

유용한 Tools



Browser Use



Enable AI to control your browser 🤖

Stars Discord 1.7k online Cloud Documentation Follow @Gregor Follow @Magnus Weave 25.0

🌐 Browser-use is the easiest way to connect your AI agents with the browser.

💡 See what others are building and share your projects in our [Discord](#)! Want Swag? Check out our [Merch store](#).

💡 Skip the setup - try our **hosted version** for instant browser automation! [Try the cloud](#) ☁️.

Quick start

With pip (Python >= 3.11):

```
pip install browser-use
```



Install Playwright:

```
playwright install chromium
```





Browser Use

```
from langchain_openai import ChatOpenAI
from browser_use import Agent
import asyncio
from dotenv import load_dotenv
load_dotenv()

async def main():
    agent = Agent(
        task="Compare the price of gpt-4o and DeepSeek-V3",
        llm=ChatOpenAI(model="gpt-4o"),
    )
    await agent.run()

asyncio.run(main())
```

`.env` file

```
OPENAI_API_KEY=
ANTHROPIC_API_KEY=
AZURE_ENDPOINT=
AZURE_OPENAI_API_KEY=
GEMINI_API_KEY=
DEEPSEEK_API_KEY=
```

Browser Use 실습



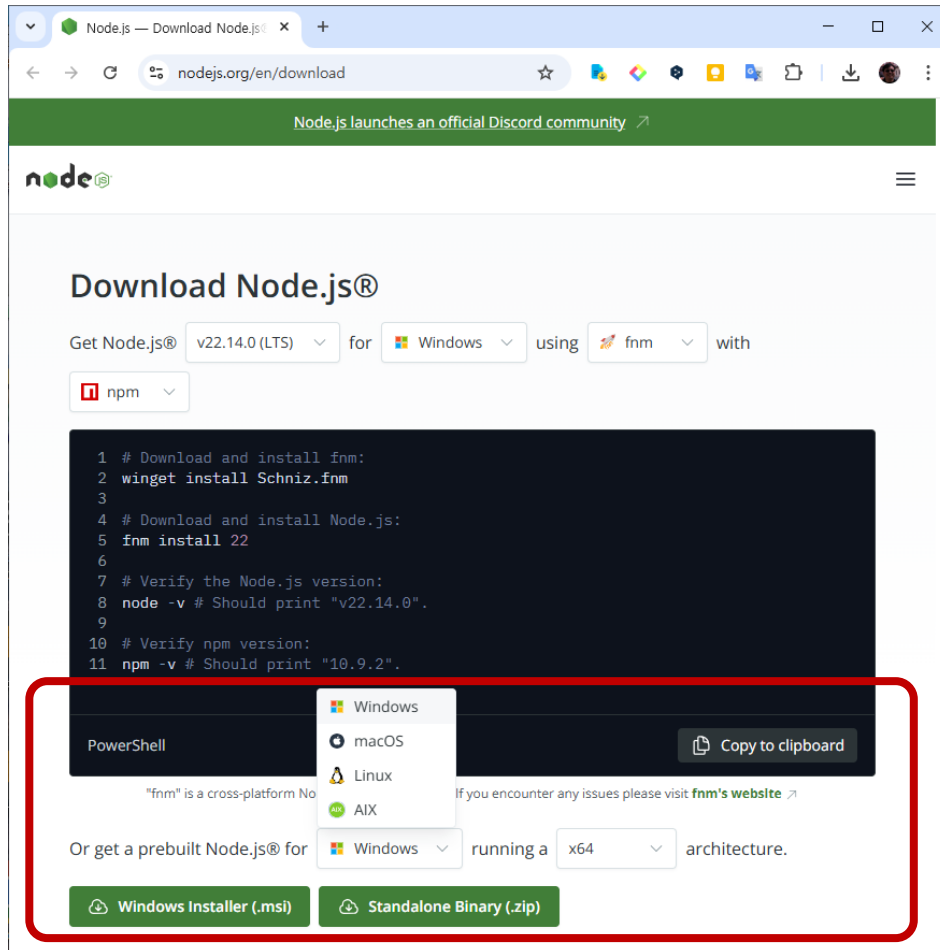
browser_use_agent



Flowise 설치

■ NodeJS 설치

<https://nodejs.org/en/download>



■ Flowise 설치

<https://github.com/FlowiseAI/Flowise>

1. Flowise 설치

`npm install -g flowise`

2. Flowise 시작

`npx flowise start`

3. <http://localhost:3000> 접속

Flowise 사용

Flowise 웹UI 에서 사용자 정의 LLM Flow를 Drag & drop 으로 구성

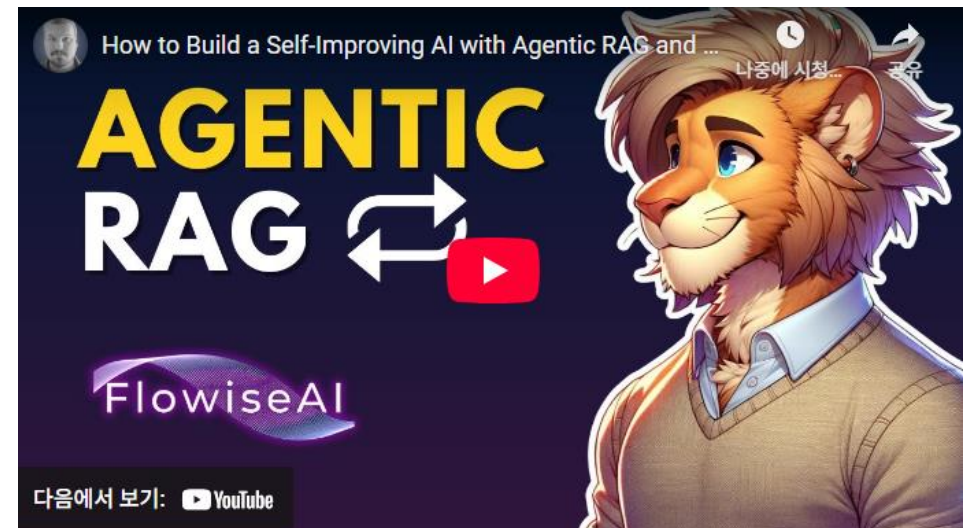
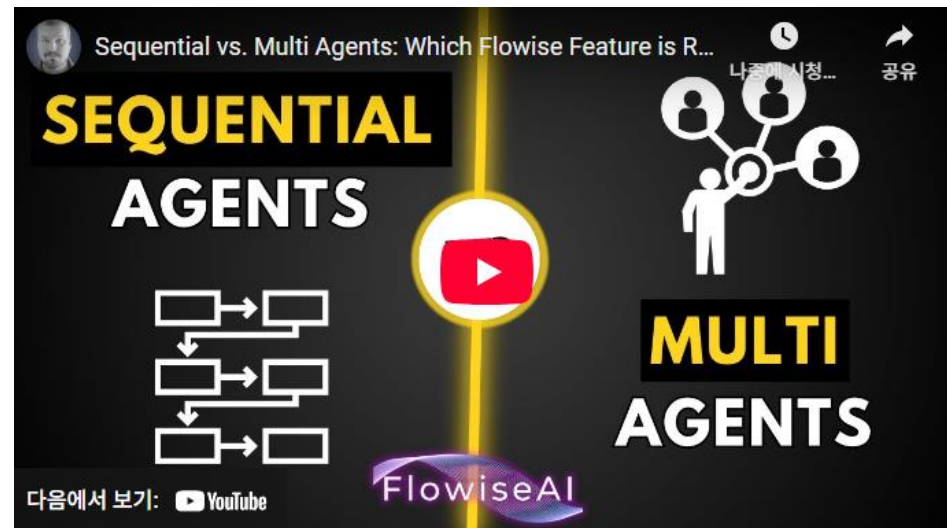
The screenshot displays the Flowise web interface for building custom LLM flows. The interface includes a top navigation bar with a back arrow, a workspace identifier 'wf1', and a plus icon for creating new flows. On the left, a blue circular button with a plus sign and a cursor icon is used to add components. The main workspace is a grid where three components are placed:

- Serp API:** A component for web search. It has an 'Inputs' section with a 'Serp Api Key' field and an 'Output' section. A 'SerpAPI' connector point is visible at the bottom right.
- OpenAI:** A component for OpenAI's LLM. It has an 'Inputs' section with 'OpenAI Api Key' and 'Model Name' (set to 'text-davinci-003') fields, and a 'Temperature' slider set to '0'. It has an 'Output' section and an 'OpenAI' connector point at the bottom right.
- MRLK Agent for LLMs:** A component for an LLM agent. It has an 'Inputs' section with 'Allowed Tools' and 'LLM Model' fields, and an 'Output' section. An 'AgentExecutor' connector point is visible at the bottom right.

At the bottom center, there are icons for zooming in (+), zooming out (-), and a lock icon. The bottom right corner shows the text 'React Flow'.



<https://youtube.be/6LbvGTbS0BE?si=IQK-nrzaSioalvHf>





CURSOR

<https://www.cursor.com/>

Cursor - The AI Code Editor

cursor.com

CURSOR PRICING FEATURES ENTERPRISE BLOG FORUM CAREERS SIGN IN DOWNLOAD

The AI Code Editor

Built to make you extraordinarily productive, Cursor is the best way to code with AI.

DOWNLOAD FOR WINDOWS ALL DOWNLOADS

mod.rs

Implement the cleanup function for the transport stack. Do not make the upgrade listeners optional.

Accept Reject Follow-up instructions... ↵

```
72 pub(crate) struct TransportStack {
73     l4: ListenerEndpoint,
74     tls: Option<Arc<Acceptor>>,
75     // listeners sent from the old process for graceful upgrade
76     #[cfg(unix)]
77     upgrade_listeners: Option<ListenFds>,
78     upgrade_listeners: ListenFds
79 }
```

CHAT COMPOSER

mod.rs Current File

Could you make it easier to switch certificates in the transport listeners?

I'll help modify the code to make certificate switching more flexible. The main changes will be to enhance the `TlsAccept` trait and modify how certificates are handled in the `TlsSettings`. Here are the key changes:

mod.rs Ask Copy Apply

// ... existing code ...

Thank you 😊





수고하셨습니다. 감사합니다.