

AI Dev 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



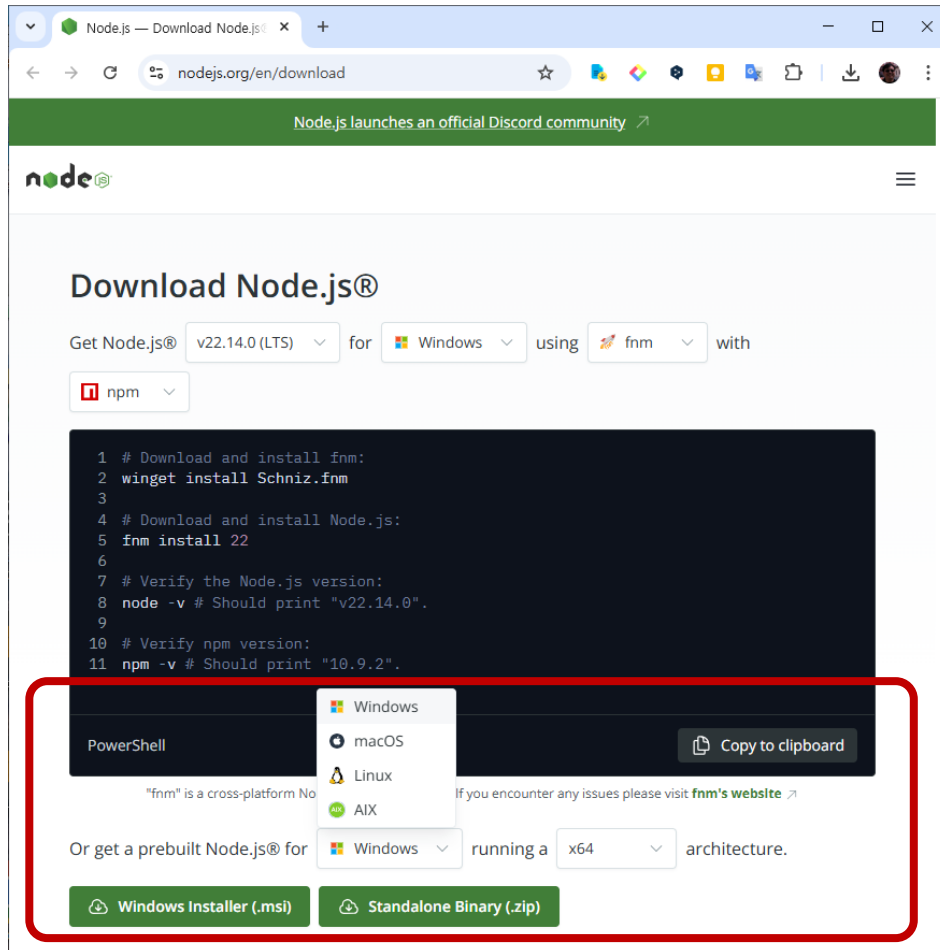
browser_use_agent



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

< * wf1

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

+

Serp API

Inputs

Serp Api Key *

Output

SerpAPI

OpenAI

Inputs

OpenAI Api Key *

Model Name

text-davinci-003

Temperature

0

Output

OpenAI

MRLK Agent for LLMs

Inputs

Allowed Tools *

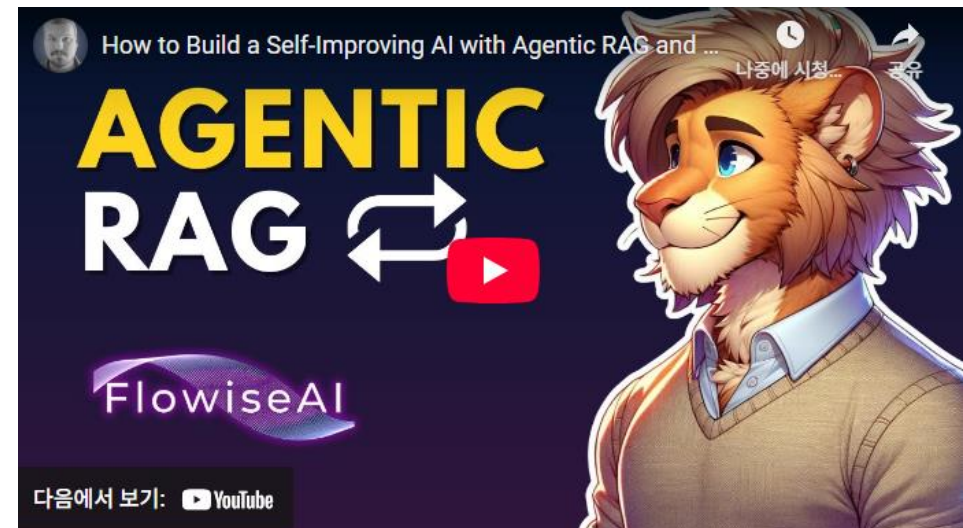
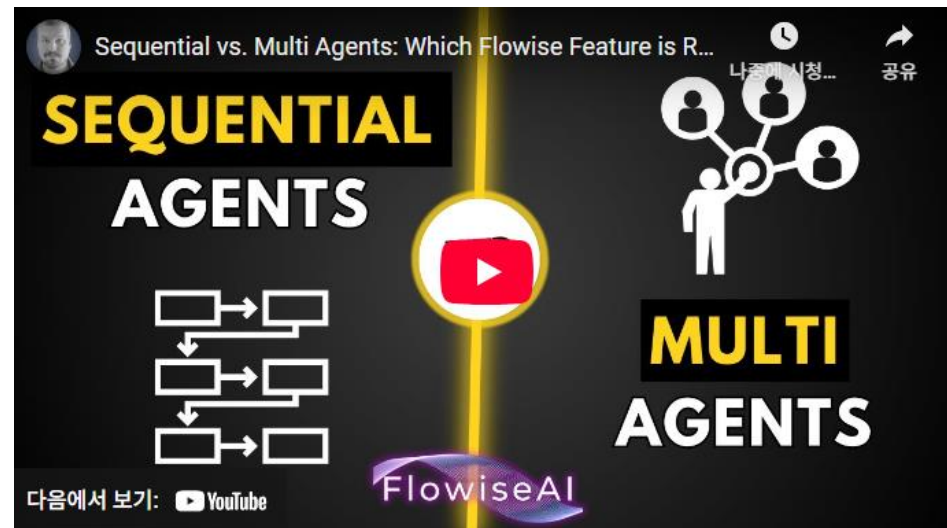
LLM Model *

Output

AgentExecutor

+ - 🔗 🔒

React Flow





CURSOR

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

The screenshot displays the Cursor website and its integrated code editor. The website header includes the Cursor logo and navigation links for PRICING, FEATURES, ENTERPRISE, BLOG, FORUM, and CAREERS. It also features 'SIGN IN' and 'DOWNLOAD' buttons. The main content area has a large heading 'The AI Code Editor' and a subtext 'Built to make you extraordinarily productive, Cursor is the best way to code with AI.' Below this are 'DOWNLOAD FOR WINDOWS' and 'ALL DOWNLOADS' buttons. The bottom portion of the image shows the Cursor editor interface with a Rust file named 'mod.rs' open. A context menu is visible over the code, offering to 'Accept', 'Reject', or provide 'Follow-up instructions...'. The right sidebar contains a 'CHAT' and 'COMPOSER' panel with a conversation about enhancing certificate switching in the transport listeners.

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... ⌘K

```
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.