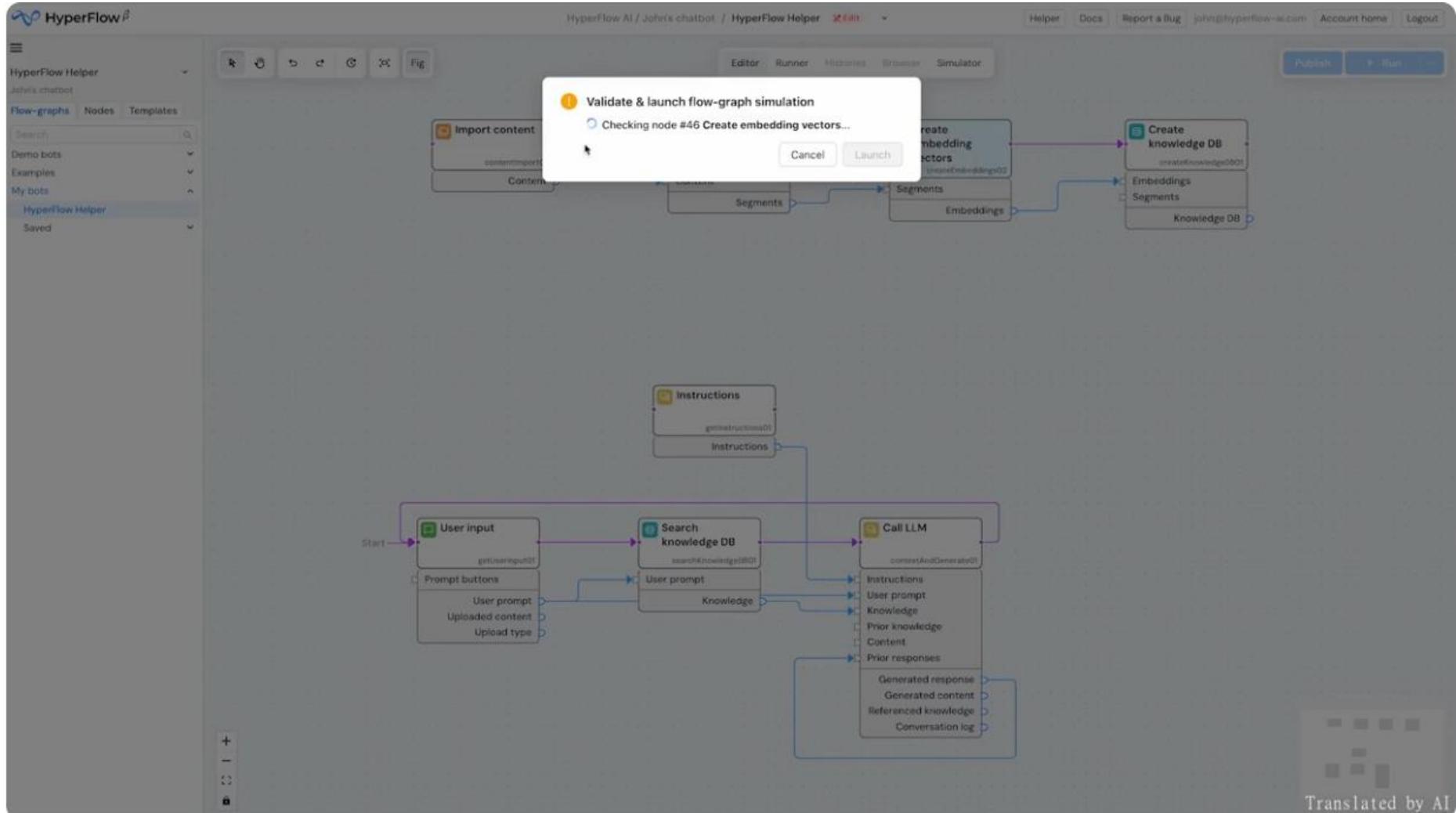
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

HyperFlow AI / John's chatbot / HyperFlow Helper Edit

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

API log

2025-01-21T23:19:12.034Z: /api/flowgraph/control/start

Post body:

```
flowGraphID: '6783a05a822591e0afb0c7b', queryParams: { 'fg': '6783a05a822591e0afb0c7b' }
```

Progress:

Response:

```
sessionId: 67902b6fd2ac69c60a89676e, step: 1
- start: flowGraphName: 'HyperFlow Helper'
- interaction: interactionSpec: [{ type: 'prompt', label: 'User input', pathName: 'prompt.text', notWireable: true, endUserInteractive: true }]
```

HyperFlow AI Chatbot Simulator

Greetings from HyperFlow Helper!

Can I give

# 프로덕션 AI 앱을 위한 플로우그래프 배포



 **HyperFlow** βHyperFlow AI / John's chatbot / HyperFlow Helper [Edit](#)Helper Docs Report a Bug [john@hyperflow-ai.com](#)Account homeLogout

---

API log

2025-01-21T23:19:12.034Z: /api/flowgraph/control/start

Post body:

flowGraphID: "6783a05a822591le0afb0c7b", queryParams: {"lg": "6783a05a822591le0afb0c7b"}

Progress:

Response:

sessionID: 67902b6fd2ac69c60a89676e, step: 1

- start: flowGraphName: "HyperFlow Helper"

- interaction: interactionSpec: {"type": "prompt", "label": "User input", "pathName": "prompt.text", "notWireable": true, "endUserInteractive": true}

2025-01-21T23:19:48.480Z: /api/flowgraph/control/progress

Post body:

sessionID: "67902b6fd2ac69c60a89676e", currentStepIndex: 1

Progress:

- prompt: data: [{"text": "Can I generate images?", "paramSpec": {"type": "prompt", "label": "User input", "pathName": "prompt.text", "notWireable": true, "endUserInteractive": true}}]

HyperFlow AI Chatbot Simulator

Greetings from HyperFlow Helper!

Can I generate images?

?

?

>

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow <sup>β</sup>

HyperFlow AI / John's chatbot / HyperFlow Helper [Edit](#)

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

**API log**

2025-01-21T23:19:12.034Z: /api/flowgraph/control/start

Post body:

```
flowGraphID: "6783a05a822591e0afb0c7b", queryParams: {"tg": "6783a05a822591e0afb0c7b"}
Progress:
Response:
sessionID: 67902b6fd2ac69c60a89676e, step: 1
- start: flowGraphName: "HyperFlow Helper"
- interaction: interactionSpec: {"type": "prompt", "label": "User input", "pathName": "prompt.text", "notWireable": true, "endUserInteractive": true}
```

2025-01-21T23:19:48.480Z: /api/flowgraph/control/progress

Post body:

```
sessionID: "67902b6fd2ac69c60a89676e", currentStepIndex: 1
Progress:
- prompt: data: {"text": "Can I generate images?", "paramSpec": {"type": "prompt", "label": "User input", "pathName": "prompt.text", "notWireable": true, "endUserInteractive": true}}
```

2025-01-21T23:19:52.341Z: /api/flowgraph/control/progress

Response:  
sessionID: 67902b6fd2ac69c60a89676e, step: 4  
- busyMessage: text: {"type": "text", "value": "", "label": "", "locked": true}

2025-01-21T23:19:52.347Z: /api/flowgraph/control/progress

Post body:

```
sessionID: "67902b6fd2ac69c60a89676e", currentStepIndex: 4
Progress:
- updatePoll:
2025-01-21T23:19:56.789Z: /api/flowgraph/control/progress
```

Response:  
sessionID: 67902b6fd2ac69c60a89676e, step: 5  
- generatedText: text: "Yes, you can generate images in HyperFlow. Image generation is one of the LLM Services available, enabling you to create and manipulate images. To get started, you may need to use specific nodes or components in your flow-graph that are tailored for image generation tasks."  
- references: references: [{"document": "...", "pages": [{"page": "...", "images": [{"data": "image/gif;base64,R0lGODlhAQABIAAAQ||wAAACH5BAEAAAALAAAAAABAAEAAAICRAEAOw=="}]}, {"links": [{"title": "Call LLM", "url": "https://docs.hyperflow-ai.com/Call-LLM-158771982bb78lb3b7d2dc6b70689fad?pvs=25"}, {"title": "Import Services", "url": "https://docs.hyperflow-ai.com/Import-Services-158771982bb781428cb9dfba416545d2?pvs=25"}]}]]}  
- interaction: interactionSpec: {"type": "prompt", "label": "User input", "pathName": "prompt.text", "notWireable": true, "endUserInteractive": true}

**HyperFlow AI Chatbot Simulator**  
Getting started with HyperFlow Helper

Can I generate images?

Yes, you can generate images in HyperFlow. Image generation is one of the LLM Services available, enabling you to create and manipulate images. To get started, you may need to use specific nodes or components in your flow-graph that are tailored for image generation tasks.

You can find image generation features under the "Import Services" menu, where it's represented alongside other media generation capabilities like text, audio, and video generation.

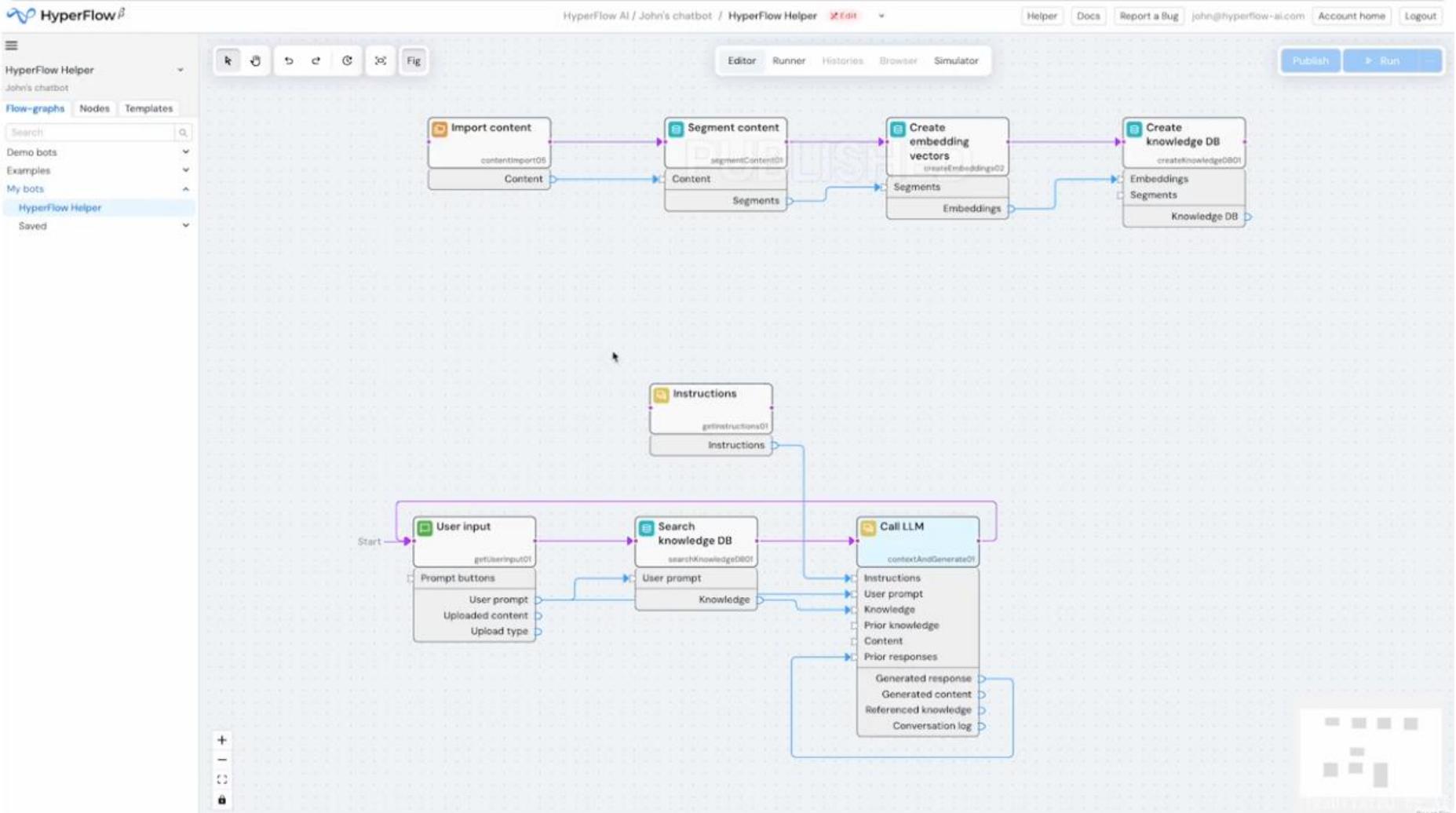
For a practical approach to image generation, consider using agentic workflows where multiple LLMs can cooperate, such as one LLM generating images and another reviewing and suggesting improvements.

References:

Call LLM Import Services

Call LLM Import Services

# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
+ Invite member

**Projects**  
John's chatbot →

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	testing	2025-01-21 15:13	External test API calls Embed code	Activate ...

Bot mods →  
+ Create Project

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

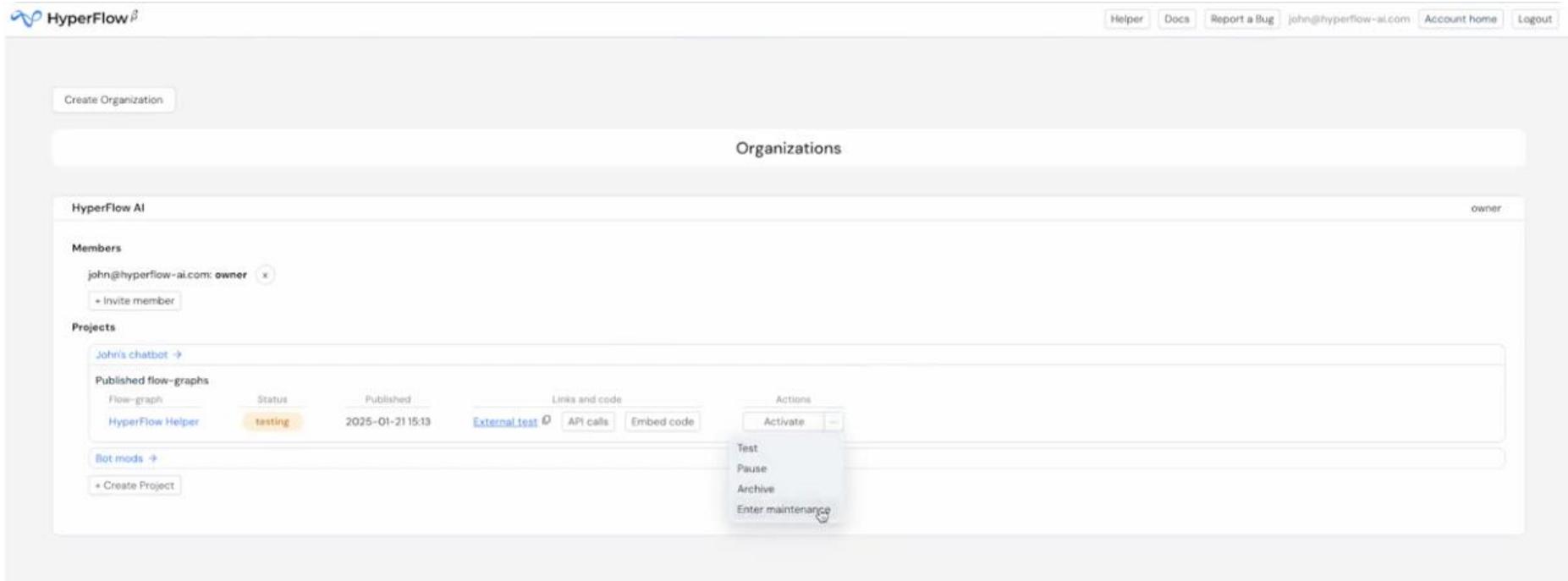
**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
John's chatbot →  
**Published flow-graphs**

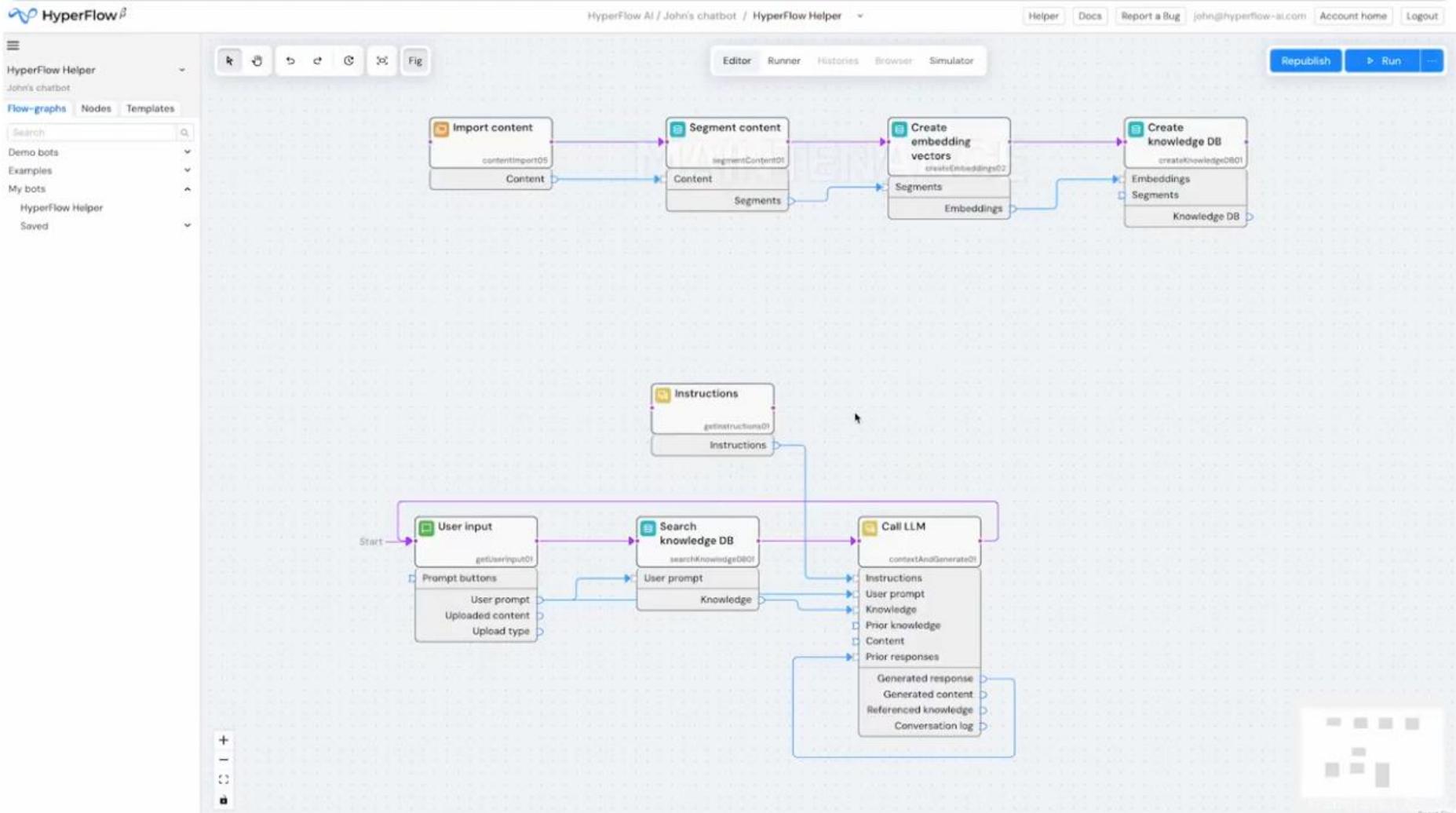
Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	testing	2025-01-21 15:13	<a href="#">External test</a> API calls Embed code	<a href="#">Activate</a> <small>-</small>

**Bot mods** →  
[+ Create Project](#)

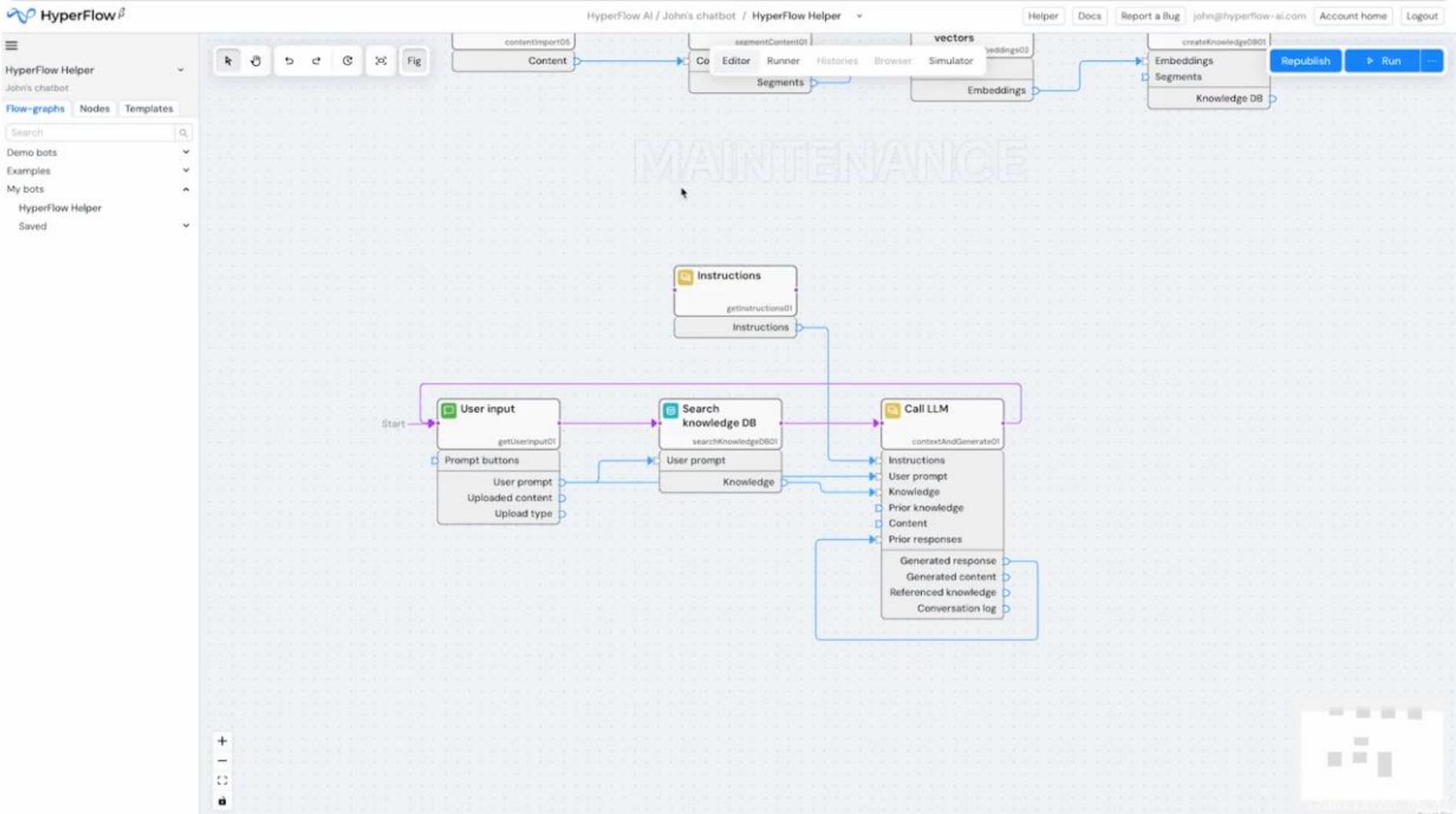
Actions menu (dropdown): Test, Pause, Archive, Enter maintenance



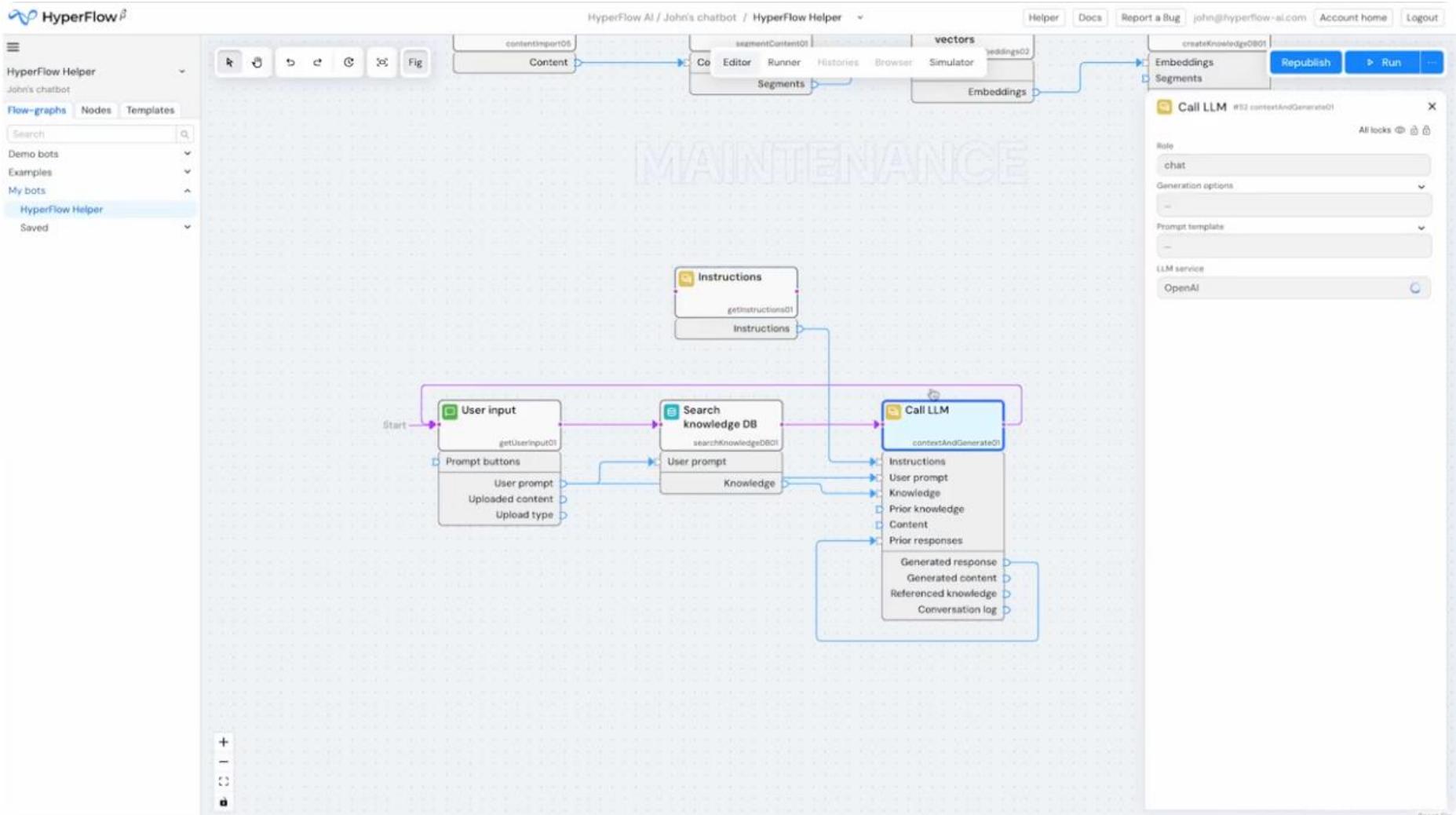
# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow Helper

Validate & confirm flow-graph publish  
Checking node #45 Import content...

Cancel Confirm

ContentImport01 Content ContentContent01 Content

vectors vectors vectors

Helper Editor Runner Monitor Simulator

Report a Bug john@hyperflow.ai Account home Logout

Flow-graphs Nodes Templates

Demo bots Examples My bots HyperFlow Helper Saved

Instructions getInstruction01 Instructions

User input get/setInput01 Prompt buttons User prompt Uploaded content Upload type

Search knowledge DB searchKnowledge0001 Knowledge

Call LLM generateAnswer001 Instructions User prompt Knowledge Prior knowledge Content Prior responses Generated response Generated content Referenced knowledge Conversation log

EmbeddingContent01 Embedding Segments Knowledge DB

createKnowledge0001 Republish Run

Start

Translated by AI

```
graph LR
 Start((Start)) --> UI[User input]
 UI -- "get/setInput01" --> UIButtons[Prompt buttons]
 UIButtons -- "User prompt, Uploaded content, Upload type" --> UIButtons
 UI -- "get/setInput01" --> SKD[Search knowledge DB]
 SKD -- "searchKnowledge0001" --> CallLLM[Call LLM]
 CallLLM -- "generateAnswer001" --> GR[Generated response]
 CallLLM -- "generateAnswer001" --> GC[Generated content]
 CallLLM -- "generateAnswer001" --> RKN[Referenced knowledge]
 CallLLM -- "generateAnswer001" --> CL[Conversation log]
 Instructions[Instructions getInstruction01] -- "Instructions" --> CallLLM
 Instructions -- "Instructions" --> CallLLM
```

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
+ Invite member

**Projects**  
John's chatbot →  
**Published flow-graphs**  
Flow-graph Status Published Links and code Actions  
HyperFlow Helper active 2025-01-21 15:24 External test API calls Embed code Pause ...

Bot mode →  
+ Create Project

A screenshot of the HyperFlow AI web application. At the top, there's a navigation bar with links for Helper, Docs, Report a Bug, email (john@hyperflow-ai.com), Account home, and Logout. Below the navigation is a "Create Organization" button. The main area is titled "Organizations" and shows a single organization named "HyperFlow AI" owned by "john@hyperflow-ai.com". Under "Members", it lists "john@hyperflow-ai.com" as the owner with a delete icon. There's a "+ Invite member" button. Under "Projects", it shows a link to "John's chatbot". The "Published flow-graphs" section displays one entry: "HyperFlow Helper" (Flow-graph, active status, published 2025-01-21 15:24, external test link, API calls, embed code, pause button, more options). Below this is a "Bot mode" section with a "+ Create Project" button.

# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**  
[John's chatbot →](#)  
**Published flow-graphs**

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	active	2025-01-21 15:24	<a href="#">External test</a> <small>ID</small> API calls Embed code	<a href="#">Pause</a> <small>...</small>

[Bot mods →](#)  
[+ Create Project](#)



# 프로덕션 AI 앱을 위한 플로우그래프 배포



HyperFlow β

Helper Docs Report a Bug john@hyperflow-ai.com Account home Logout

Create Organization

### Organizations

HyperFlow AI owner

**Members**  
john@hyperflow-ai.com: owner x  
[+ Invite member](#)

**Projects**

John's chatbot →

Flow-graph	Status	Published	Links and code	Actions
HyperFlow Helper	active	2025-01-21 15:24	<a href="#">External test</a> <small>0</small> API calls Embed code	<a href="#">Pause</a> <small>...</small>

[Bot mods →](#)

[+ Create Project](#)

# Publishing a production flow-graph

- Style editor for the built-in chatbot
- Chatbot plugins for website builders
- Open-source React chatbot components

# 프로덕션 AI 앱을 위한 플로우그래프 배포



## Publishing a production flow-graph

- ▶ Style editor for the built-in chatbot
- ▶ Chatbot plugins for website builders
- ▶ Open-source React chatbot components

### 플로우그래프 제품 배포

- 내장 챗봇용 스타일 편집기
- 웹사이트 제작자를 위한 챗봇 플러그인
- 오픈 소스 React 챗봇 구성 요소





김 진 수

CEO, Data Actionist

100-791 서울특별시 중구 청파로 463번지 3F BigData R&D Center

CP. 010-5670-3847      Tel. 02-360-4047      Fax. 02-360-4899

E-mail. [bigpycraft@gmail.com](mailto:bigpycraft@gmail.com)

<http://www.biggycraft.com>

수고하셨습니다!