Kasadara

Cursor AI 2.0+: An Advanced Agent-Driven Development Environment

From AI-assisted editing to autonomous, multi-agent software engineering workflows

Kasadara

Kasadara

Table of Content

1. Executive Summary

- Showcases **agent-centric development workflows** in Cursor
- Demonstrates **parallel multi-agent execution and evaluation**
- Illustrates **MCP server integrations** with external context
- Highlights **Cloud Agent handoff and background execution**
- Captures **agent-driven debugging and refactoring flows**
- Presents **end-to-end autonomous development** visually

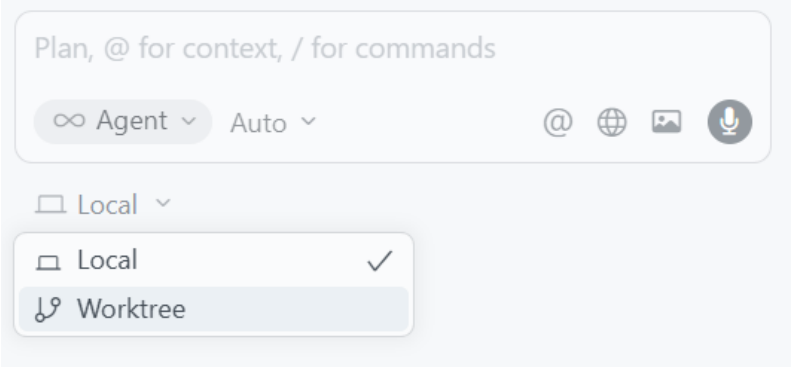
2. Key Purposes

- Visually document **multi-agent orchestration**
- Showcase **MCP integration workflows**
- Capture **Cloud Agent execution flows**
- Highlight **debugging and auto-fix screenshots**
- Present **Cursor UI and agent layouts**
- Provide **visual evidence of autonomy**

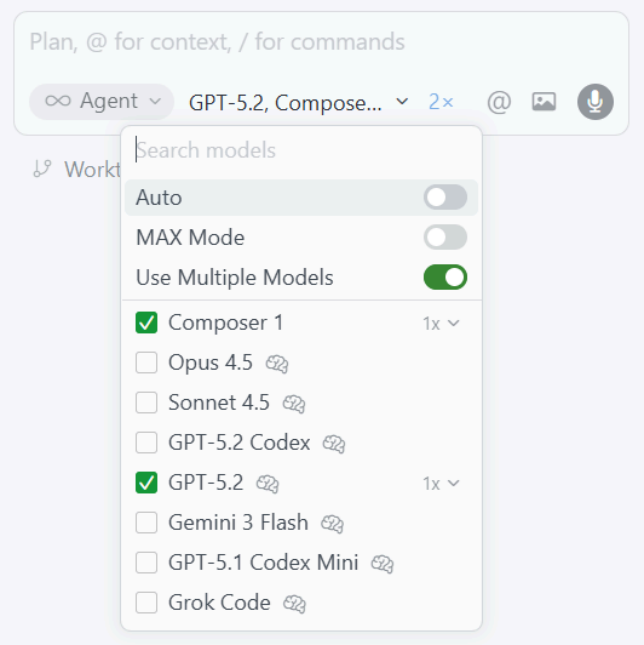
3. Multiple Agents

To run multiple agents, you must first switch from **Local** to **Worktree**, which creates an isolated copy of the project without affecting the main branch

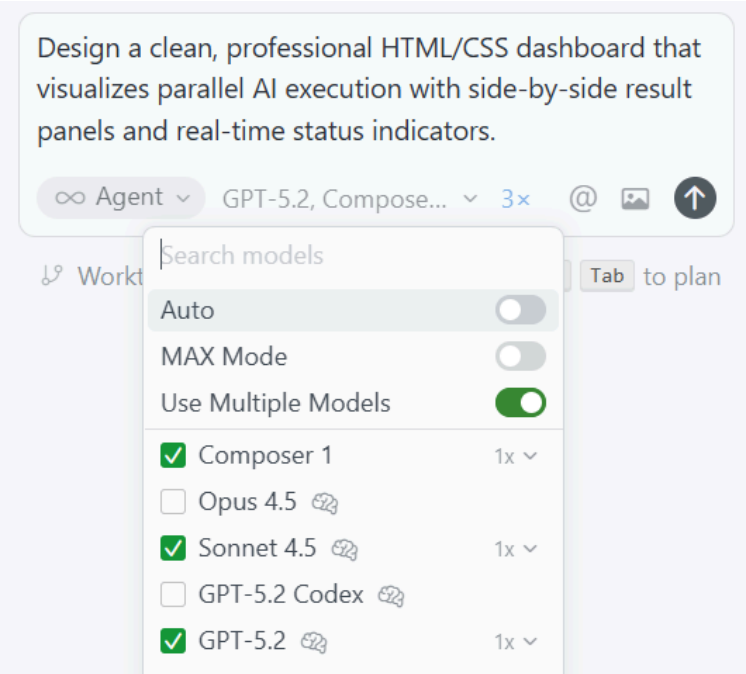
In the agent window, switch from **Local** to **Worktree**



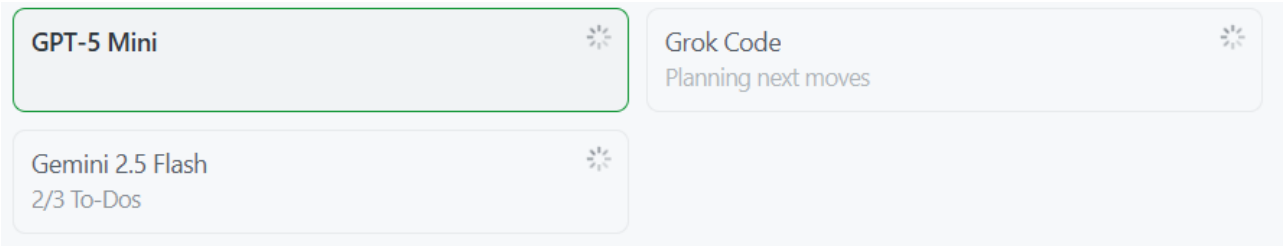
After switching to **Worktree**, enable **Use Multiple Models** in the model settings. You can now run multiple agents simultaneously



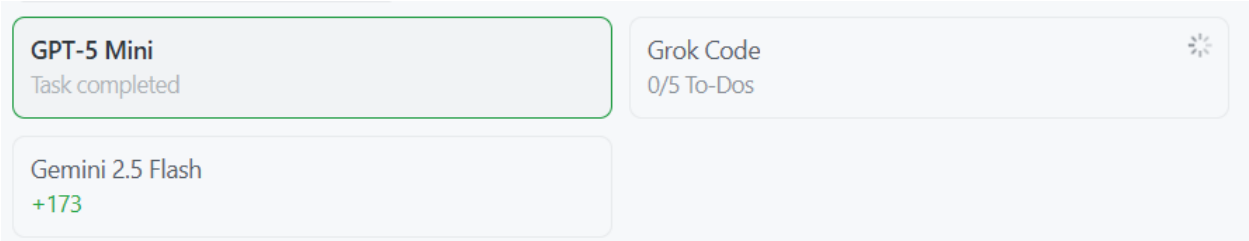
Let us now see how multiple models can be run simultaneously



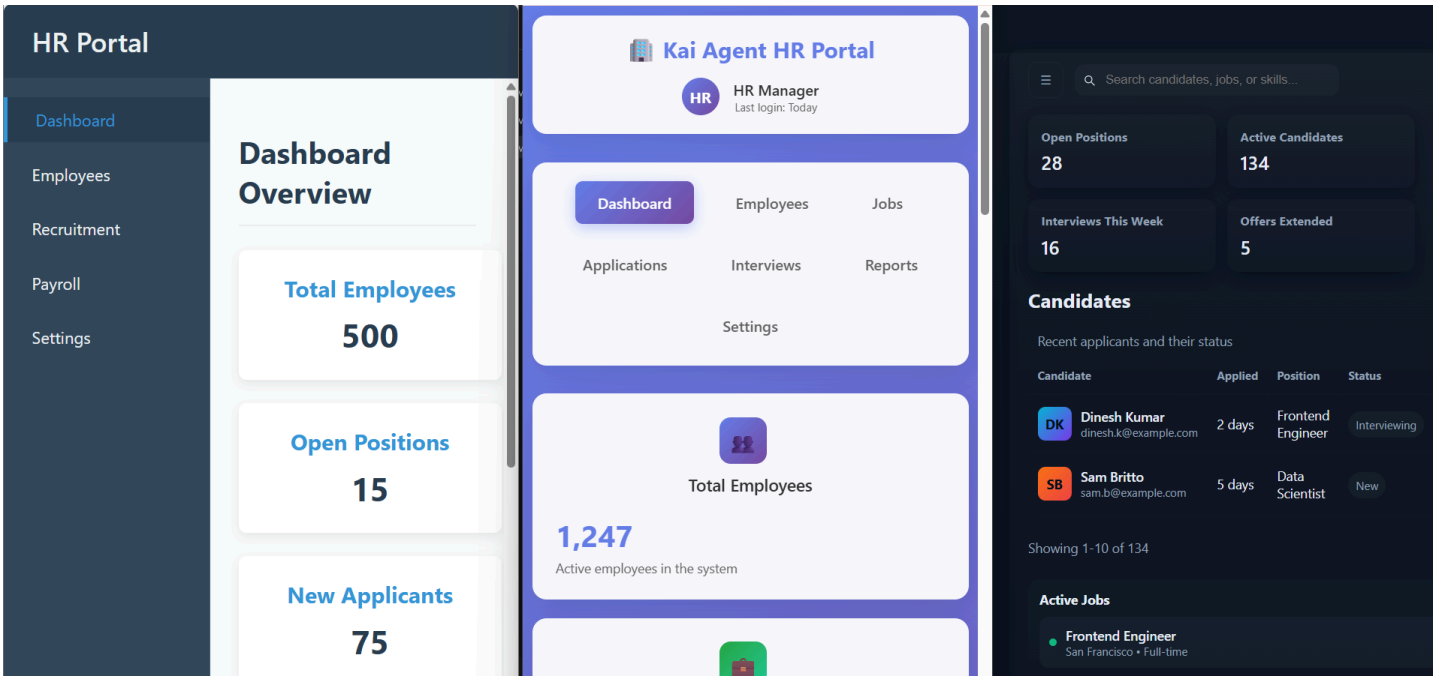
You can observe multiple agents running simultaneously on the same task



You can see that **Gemini 2.5 Flash** completed the task first, followed by **GPT-5 Mini**, while **Grok Code** is still running



The third output, generated by **GPT-5.2 Mini**, performs better when compared to the first output from **Gemini** and the second output from **Grok Code**



4. MCP Integration

4.1 Integrating Context7 MCP Server

Context7 integration in Cursor via MCP ensures the AI uses up-to-date, official documentation and real code examples, reducing hallucinations and improving accuracy.

Go to **Context7's official GitHub repository**, scroll down, and you will find the **installation instructions for Cursor**

> ⓘ Note
>
> **API Key Recommended**: Get a free API key at context7.com/dashboard for higher rate limits.

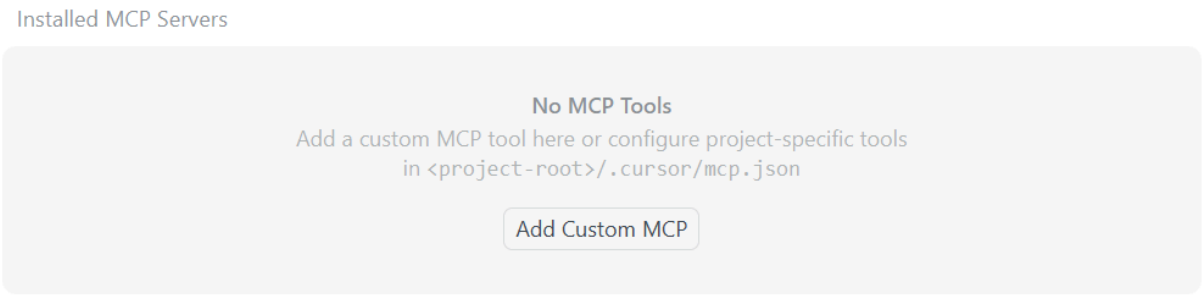
▼ **Install in Cursor**

Go to: `Settings` -> `Cursor Settings` -> `MCP` -> `Add new global MCP server`

Pasting the following configuration into your Cursor `~/.cursor/mcp.json` file is the recommended approach. You may also install in a specific project by creating `.cursor/mcp.json` in your project folder. See Cursor MCP docs for more info.

> Since Cursor 1.0, you can click the install button below for instant one-click installation.

Go to **Cursor Settings → Tools & MCP → Add Custom MCP** and paste the code



Cursor Local Server Connection:

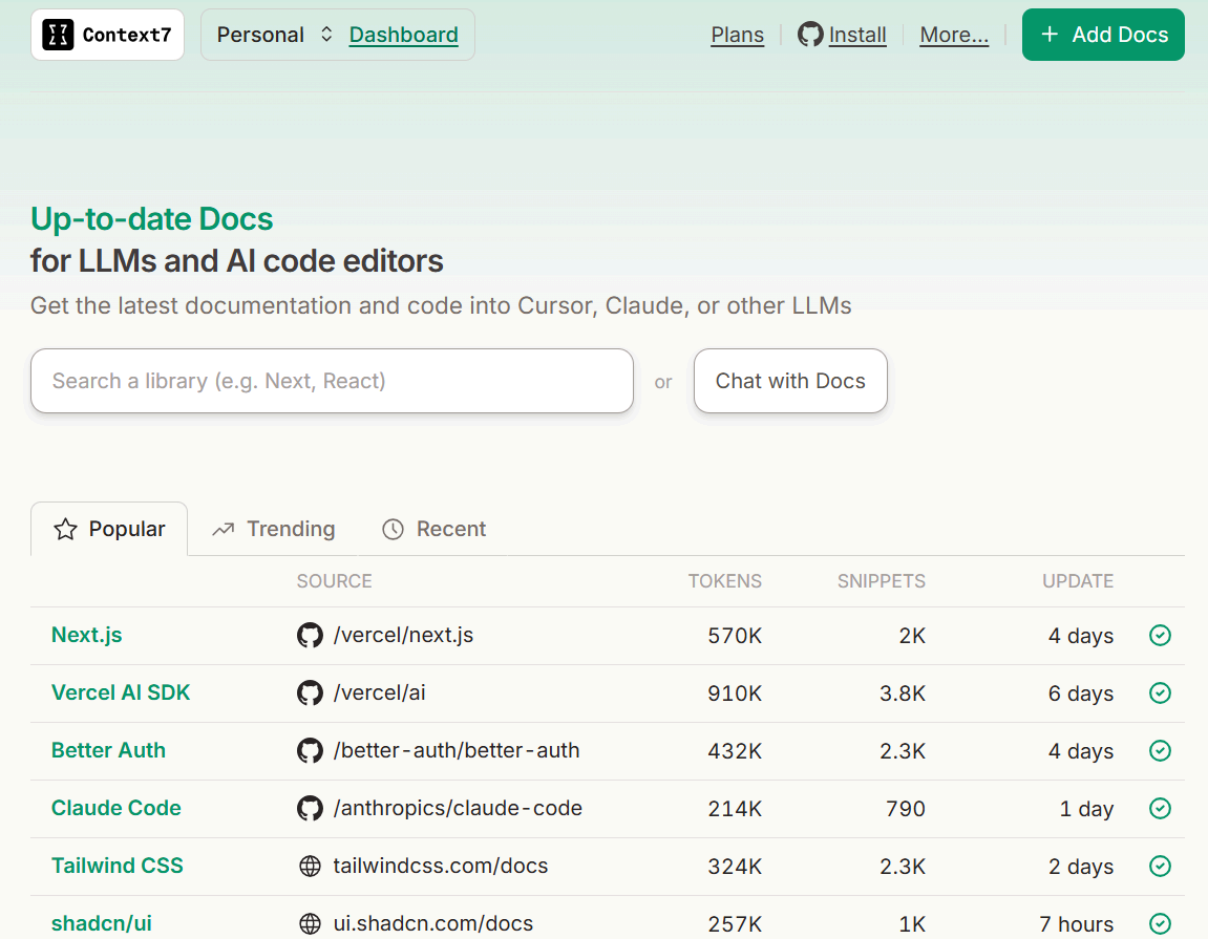
```
{
  "mcpServers": {
    "context7": {
      "command": "npx",
      "args": ["-y", "@upstash/context7-mcp", "--api-key",
"YOUR_API_KEY"]
    }
  }
}
```

Cursor Remote Server Connection:

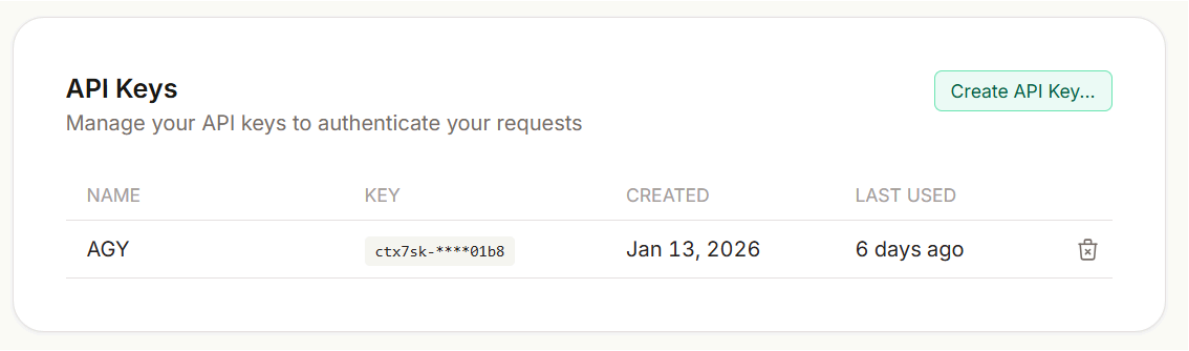
```
{
  "mcpServers": {
    "context7": {
      "url": "https://mcp.context7.com/mcp",
      "headers": {
        "CONTEXT7_API_KEY": "YOUR_API_KEY"
      }
    }
  }
}
```

Replace the API key placeholder with your actual API key, which can be obtained from the Context website's dashboard

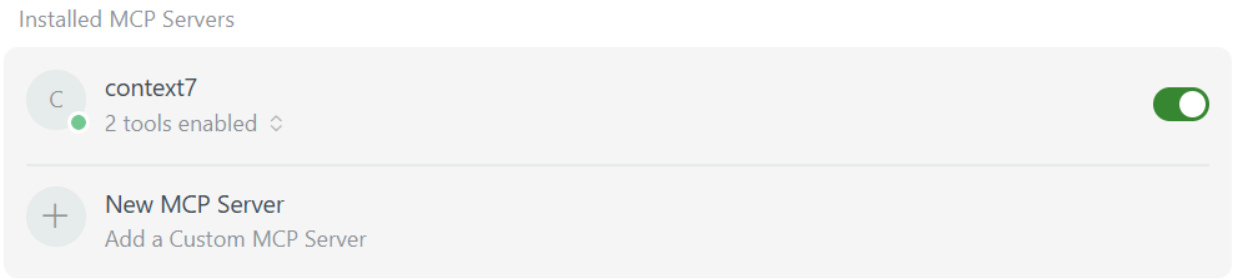
To obtain the API key, go to the **Context7 website** and click on **Dashboard**



Click **Create API Key**, enter a name, and the API key will be generated.



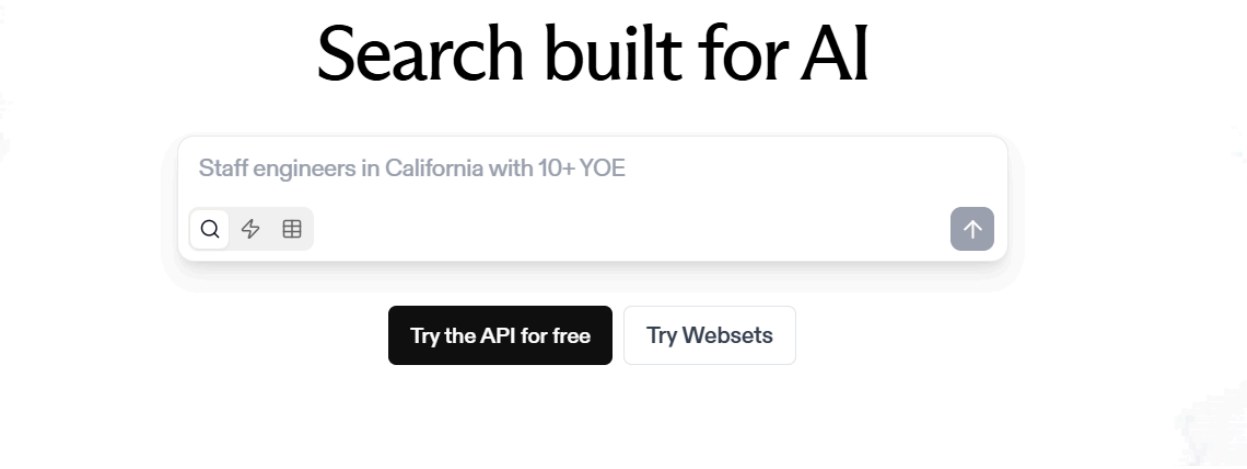
Now the **Context7 MCP Server** is ready to be used



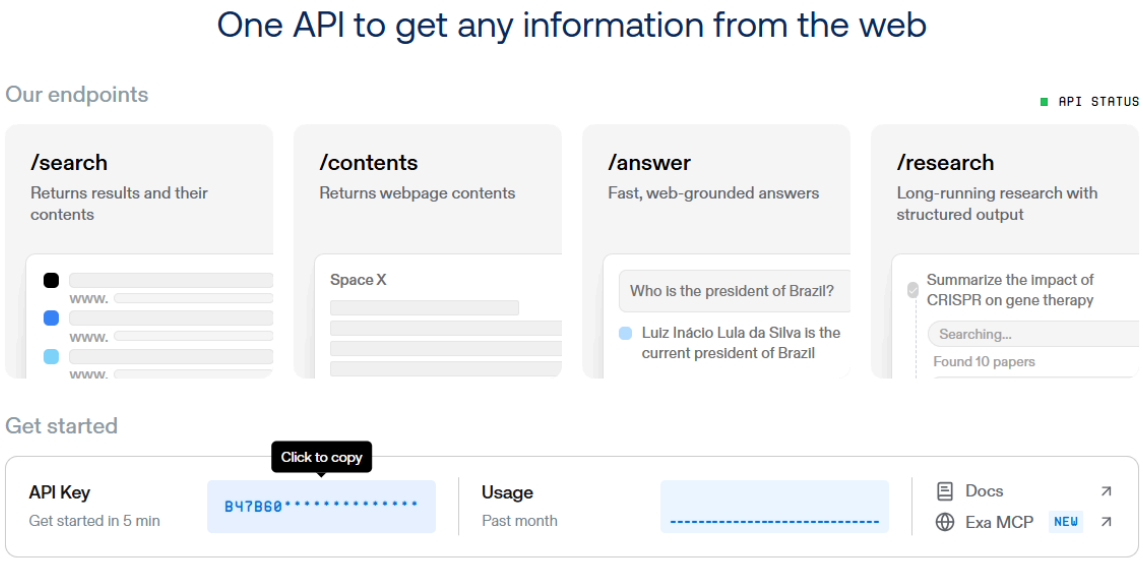
4.2 Integrating Exa AI MCP Server

Exa AI in Cursor enables AI agents to fetch precise, real-time web knowledge (code, docs, GitHub) using semantic search, reducing hallucinations and powering research-aware development.

Go to **Exa AI's official website**, log in, and navigate to the **dashboard**



From the dashboard, generate and copy the **API key**, then click **Exa MCP** to obtain Cursor's **Exa AI MCP configuration code**

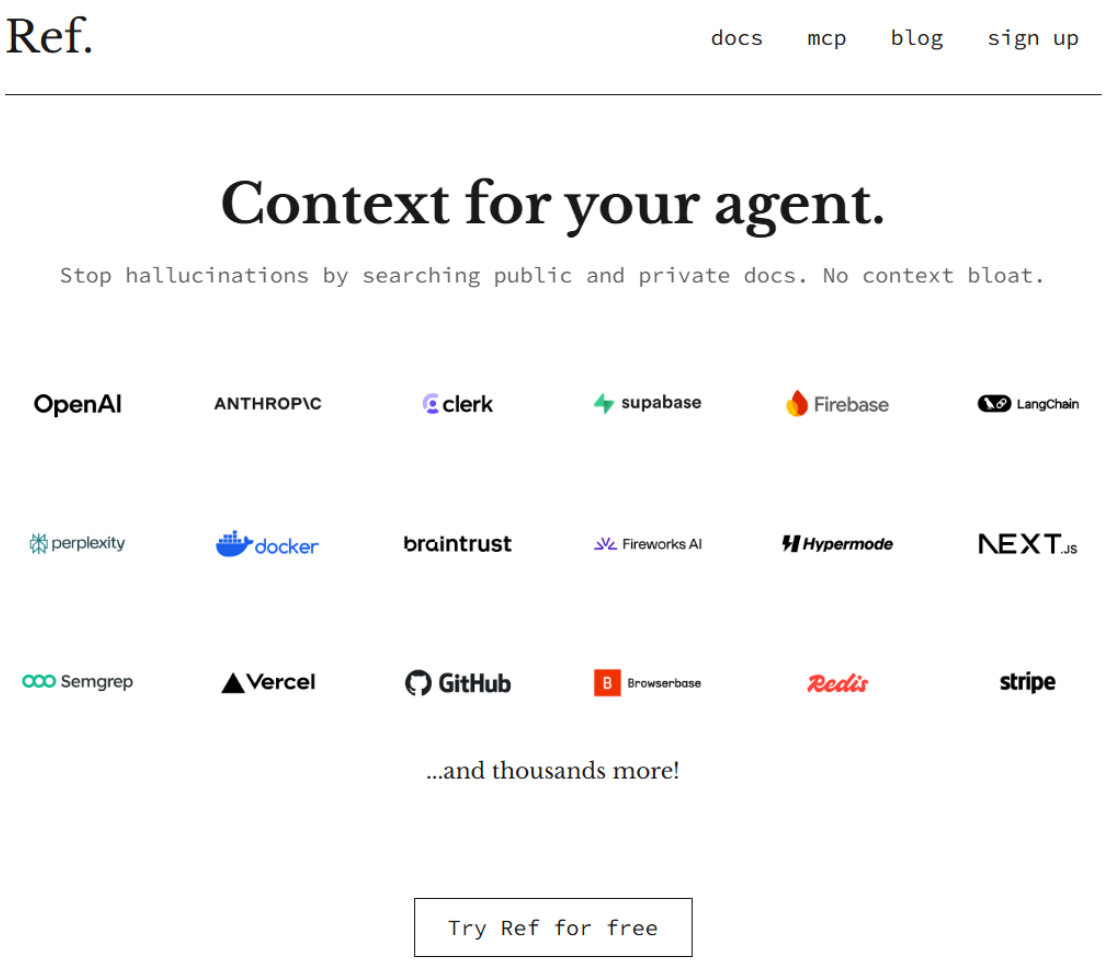


Click **Install in Cursor** to automatically open Cursor and add the MCP server

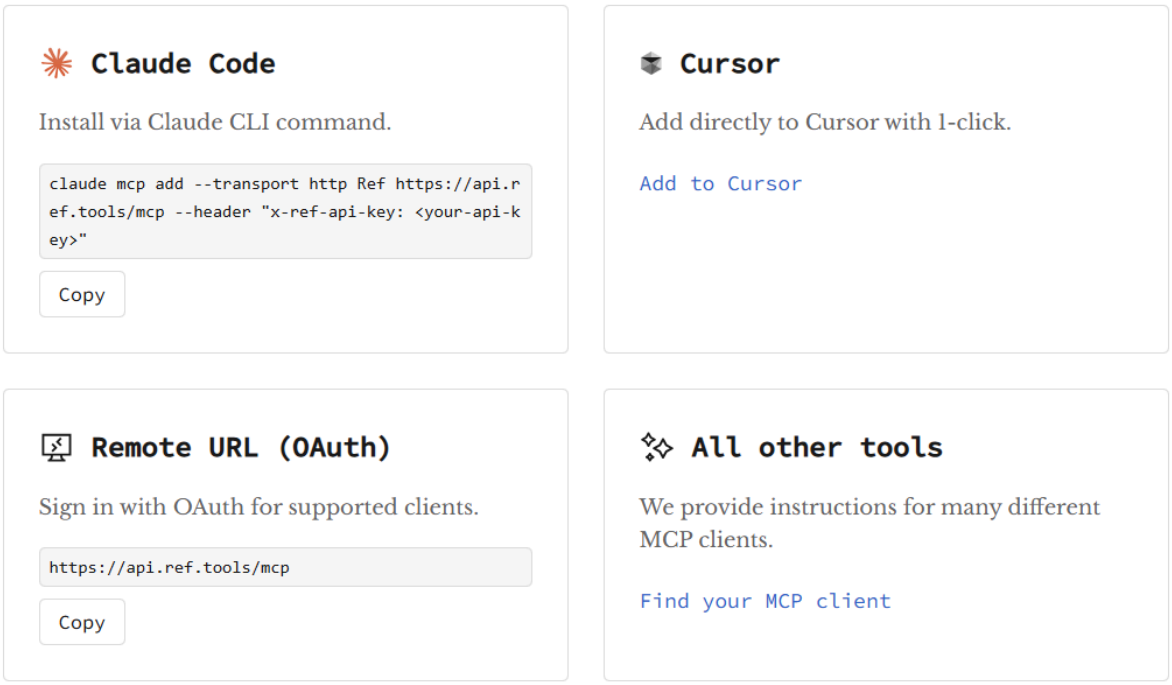
4.3 Integrating Ref MCP Server

Ref in Cursor enables efficient, token-optimized retrieval of relevant information from PDFs and GitHub repositories (including private sources) to provide accurate, grounded context for AI workflows.

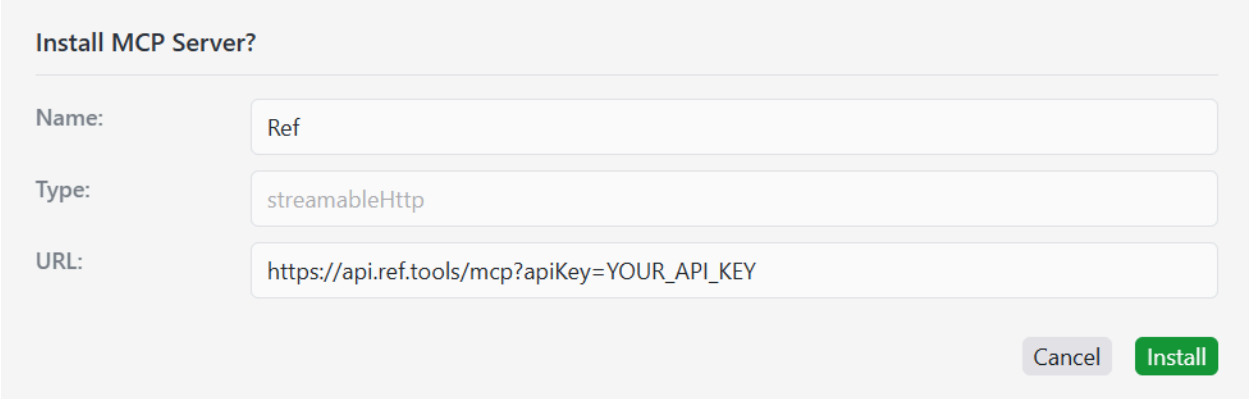
Go to https://ref.tools/ and and click **MCP** in the top-right corner



Click **Add to Cursor** to automatically add the MCP configuration code to Cursor



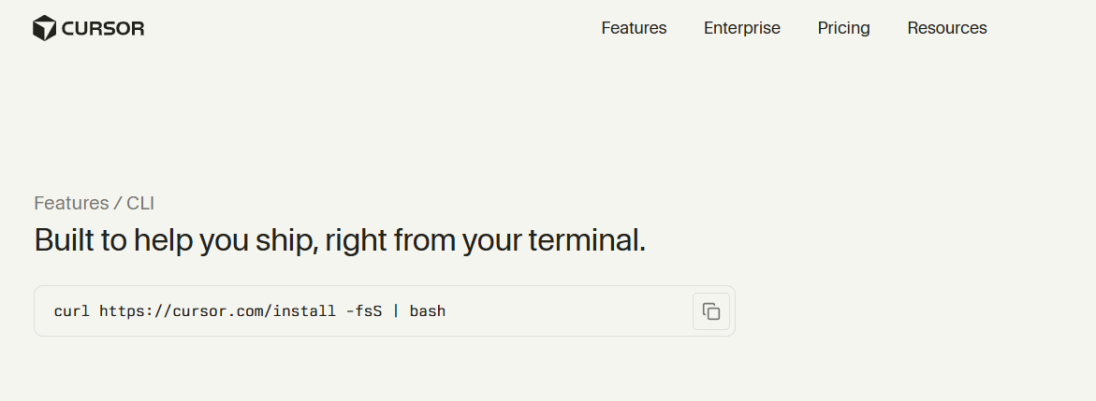
Go to the **Ref dashboard**, log in to generate the **API key**, and replace the placeholder with the actual key



5. Cursor CLI

Cursor CLI is a terminal-based AI assistant that lets developers write, review, debug, and test code using natural language directly from the command line.

Go to https://cursor.com/cli and copy the provided **curl** command



AWS CloudShell is used as the terminal environment

```
curl https://cursor.com/install -fsS | bash
```

```
Cursor Agent Installer

✓ Detected linux/x64
✓ Package downloaded and extracted
✓ Package installed successfully
✓ Bin directory ready
✓ Symlink created

✨ Installation Complete!


Start using Cursor Agent:
   agent


Happy coding! 🚀
```

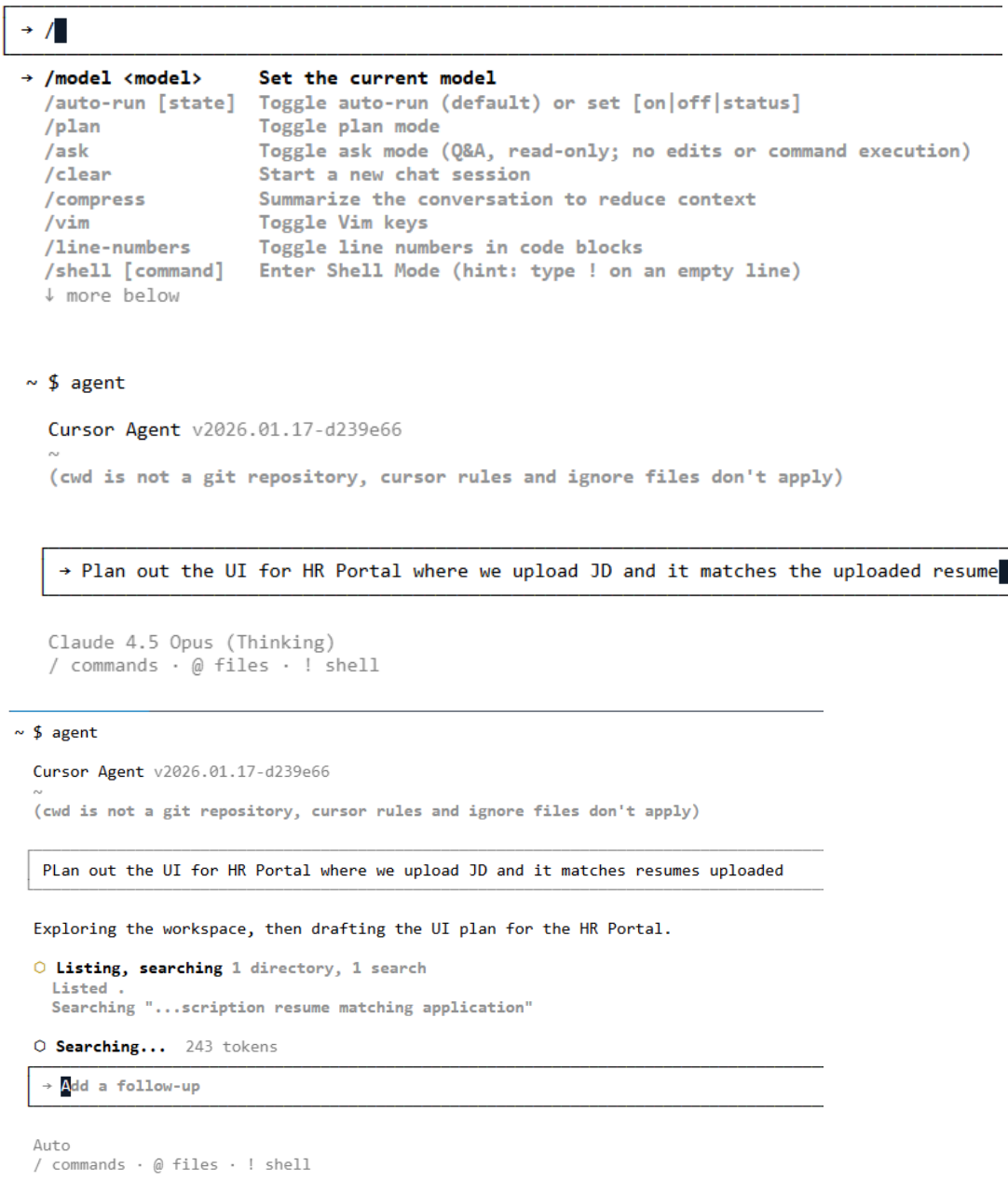
The installation is complete, and the Cursor agent is ready to use

```
~ $ agent
```


```
~ $ agent
 Signing in
 If your browser didn't open, click this link to log in:

https://cursor.com/loginDeepControl?challenge=G31qFQo2j5FePhaCMiaNUKfL_vVx1EM9ypkOpNWK_c8&uuid=f02cd9b
```

```
 Cursor Agent v2026.01.17-d239e66
 ~
 (cwd is not a git repository, cursor rules and ignore files don't apply)

 → Plan, search, build anything

 Claude 4.5 Opus (Thinking)
 / commands · @ files · ! shell
```

Type **/** to view all available commands



```
Creating a UI plan document for the HR Portal covering structure, pages, components, and user flows.

HR_PORTAL_UI_PLAN.md +843

+ # HR Portal UI Plan
+ ## Job Description & Resume Matching System

+ ---
+
+ ## 1. Overview
+
+ ### Purpose
+ An enterprise HR Portal that enables HR teams to:
+ - Upload and manage Job Descriptions (JD)
+ - Upload and manage candidate resumes
+ - Automatically match resumes to job descriptions
  … truncated (832 more lines) · ctrl+r to review

Created a UI plan for the HR Portal. Summary:

Plan highlights

Core sections

1. Dashboard — Overview with stats, quick actions, and activity feed
2. Job Descriptions — CRUD for job postings with matching criteria
3. Resumes — Library management and bulk/single upload
4. Matches — Results, detailed analysis, and comparison views
5. Candidates — Profile management and application tracking
6. Settings — Profile and matching algorithm configuration
```

It also prompts for follow-up actions or questions

```
Should I:
1. Create wireframe mockups for specific pages?
2. Generate a component library specification?
3. Create a technical architecture document?
4. Build a prototype of any specific section?


→ Add a follow-up
```

Cursor CLI supports planning, writing, and running code—just like the IDE

6. Cloud Agent

- Cloud agents in Cursor run AI coding tasks remotely, allowing parallel execution without relying on the local machine.
- Cloud agents in Cursor can be monitored and managed seamlessly from both web and mobile devices

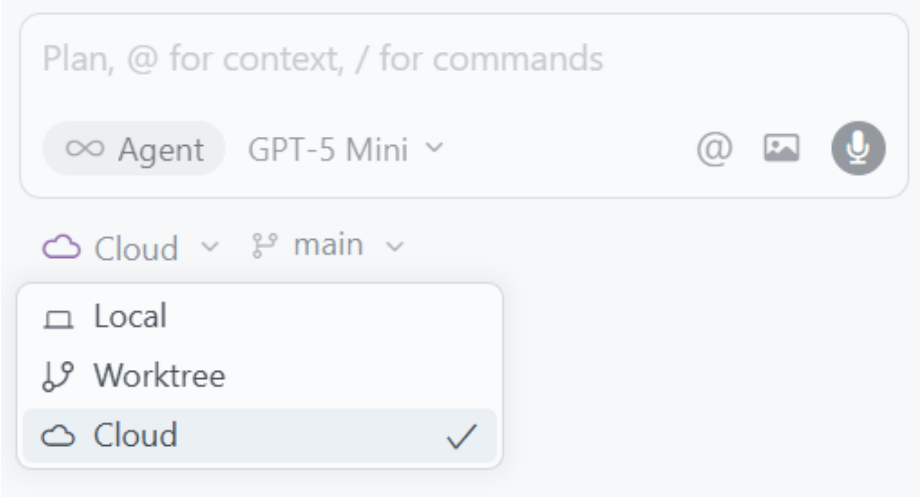
You can access cloud agents in two ways:

- Select **Cloud** from the dropdown under the agent input to run agents from the editor.
- Start and manage cloud agents directly from **cursor.com/agents**

6.1 Accessing Cloud Agents from the Editor

To run **cloud agents**, switch the agent execution mode from **Local** to **Cloud**, which runs agents in an isolated remote environment without affecting the local workspace.

In the agent window, select **Cloud** as the execution mode to start using cloud agents



6.1 Accessing Cloud Agents from the Web

Go to https://cursor.com/agents