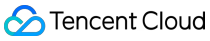
LLM Service TokenHub

AI Tools Integration

Product Documentation

Copyright Notice

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by the Tencent corporate group, including its parent, subsidiaries and affiliated companies, as the case may be. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

Contents

AI Tools Integration
CodeBuddy Code

Last updated: 2026-06-11 17:53:15

CodeBuddy Code is an intelligent programming tool built on Tencent Cloud AI technology. It deeply integrates with the Tencent Cloud ecosystem and provides end-to-end AI assistance, covering everything from code writing to project deployment. This article demonstrates how to integrate a model for use with CodeBuddy Code, using DeepSeek V4 Pro as an example.

Installing CodeBuddy Code

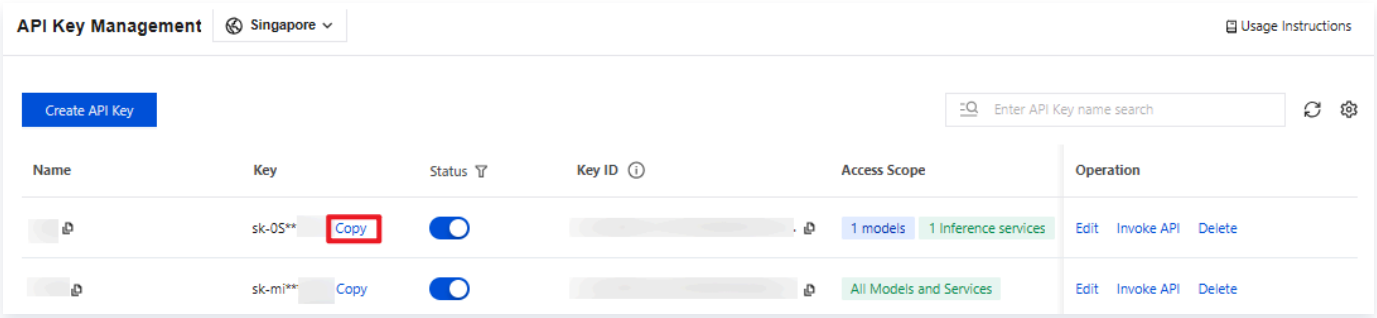
Install CodeBuddy Code according to your actual environment type. For details, see CodeBuddy Code Installation. If CodeBuddy Code is already installed, you can skip this step.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



Configuring CodeBuddy Code

1. Locate the configuration file for CodeBuddy Code. The configuration file path is provided below. If the configuration file does not exist, create it.
   - Windows: `C:\Users\<username>\.codebuddy\models.json`.
   - macOS/Linux: `~/.codebuddy/models.json`.
2. Copy and paste the following configuration, then save it.

Note:

- Replace `<USER_API_KEY>` in your code with the API Key string you obtained from the API Key Management page. This key is used for authentication. **Keep it secure and do not expose it or commit it to a code repository.**
- `models.json` is the model configuration file for CodeBuddy Code. If it already exists, you only need to append the `deepseek-v4-pro` configuration to `models` and `availableModels`. For more related configurations, see CodeBuddy Code Model Configuration.

```
{
  "models": [
    {
      "id": "deepseek-v4-pro",
      "name": "deepseek-v4-pro",
      "vendor": "Tencent Cloud",
      "apiKey": "<USER_API_KEY>",
      "url": "https://tokenhub-intl.tencentcloudmaas.com/v1"
    }
  ],
  "availableModels": ["deepseek-v4-pro"]
}
```

Using Tikit

After configuration is complete, run the following command to start CodeBuddy Code.

```
codebuddy
```

When you use CodeBuddy Code for the first time, the **Trust File Selection** page appears first, asking whether to trust the project files in the current working directory. Follow the recommendations below to make your selection:

- **Yes, proceed** (Recommended): Select this option only if you trust all files in the current directory. This allows CodeBuddy Code to read and write project files and run commands to assist with development.
- **No, exit**: Select this option when the current directory contains code you do not trust or whose source is unknown. The tool will then exit to ensure security.

Complete the login authentication. After startup, the login method selection page is displayed:

```
Select login method:
› Log in via Chinese Site
  Log in via International Site
  Log in via Enterprise Domain
  Log in via iOA (Tencent only)
```

The login methods are described as follows:

Login Method	Use Cases	Description
Chinese Site	Users of the Chinese Site	Authenticate via Tencent Cloud China Site (copilot.tencent.com), supporting mainstream models within China.
International Site	International Site users	Authenticate via Tencent Cloud International Site (codebuddy.ai), supporting mainstream models outside China.
Enterprise Domain	Dedicated Edition/Private Deployment	Connect to the enterprise-exclusive or self-built CodeBuddy service, and enter the service address provided by your enterprise.
iOA	Tencent internal employees	Authenticate via Tencent iOA Zero Trust system, and is restricted to Tencent internal employees only.

Use the ↑ ↓ keys to select a login method. After you press Enter to confirm, a browser will automatically open to complete the authentication.

After successful authentication, you are returned to the **CodeBuddy Code command-line interface**. In the input box, directly type `/model` and press Enter to go to the model selection list. Select the `deepseek-v4-pro` model configured in this document to start using it.

For more information about using CodeBuddy Code, see the CodeBuddy Code documentation.

WorkBuddy

Last updated：2026-06-11 17:53:15

WorkBuddy is a full-scenario desktop AI agent launched by Tencent Cloud. It integrates with the QQ / WeCom ecosystems, supports local computer operations and multi-model switching, and provides commercial services through a Token quota package model, catering to both individual and team usage. This document uses `deepseek-v4-pro` as an example to explain how to configure and use the Tencent Cloud TokenHub large model within WorkBuddy.

Installing WorkBuddy

Visit the WorkBuddy official website to download and install WorkBuddy:

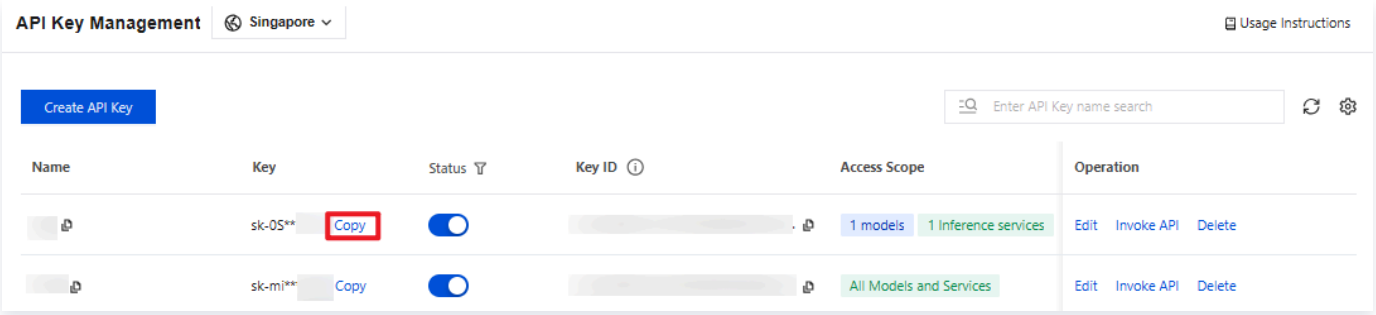
- If you are using a Mac system, see the Mac System Installation Guide to install it.
- If you are using a Windows system, see the Windows System Installation Guide to install it.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

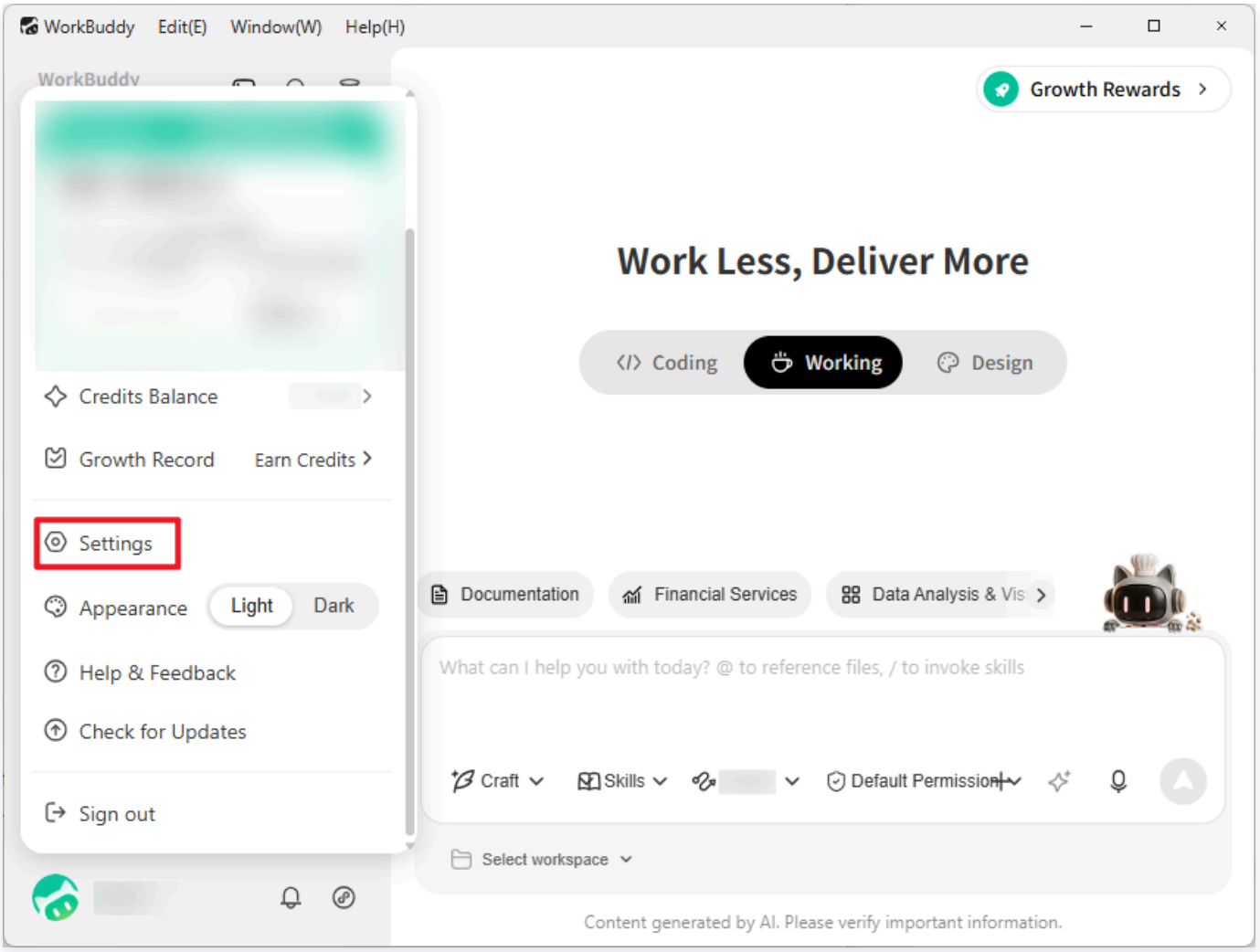
   > **Note:**
   > Confirm the Key's **Access Scope**. If it is "Range", ensure that `deepseek-v4-pro` is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.

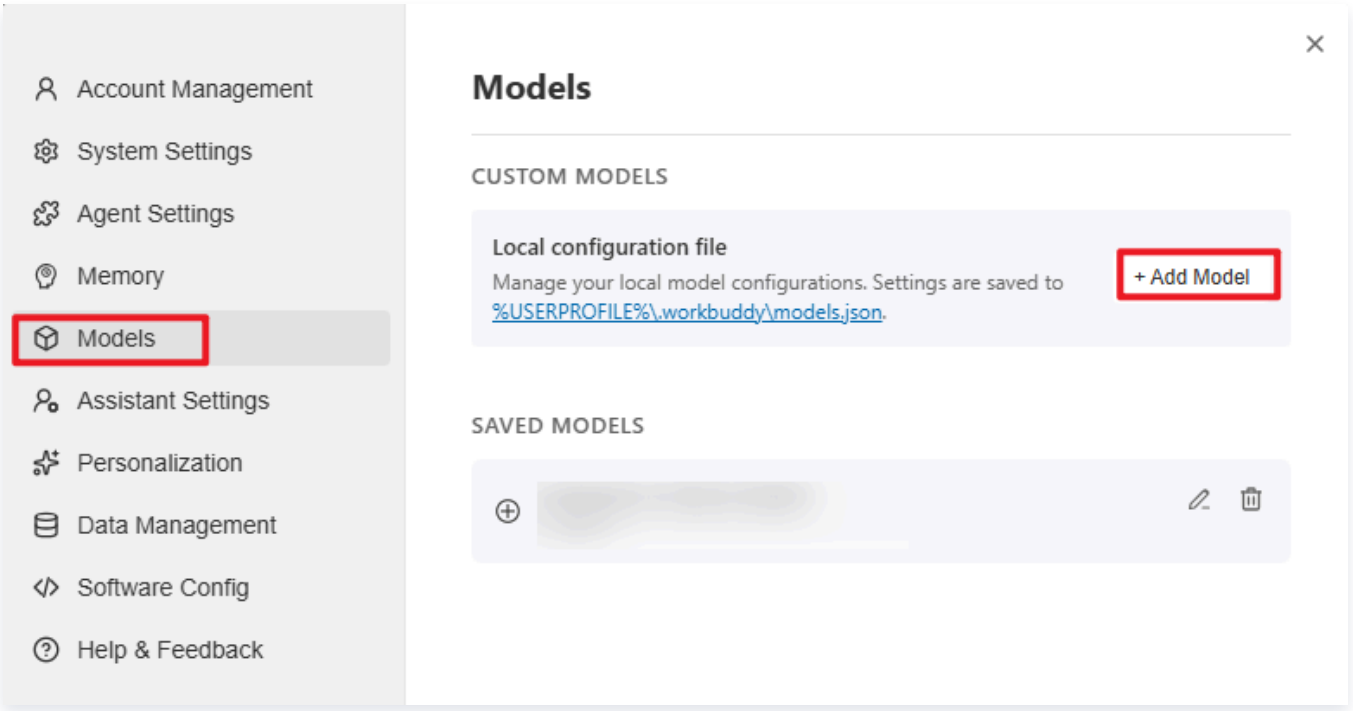


Configuring WorkBuddy

1. Start WorkBuddy, click the account icon in the lower-left corner, and select **Settings**.

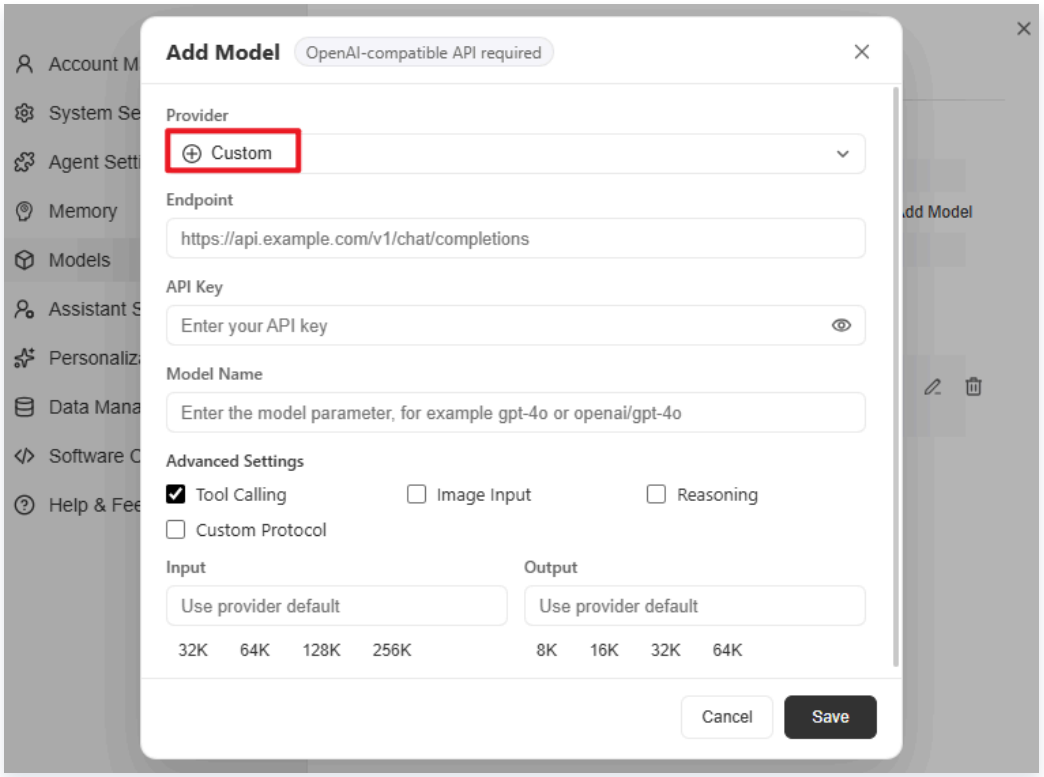


2. In the left navigation, select **Models**. Then, in the Custom Models section, click **Add Model**.



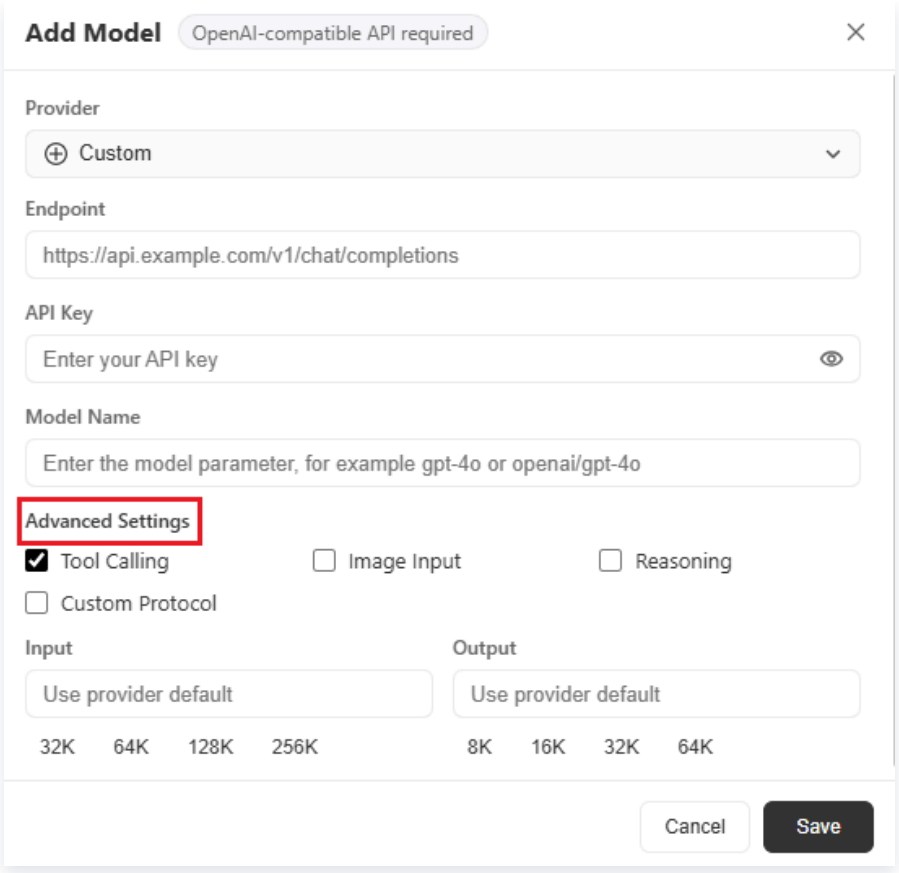
3. Configure the following information and click **Save**:

3.1 In the provider section, select **Custom**. The following figure illustrates this:

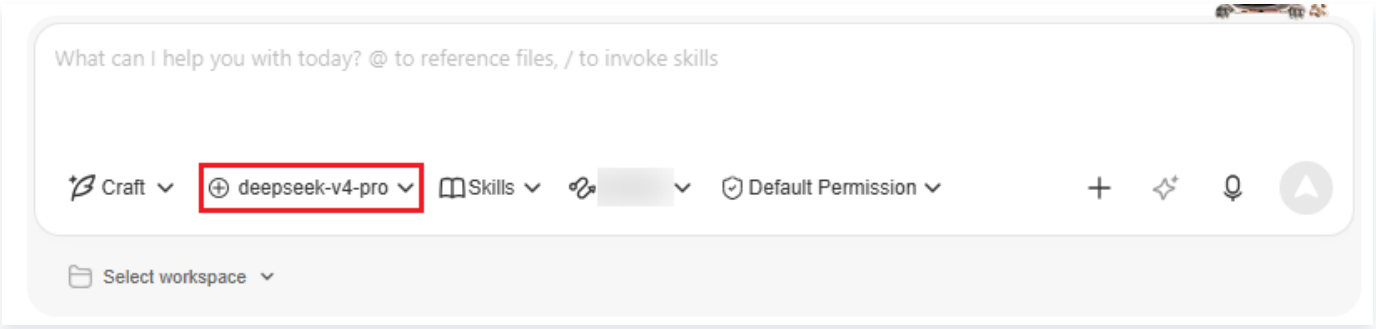


3.2 Fill in the following field information:

- **Endpoint**: https://tokenhub-intl.tencentcloudmaas.com/v1/chat/completions
- **API Key**: Enter your API Key.
- **Model Name**: Enter `deepseek-v4-pro`.
- **Advanced** Settings: This is optional. Use it as needed. It is recommended to select: Tool Calling, Image Input, and Reasoning.



4. After configuration is complete, select the added model from the WorkBuddy model selection box to start a conversation.



Claude Code

Last updated：2026-06-11 17:53:15

Claude Code is an AI-powered programming tool that runs in the terminal. It assists developers in efficiently coding, debugging, and managing code through flexible natural language. This article demonstrates how to integrate a model, using DeepSeek V4 Pro as an example, for use with Claude Code.

Installing Claude Code

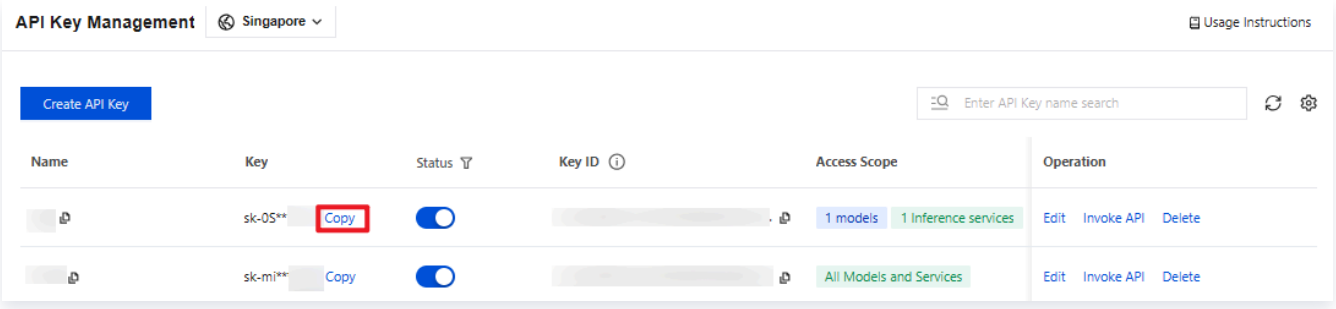
Install Claude Code according to your actual environment type. For details, see Claude Code Installation. If Claude Code is already installed, you can skip this step.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



Configuring Claude Code

1. Manually Configuring Model Environment Variables

After completing the installation of Claude Code and obtaining the API Key, you need to configure the API information for the DeepSeek V4 Pro model. The specific configuration steps are as follows:
Edit or create the `settings.json` file. Then, based on the channel through which you obtained the API Key, add the corresponding configuration content. Note that the configuration file path varies across different systems. For details, refer to:

- For MacOS/Linux, the corresponding path is `~/.claude/settings.json`.
- For Windows, the corresponding path is `%USERPROFILE%/.claude/settings.json`.

If the `settings.json` file does not exist, you can manually create it using a text editor (such as VSCode or Notepad) or run the following command in the terminal to create it:

```
# MacOS/Linux
mkdir -p ~/.claude && touch ~/.claude/settings.json

# Windows(PowerShell)
New-Item -ItemType Directory -Force -Path "$HOME\.claude" | Out-Null;
New-Item -ItemType File -Force -Path "$HOME\.claude\settings.json" |
Out-Null
```

Configure the following content (note: replace `YOUR_API_KEY` with your actual API Key):

```
{
  "env": {
    "ANTHROPIC_BASE_URL": "https://tokenhub-intl.tencentcloudmaas.com",
    "ANTHROPIC_AUTH_TOKEN": "YOUR_API_KEY",
    "ANTHROPIC_MODEL": "deepseek-v4-pro",
    "ANTHROPIC_DEFAULT_OPUS_MODEL": "deepseek-v4-pro",
    "ANTHROPIC_DEFAULT_SONNET_MODEL": "deepseek-v4-pro",
    "ANTHROPIC_DEFAULT_HAIKU_MODEL": "deepseek-v4-pro",
    "CLAUDE_CODE_SUBAGENT_MODEL": "deepseek-v4-pro",
    "ENABLE_TOOL_SEARCH": false
  }
}
```

The meanings of the environment variables are as follows:

Environment Variable	Required	Description
`ANTHROPIC_BASE_URL`	Yes	The API Gateway address used by Claude Code to make requests. When Claude Code is integrated with TokenHub, it is fixed as `https://tokenhub-intl.tencentcloudmaas.com`.
`ANTHROPIC_AUTH_TOKEN`	Yes	The API Key used for authentication. Replace it with the real API Key you created in the TokenHub console. Keep it secure to avoid leakage.

`ANTHROPIC_MODEL`	Yes	The model name called by default by Claude Code. This document uses `deepseek-v4-pro` as an example. You can also replace it with other models supported by TokenHub.
`ANTHROPIC_DEFAULT_OPUS_MODEL`	No	The model to which the Opus tier (for high-complexity tasks) in Claude Code is mapped. It is set to `deepseek-v4-pro` here to make all tiers point to the same model.
`ANTHROPIC_DEFAULT_SONNET_MODEL`	No	The model to which the Sonnet tier (for default daily tasks) in Claude Code is mapped.
`ANTHROPIC_DEFAULT_HAIKU_MODEL`	No	The model to which the Haiku tier (for lightweight/quick tasks) in Claude Code is mapped.
`CLAUDE_CODE_SUBAGENT_MODEL`	No	The model used by Claude Code when subtasks are created (subagents). It is recommended to keep it consistent with the main model to avoid compatibility issues caused by cross-model calls.
`ENABLE_TOOL_SEARCH`	No	Whether to enable the native web search tool built into Claude Code. Since Claude Code prohibits third-party models from using the built-in search, set it to `false` here.

After saving the configuration, **open a new terminal window** to run subsequent commands. This ensures that the environment variables are loaded and take effect.

2. Verifying Whether Environment Variables Are Effective

Go to the command-line interface and run the following command to start Claude Code:

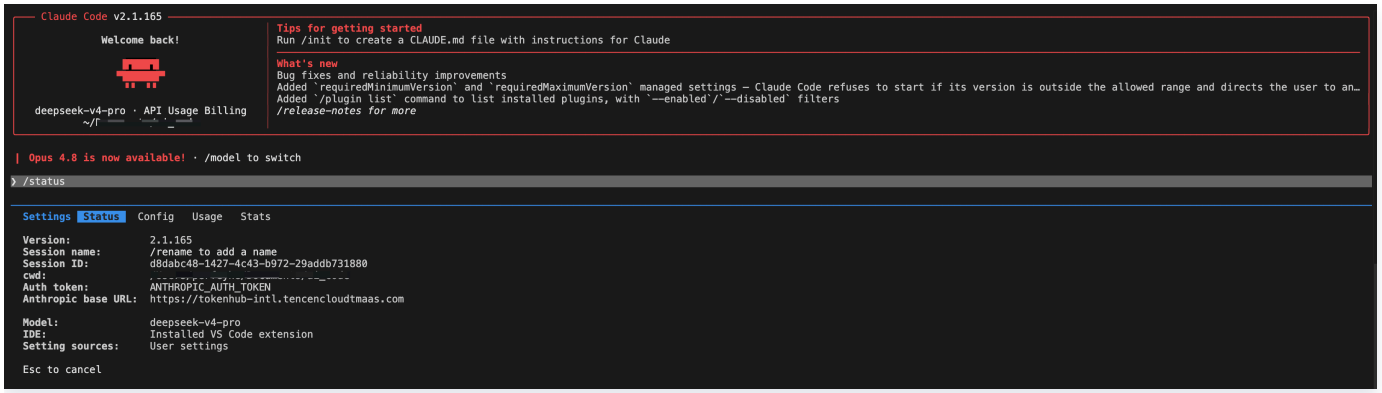
```
# Open the project directory.
cd your-project

# Start Claude Code.
claude
```

Note:

If this is your first time starting Claude Code, the terminal first enters an interactive initialization wizard. It then prompts you to select a Theme, a Login Method, and other options. Follow the

> prompts in the terminal to complete the initial configuration before proceeding with the subsequent steps.

After a successful startup, enter `/status` in Claude Code to confirm that the model status is active (refer to the example in the figure below). The model configuration is effective when `API Endpoint` displays `https://tokenhub-intl.tencentcloudmaas.com` and `Model` displays `deepseek-v4-pro`.



OpenClaw

Last updated：2026-06-11 17:53:15

OpenClaw is an open-source, local AI agent framework that enables AI to evolve from "answering questions" to "executing tasks hands-on." It supports autonomous operation on Windows, macOS, and Linux, automatically performing operations such as file organization, email processing, and code writing through built-in tools and an extensible plugin system. This article demonstrates how to integrate a model, using DeepSeek V4 Pro as an example, for use with OpenClaw.

Installation Methods

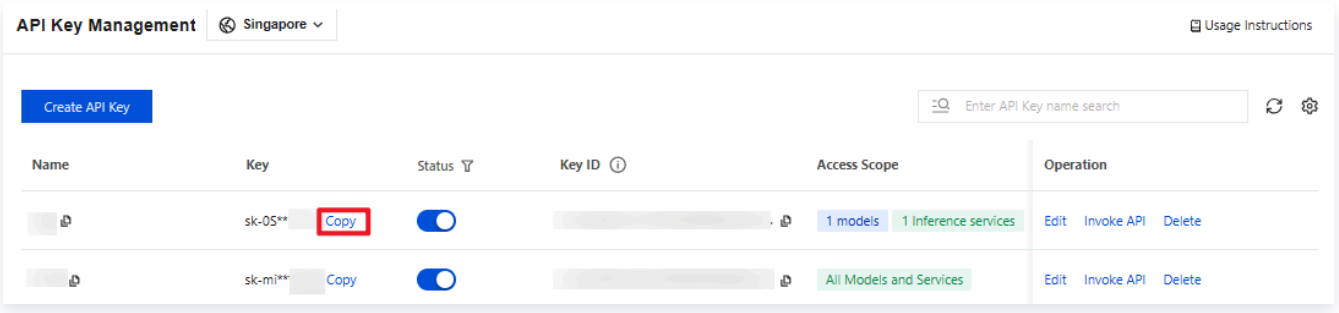
Install OpenClaw according to your actual environment type. For details, see OpenClaw Installation. If OpenClaw is already installed, you can skip this step.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



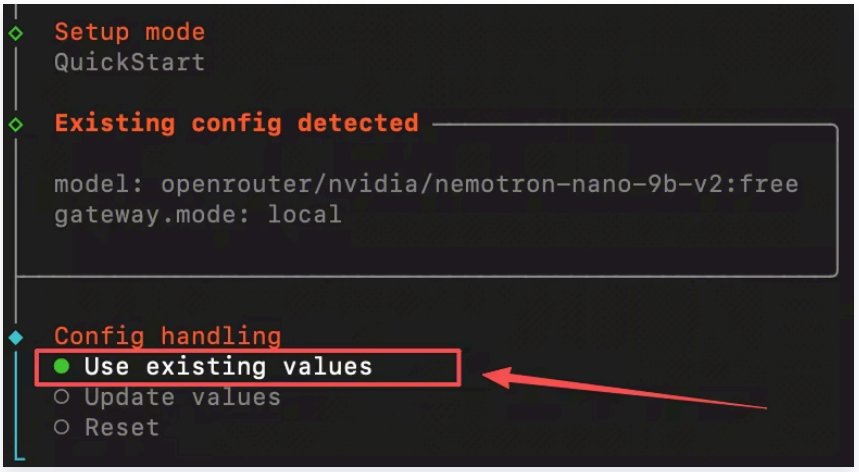
Configuring OpenClaw

Open a terminal and run the following command.

```
openclaw onboard --auth-choice tokenhub-api-key
```

Complete the relevant configurations sequentially based on the prompts displayed in the terminal. Refer to the following for the configuration details.

Basic configuration

Terminal Prompt Message	Configuration Details
I understand this is personal-by-default and shared/multi-user use requires lock-down. Continue?	Select `Yes`
Setup mode	Select `QuickStart` **Note:** If you have previously configured OpenClaw, selecting QuickStart may trigger the "Existing config detected" message as shown in the figure. The terminal will then output a Config handling options list. Simply select `Use existing values` to proceed. 

Model Configuration

Terminal Prompt Message	Configuration Details
Enter Tencent TokenHub API key	Enter the obtained API Key.
Default model	Set to tencent-tokenhub/deepseek-v4-pro.

Feature Configuration

Terminal Prompt Message	Configuration Details
Select channel (QuickStart)	Select `Skip for now` to configure later.

Configure skills now? (recommended)	Select `No` to configure later.
Enable hooks?	Press the Spacebar to select the option, and press the Enter key to go to the next step.
How do you want to hatch your bot?	Select `Hatch in TUI`.

Verifying Model Availability

After completing the configuration, you can run the following command in the terminal to verify whether the model is available:

```
openclaw models list --provider tencent-tokenhub
```

If the terminal outputs the information as shown in the following figure, the configuration is successful. You can then use the DeepSeek V4 Pro model in OpenClaw.

Hermes Agent

Last updated：2026-06-11 17:53:15

Hermes Agent is an autonomous AI agent framework open-sourced by Nous Research. You can deploy it on a local server and connect it to various large language models. It can remember across sessions, automatically create skills, and continuously self-evolve to complete complex automated tasks. This article demonstrates how to integrate a model, using DeepSeek V4 Pro as an example, for use with Hermes Agent.

Installing Hermes Agent

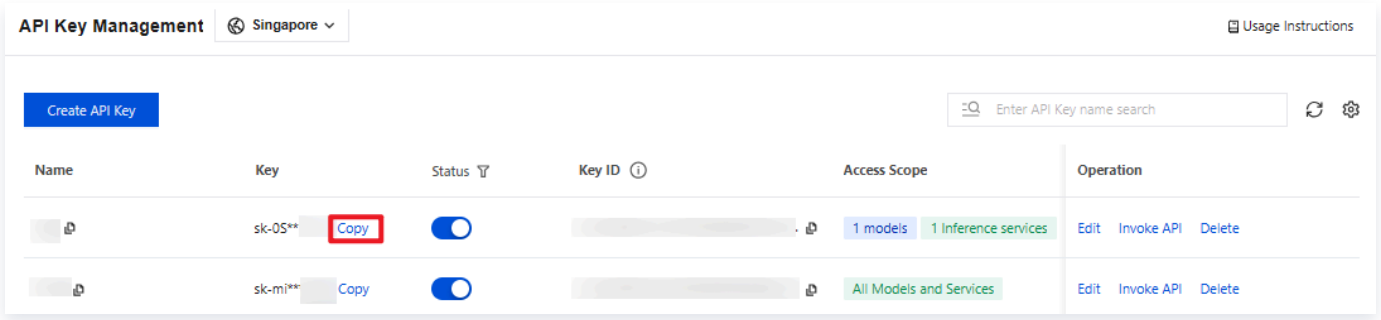
Install Hermes Agent according to your actual environment type. For details, see Hermes Agent Installation. If Hermes Agent is already installed, you can skip this step.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



Configuring Hermes Agent

> **Note:**
> Configuration steps may be adjusted depending on the Agent version. Please refer to the version you are using.

1. Open a terminal and run the following command:

```
hermes setup
```

2. If this is your first launch after installation, the terminal will display the following message: select `Quick setup — provider, model & messaging (recommended)`.

```
How would you like to set up Hermes?
  ↑↓ navigate  ENTER/SPACE select  ESC cancel

→ (●) Quick setup — provider, model & messaging (recommended)
  (○) Full setup — configure everything
```

> **Note:**
> If you have previously configured Hermes Agent, after you run `hermes setup`, the terminal output may appear as shown in the following figure. In this case, select Model & Provider to proceed to model configuration.
>
> ```
> What would you like to do?
>   ↑↓ navigate  ENTER/SPACE select  ESC cancel
>
>   (●) Quick Setup - configure missing items only
>   (○) Full Setup - reconfigure everything
> → (○) Model & Provider
>   (○) Terminal Backend
>   (○) Messaging Platforms (Gateway)
>   (○) Tools
>   (○) Agent Settings
>   (○) Exit
> ```

3. After you make your selection, the terminal will display the available model providers. Scroll to the bottom, select `Custom endpoint`, and press Enter to confirm.

```
→ (○) Custom endpoint (enter URL manually)
  (○) Remove a saved custom provider
  (○) Cancel
```

4. Configure model information: Fill in the relevant configurations step by step according to the terminal prompts.

Terminal Prompt Message	Configuration Details
API base URL [e.g. https://api.example.com/v1]:	https://tokenhub-intl.tencentcloudmaas.com/v1
API key [optional]:	Enter the corresponding API Key.

API compatibility mode	Select Chat Completions.
Model name (e.g. gpt-4, llama-3-70b):	Enter `deepseek-v4-pro`
Context length in tokens [leave blank for auto-detect]:	Enter `256000`
Display name	Enter `deepseek-v4-pro`
Terminal backend	Fill in based on the actual situation. Select Local if it is installed locally.
Gateway working directory	Configure based on actual requirements. The default value can be used.
Enable sudo support	Whether to allow sudo. Fill in based on your requirements. Selecting sudo requires entering a password.

Using Hermes Agent

After completing the configuration, execute the `hermes` command in the terminal to start Hermes and begin a conversation using DeepSeek V4 Pro.

OpenCode

Last updated：2026-06-11 17:53:15

OpenCode is an open-source AI programming agent. It can integrate with various large models for use in terminals, desktop clients, or IDE extensions, assisting developers with tasks such as code analysis and intelligent coding. This article demonstrates how to integrate a model for use with OpenCode, using DeepSeek V4 Pro as an example.

Installing OpenCode

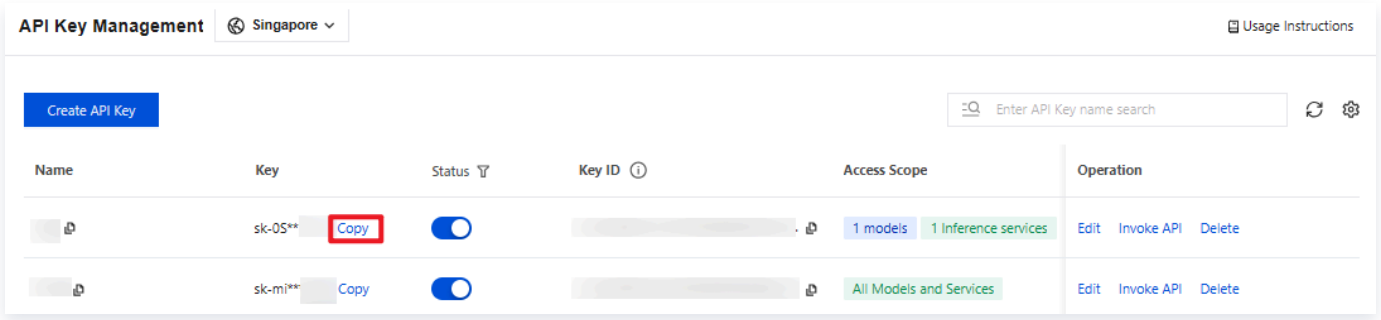
Install OpenCode. For details, see OpenCode Installation Guide. You can choose to download the desktop application Beta version. If OpenCode is already installed, you can skip this step.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

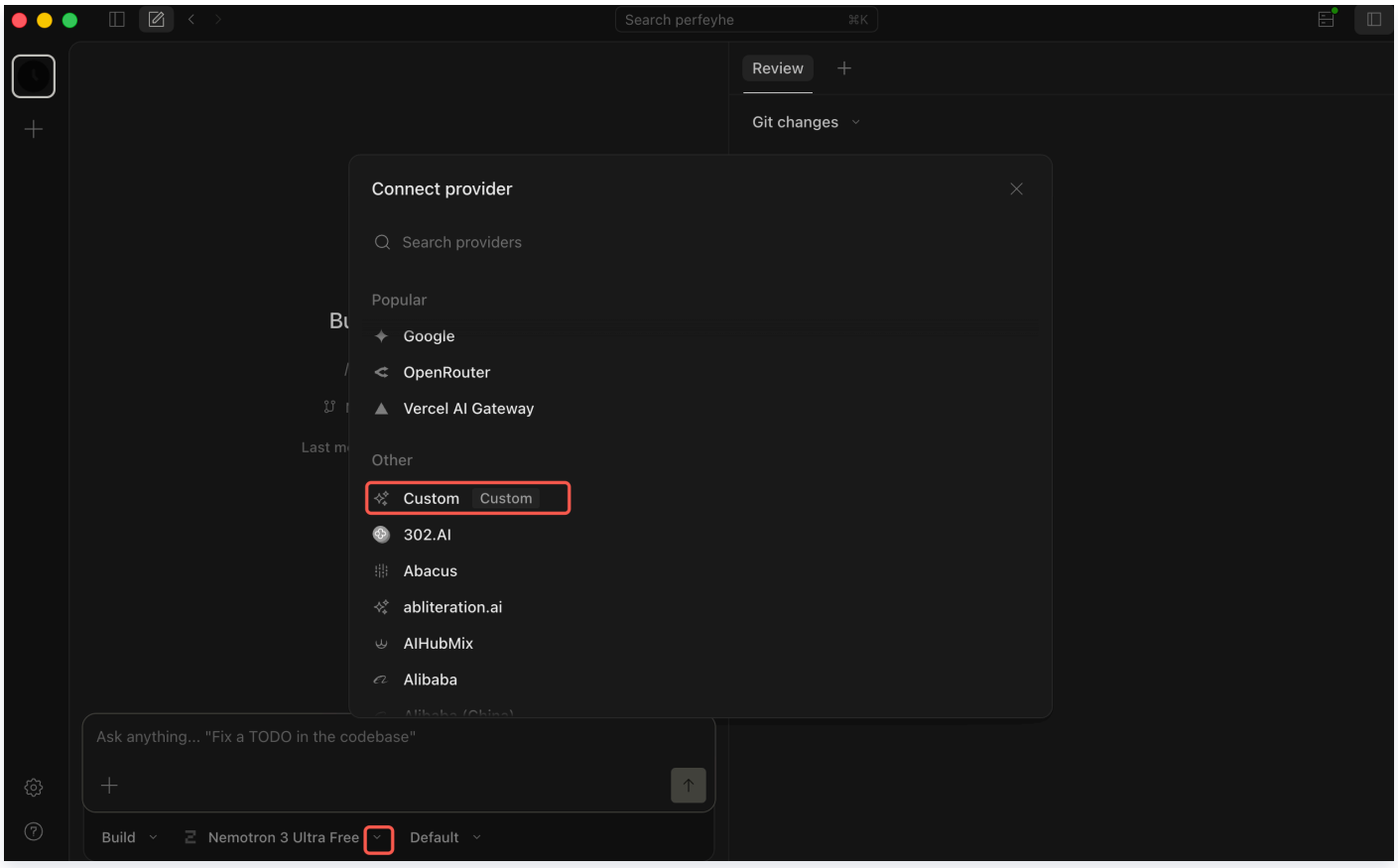
   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



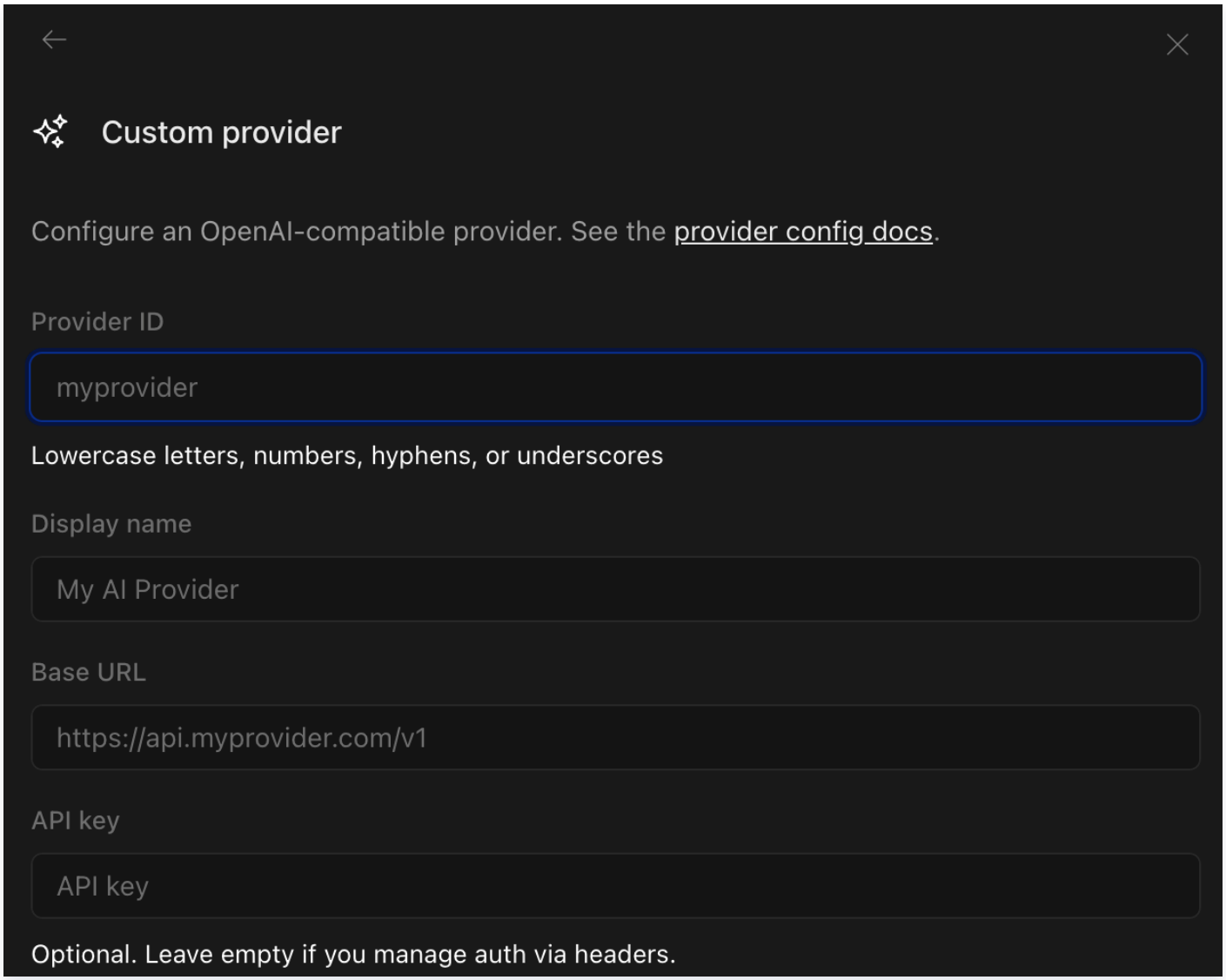
Configuring OpenCode

Open OpenCode. In the conversation interface, expand the model list and click the "+" in the upper-right corner to open the model settings page. In the pop-up window that opens, click "Show more providers" and select "Custom".

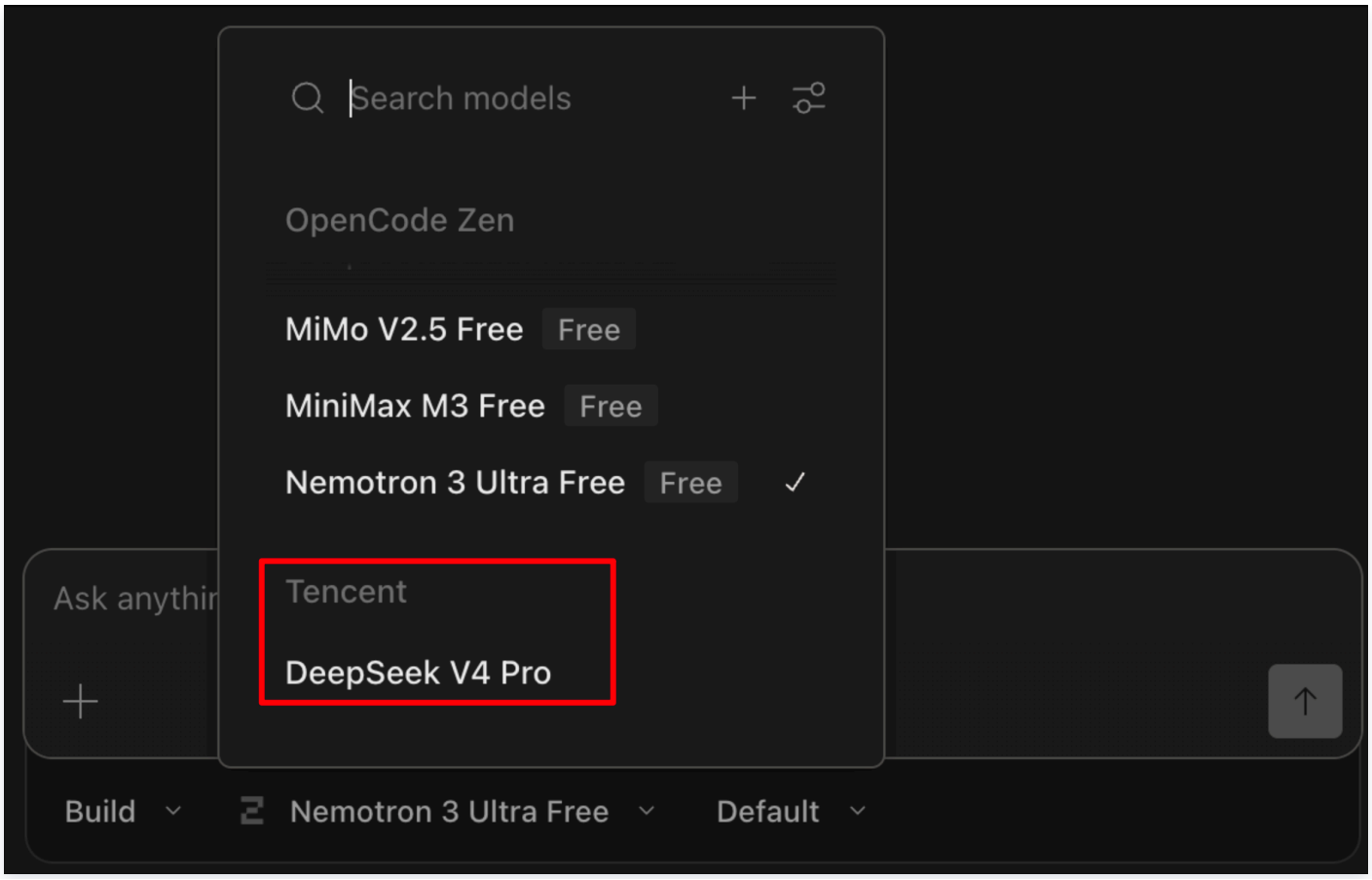


In the pop-up window that opens, click to fill in the information:

- **Provider ID**: `tencent`
- **Display Name**: `Tencent`
- **Base URL**: `https://tokenhub-intl.tencentcloudmaas.com/v1`
- **API Key**: Enter your API Key.
- **Model ID**: Enter `deepseek-v4-pro`
- **Model Display Name**: Enter `DeepSeek V4 Pro`



Restart the OpenCode client. In the model list within the conversation window, locate and select the newly configured model to start using it for conversation.



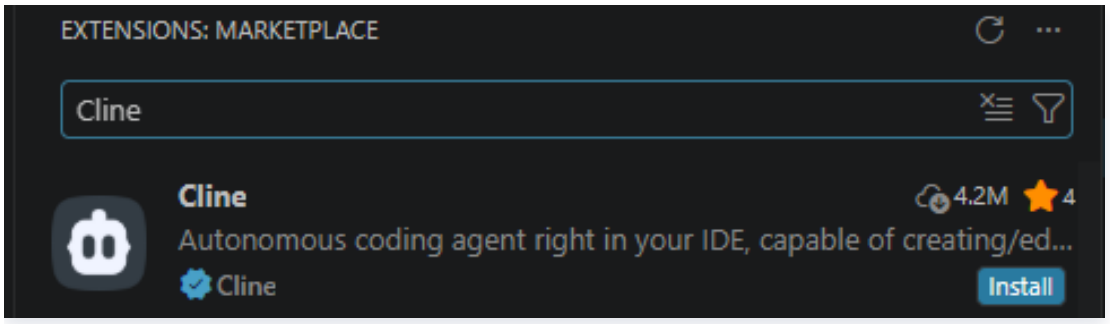
Cline

Last updated：2026-06-11 17:53:15

Cline is a VSCode extension for intelligent programming. You can integrate the Tencent DeepSeek V4 Pro model to complete complex programming tasks. This article demonstrates how to integrate a model for use with Cline, using DeepSeek V4 Pro as an example.

Installing Cline

Open VSCode, search for and install Cline from the Extensions Marketplace. For details, refer to Cline Official Installation Guide.

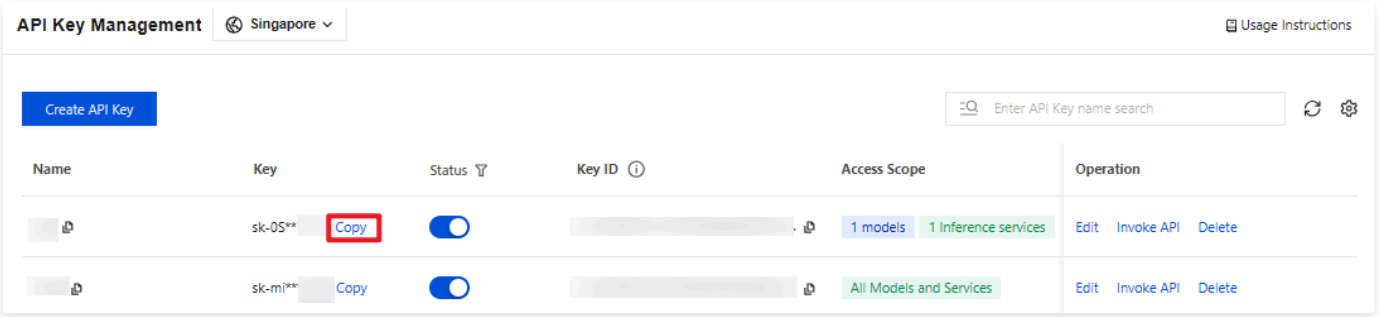


Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

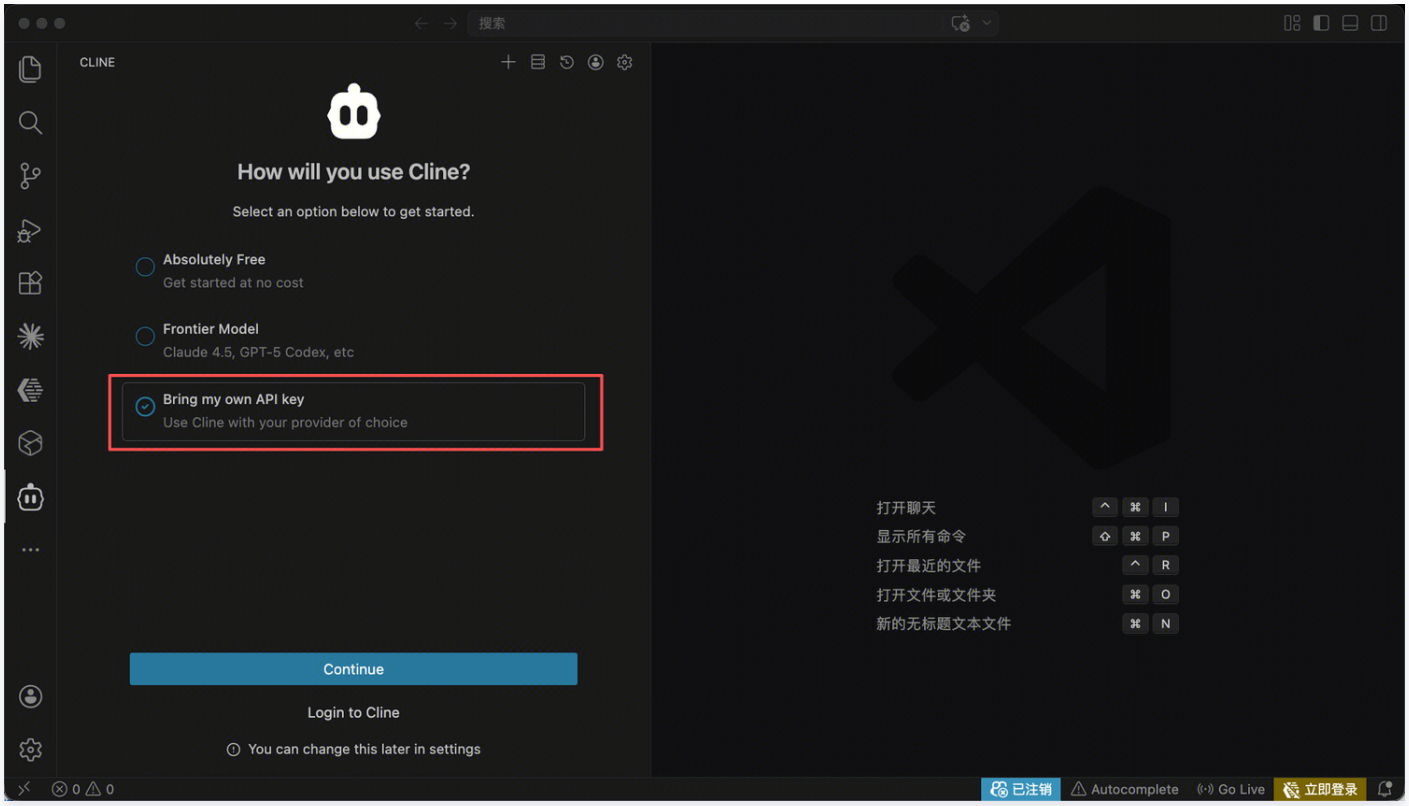
2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.



Configuring Cline

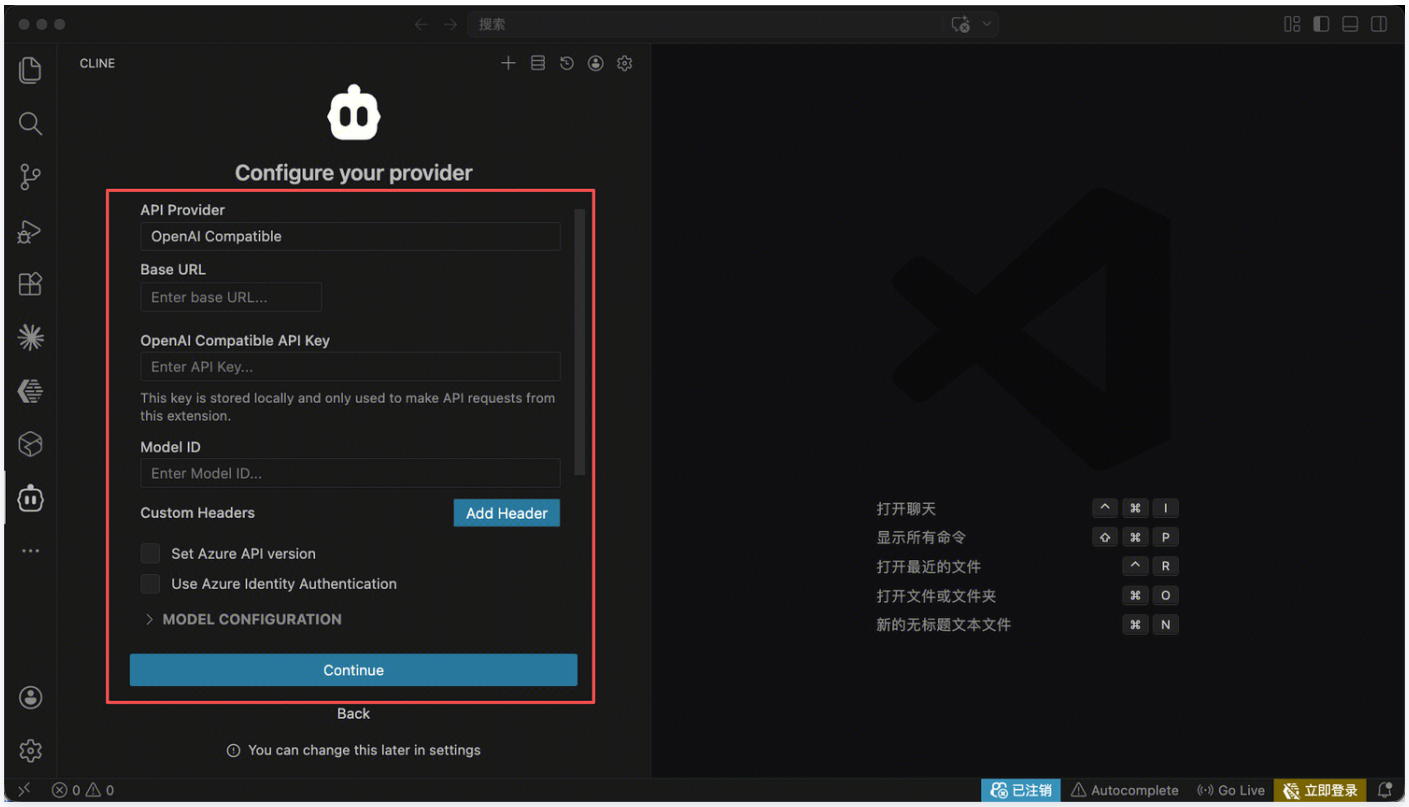
After the Cline extension is installed, select **Bring my own API key** and click **Continue** to open the configuration page. If you have used Cline before, you can click the settings button in the upper-right corner

of the main interface to configure it.



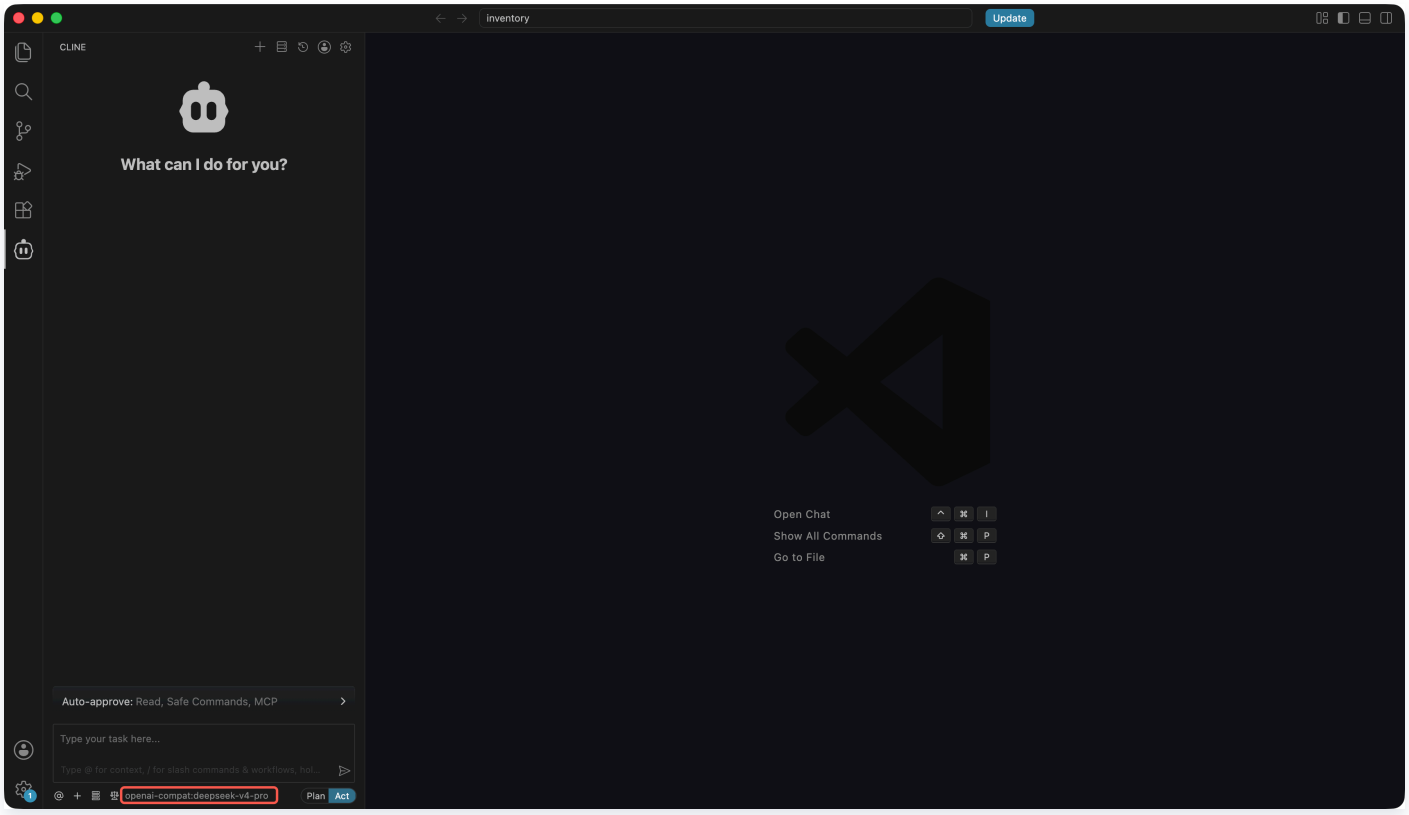
On the configuration page, configure the following information:

- **API Provider**: Select `OpenAI Compatible`.
- **Base URL**: `https://tokenhub-intl.tencentcloudmaas.com/v1`
- **OpenAI Compatible API Key**: Enter your API Key.
- **Model ID**: Enter `deepseek-v4-pro`



Using Cline

After completing the form, click **Continue** to finish the configuration. You can then use DeepSeek V4 Pro in Cline for conversation.



Cursor

Last updated：2026-06-11 17:53:15

Cursor is an AI-centric code editor built on VS Code. It enables you to converse directly with your codebase using natural language, thereby achieving automated code generation, intelligent refactoring, and rapid comprehension. This document demonstrates how to integrate a model, using DeepSeek V4 Pro as an example, into Cursor for use.

Installing Cursor

Download the installation package from the official website: Download and install Cursor via the Cursor official website.

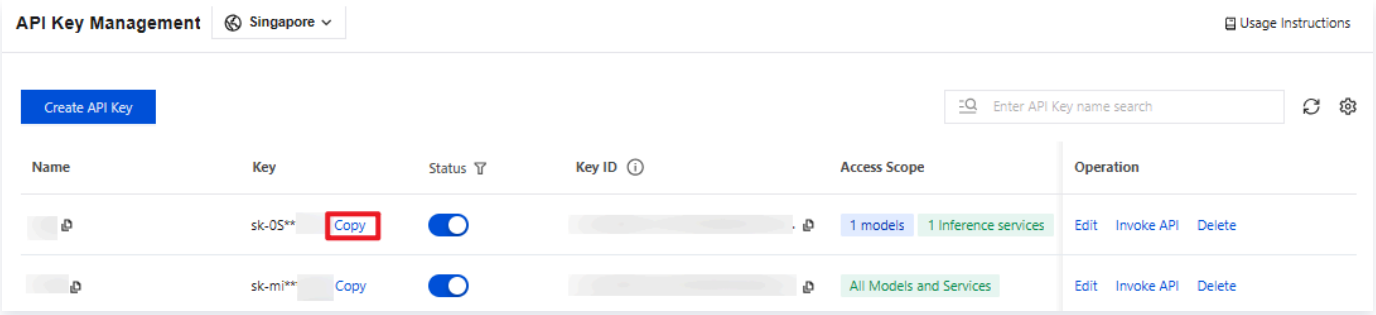
> **Note:**
> Due to Cursor's restrictions, only users subscribed to Cursor Pro or higher plans support custom model configuration.

Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

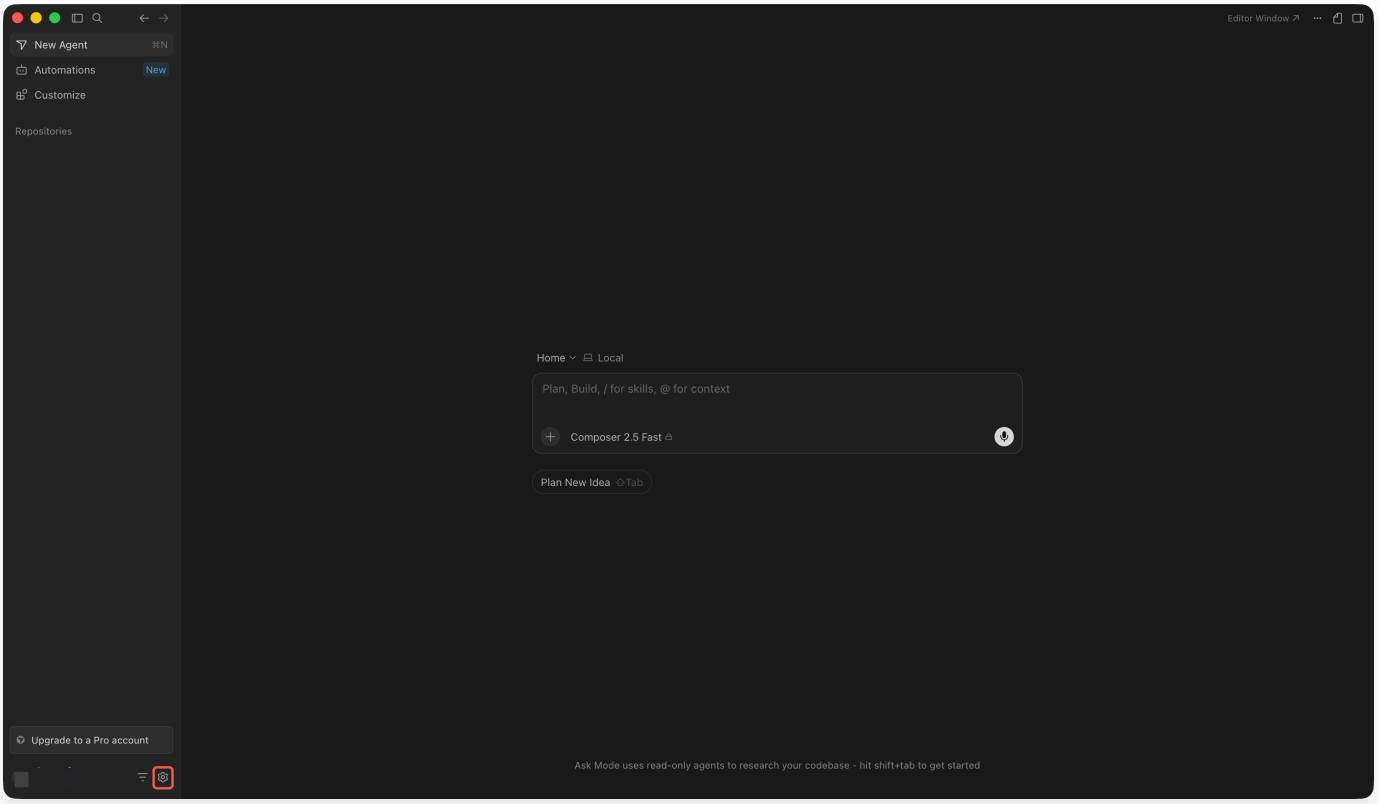
   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.

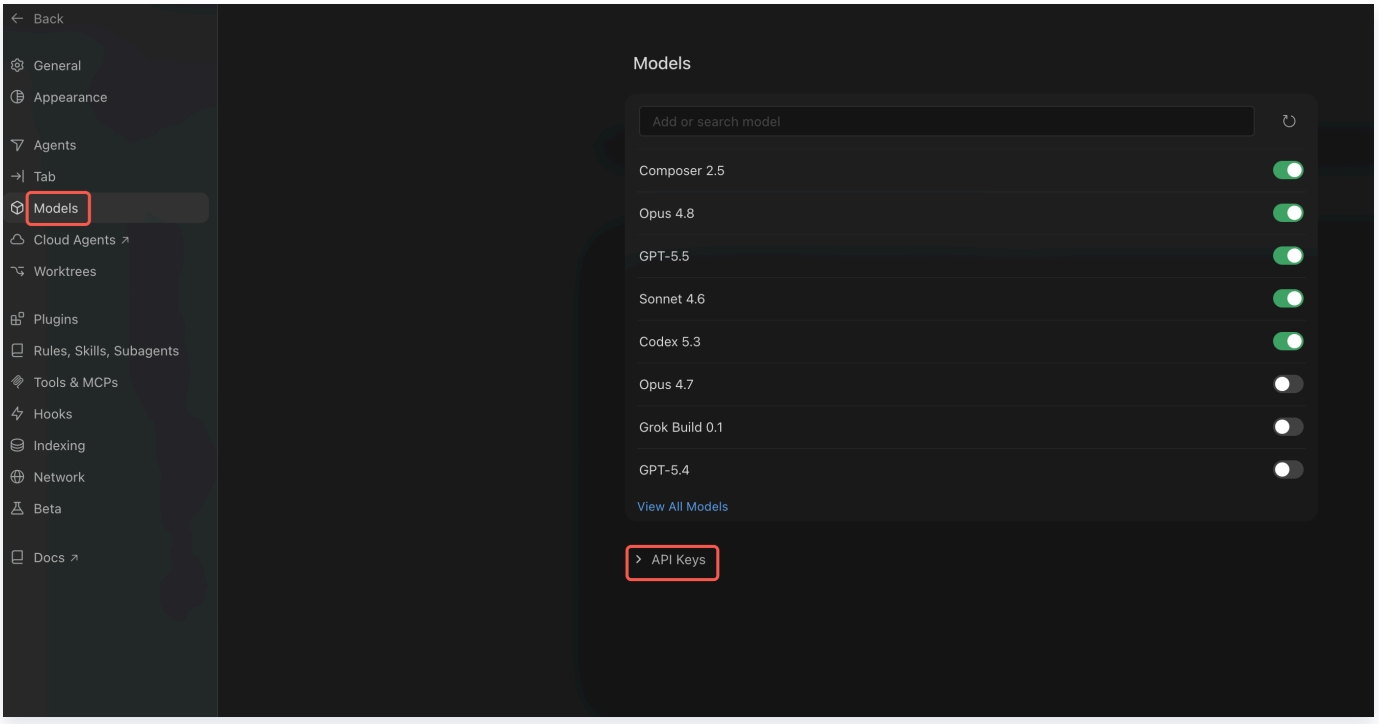


Configuring Cursor

1. After Cursor is installed, click the **Settings** icon in the lower-left corner within Cursor. Alternatively, press the shortcut key `Cmd/Ctrl + Shift + P` on your keyboard, type "Cursor Settings", and press Enter.

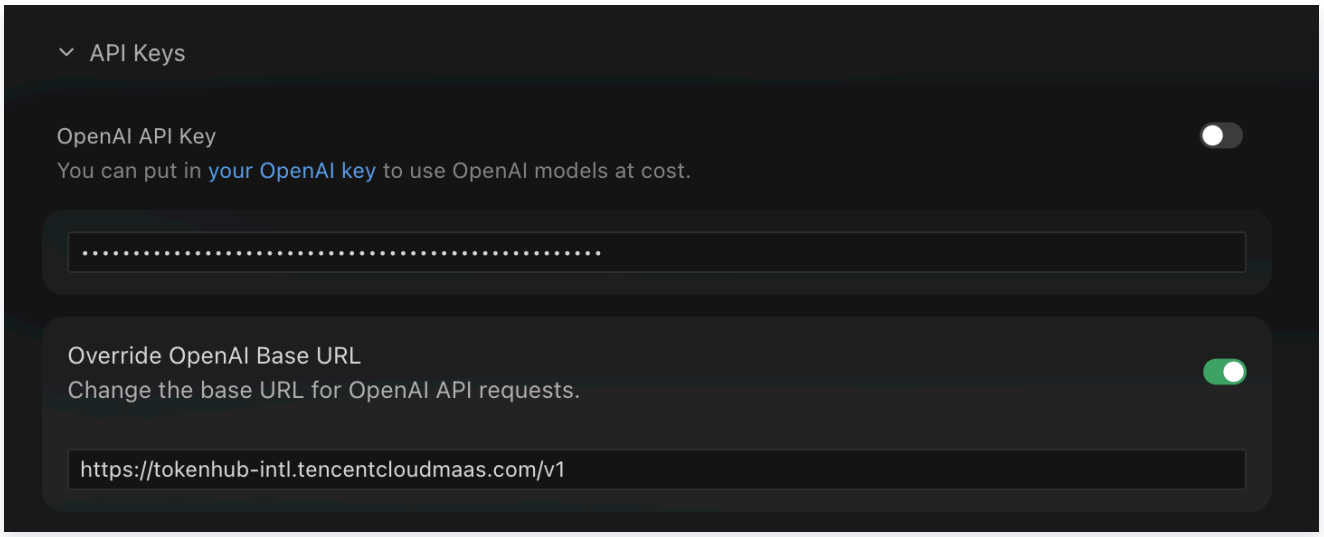


2. In the left menu bar, select and configure the **Models** module.

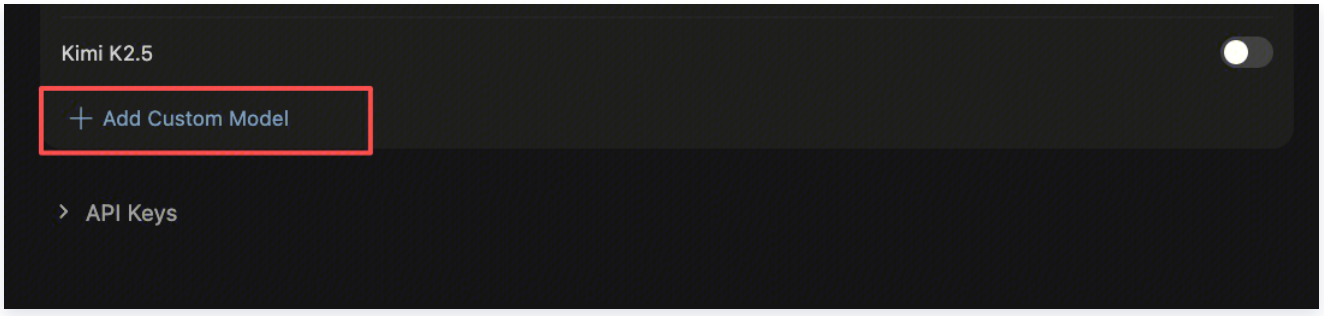


The specific configuration for the **Models** module is as follows:

- **OpenAI API Key**: Enter your API Key.
- **Override OpenAI Base URL**: `https://tokenhub-intl.tencentcloudmaas.com/v1`



- **Add Custom model**: Manually enter the model ID `deepseek-v4-pro`.



3. After completing the configuration, locate the deepseek-v4-pro model in Cursor's chat window to start a conversation.

Kilo Code

Last updated：2026-06-11 17:53:15

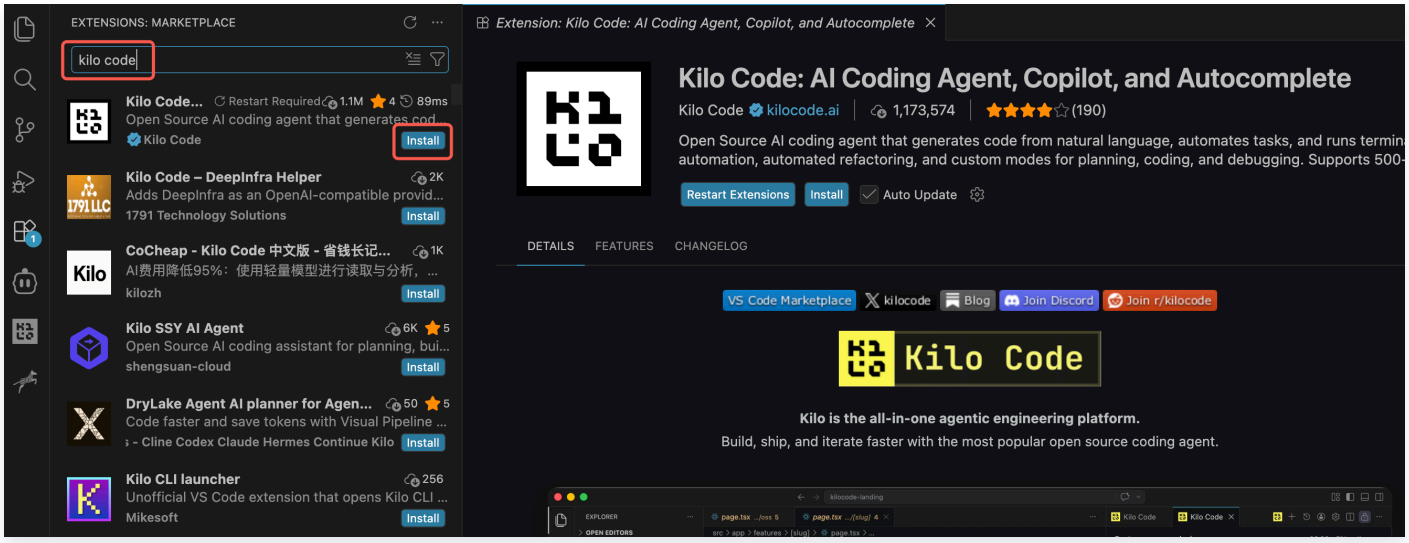
The Kilo extension (Kilo Code) is an open-source AI coding Agent. It can directly help you generate code, modify code, and run terminal tasks using natural language within the editor, and it provides capabilities such as code completion, refactoring, and debugging. This article demonstrates how to integrate a model for use with Kilo Code, using DeepSeek V4 Pro as an example.

Environment Requirements

Kilo Code supports VS Code (including VS Code-compatible editors such as Cursor, VSCodium, and Windsurf) and JetBrains IDEs. For specific environment requirements regarding editor editions, operating systems, Node.js, and others, refer to the official Kilo Code documentation. For details, see the Kilo Code Installation Guide.

Installing Kilo

Search for and install the Kilo Code extension in your IDE. This article demonstrates the installation and configuration process using VS Code as an example.
Open VS Code, search for Kilo Code in the Extensions Marketplace, and install it.

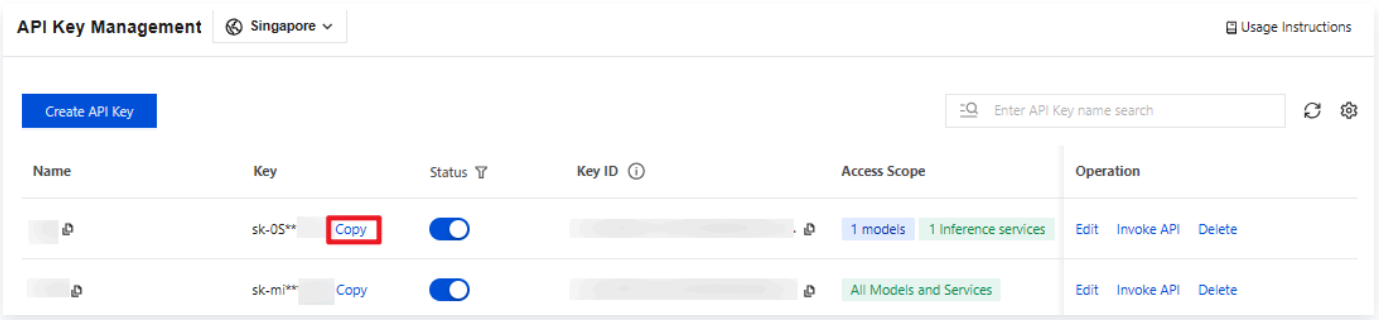


Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

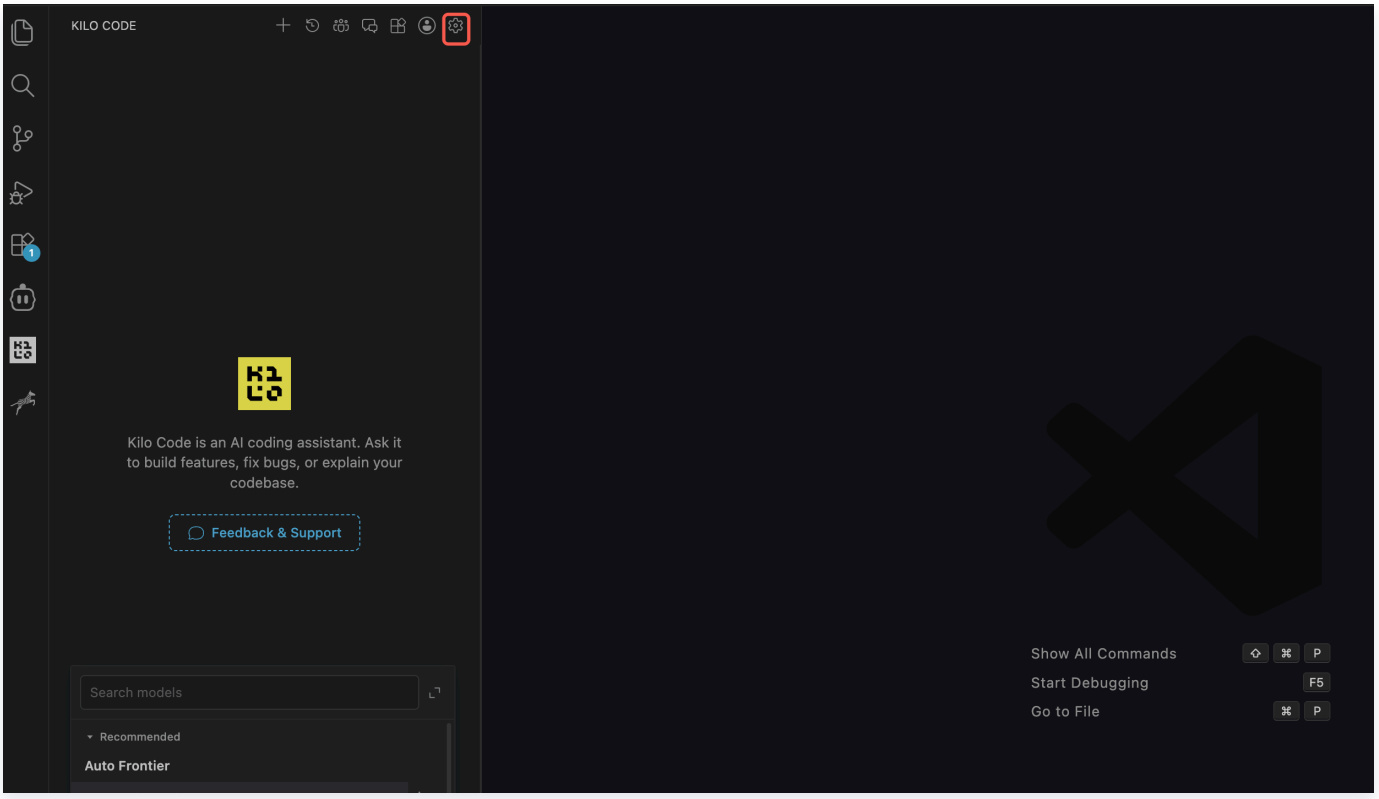
> **Note:**
> When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.

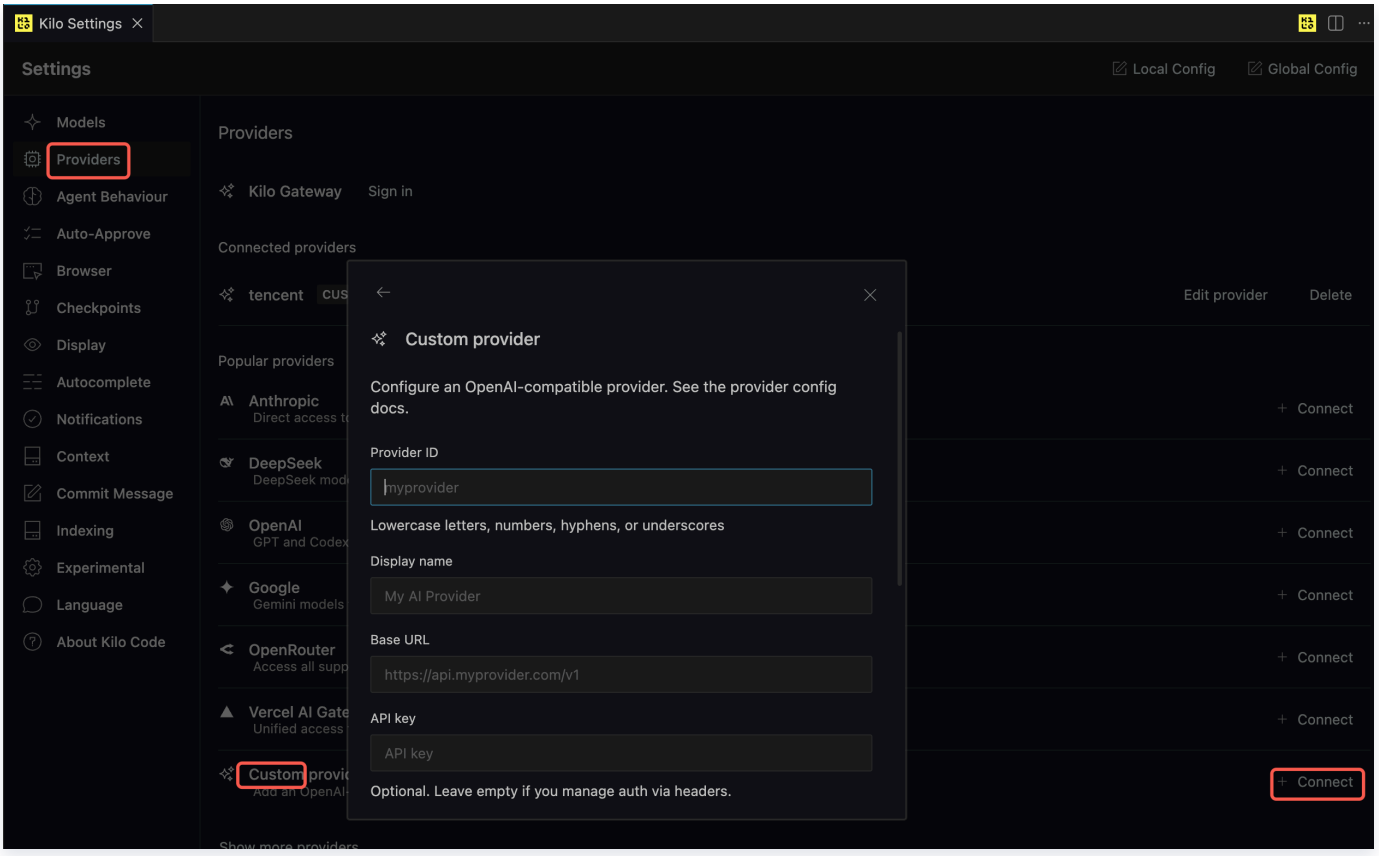


Configuring Kilo Code

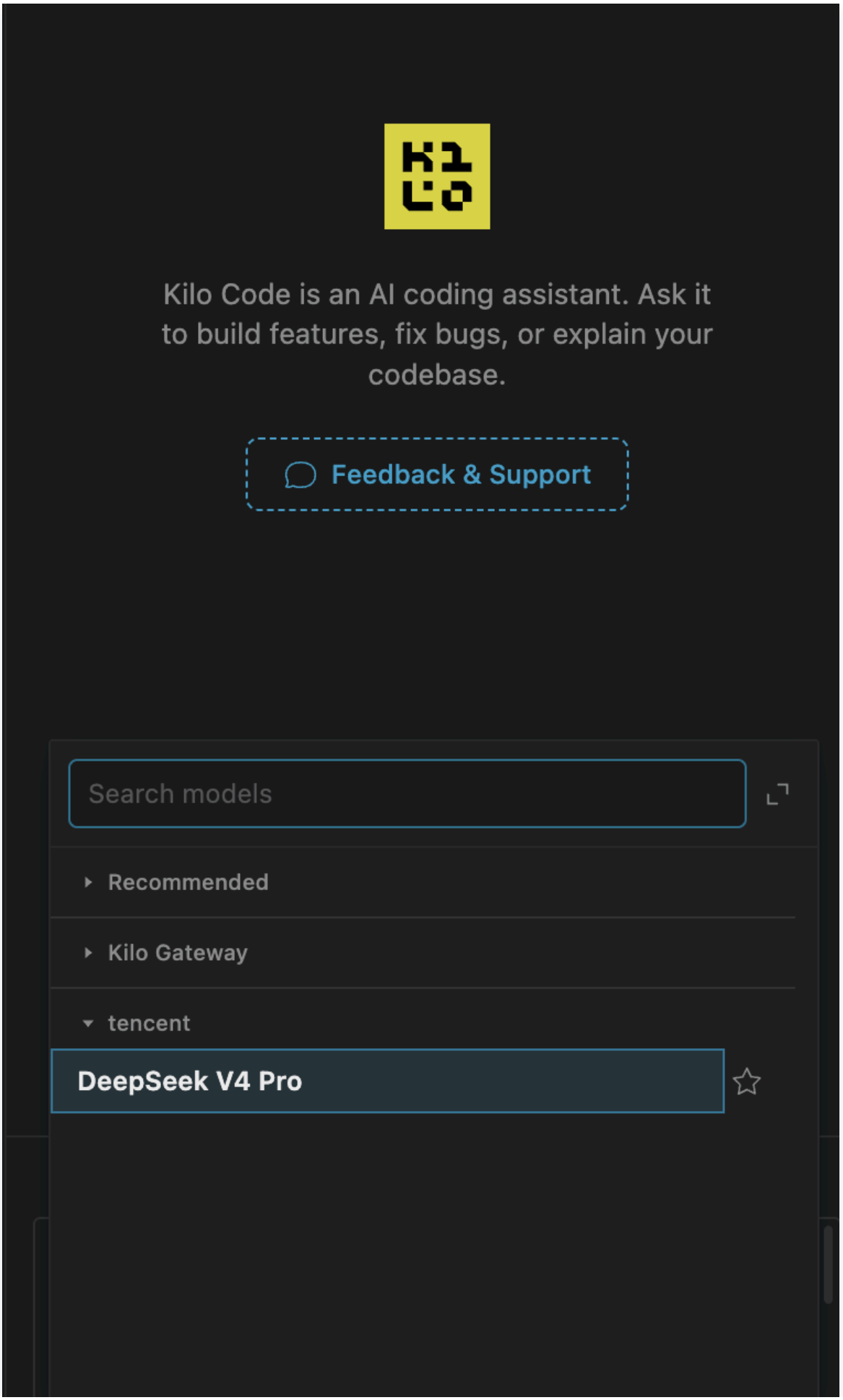
1. Open Kilo Code in VS Code, and click the **Settings** button in the upper-right corner to go to Kilo Settings.

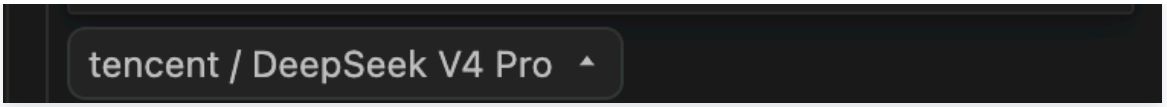


2. In Kilo Settings, choose **Providers > Custom Provider** and click "**+ Connect**" to open the configuration pop-up window.



3. Configure the following information and save to submit.
   - **Provider ID**: Enter the provider ID, for example, `tencent`.
   - **Display Name**: Enter the provider name, for example, `tencent`.
   - **Base URL**: `https://tokenhub-intl.tencentcloudmaas.com/v1`
   - **API Key**: Enter the API Key you created in the Obtain API Key step.
   - **Model ID**: Enter the model ID `deepseek-v4-pro`.
   - **Model Display Name**: Enter `DeepSeek V4 Pro`.
4. In the model list of Kilo Code, locate and select the **DeepSeek V4 Pro** model to start a conversation.





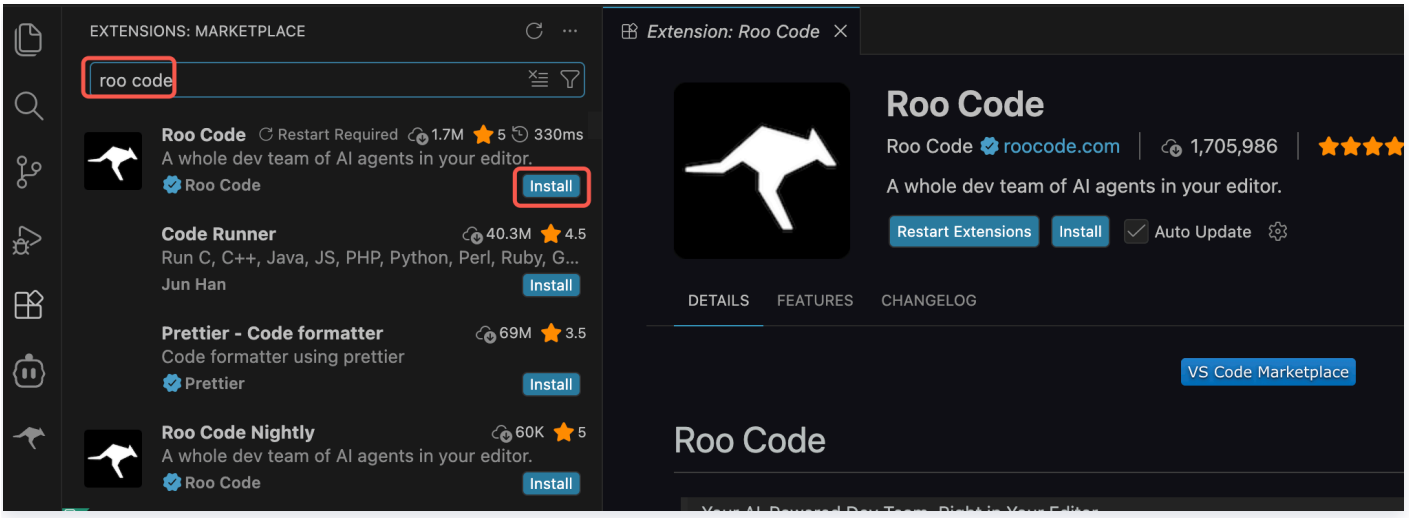
Roo Code

Last updated：2026-06-11 17:53:16

Roo Code is a VS Code extension for AI-powered programming. This article demonstrates how to integrate a model, using DeepSeek V4 Pro as an example, for use with Roo Code.

Installing Roo Code

If you have already installed Roo Code, you can skip this step.
Search for and install the Roo Code extension in your IDE. The following sections use VS Code as an example to demonstrate the installation and configuration process. After installation, it is recommended to restart the VS Code client.

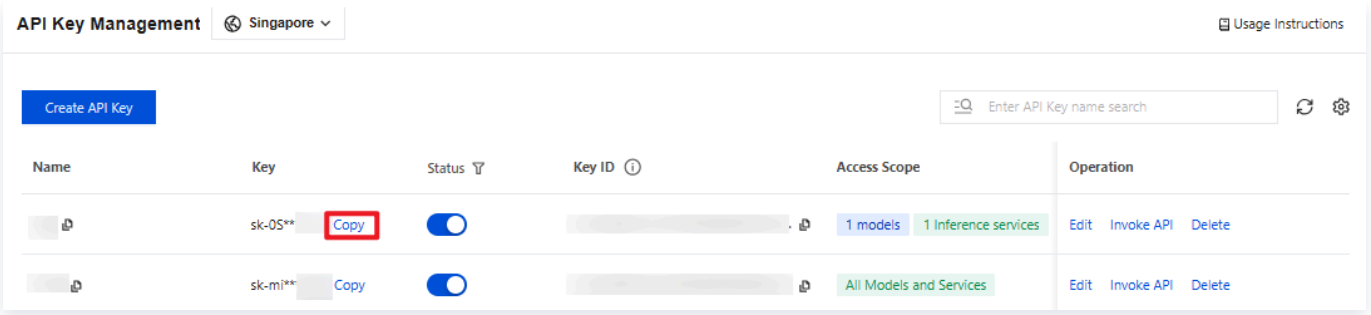


Obtaining an API Key

1. Go to the API Key Management page and click **Create API Key**. For operational details, see Create API Key.

   > **Note:**
   > When setting the **Access Scope**, if you select "Range", ensure that **DeepSeek V4 Pro** is selected.

2. After creation, be sure to **copy and securely store the API Key**. This information will be used in the subsequent process of configuring the tool.

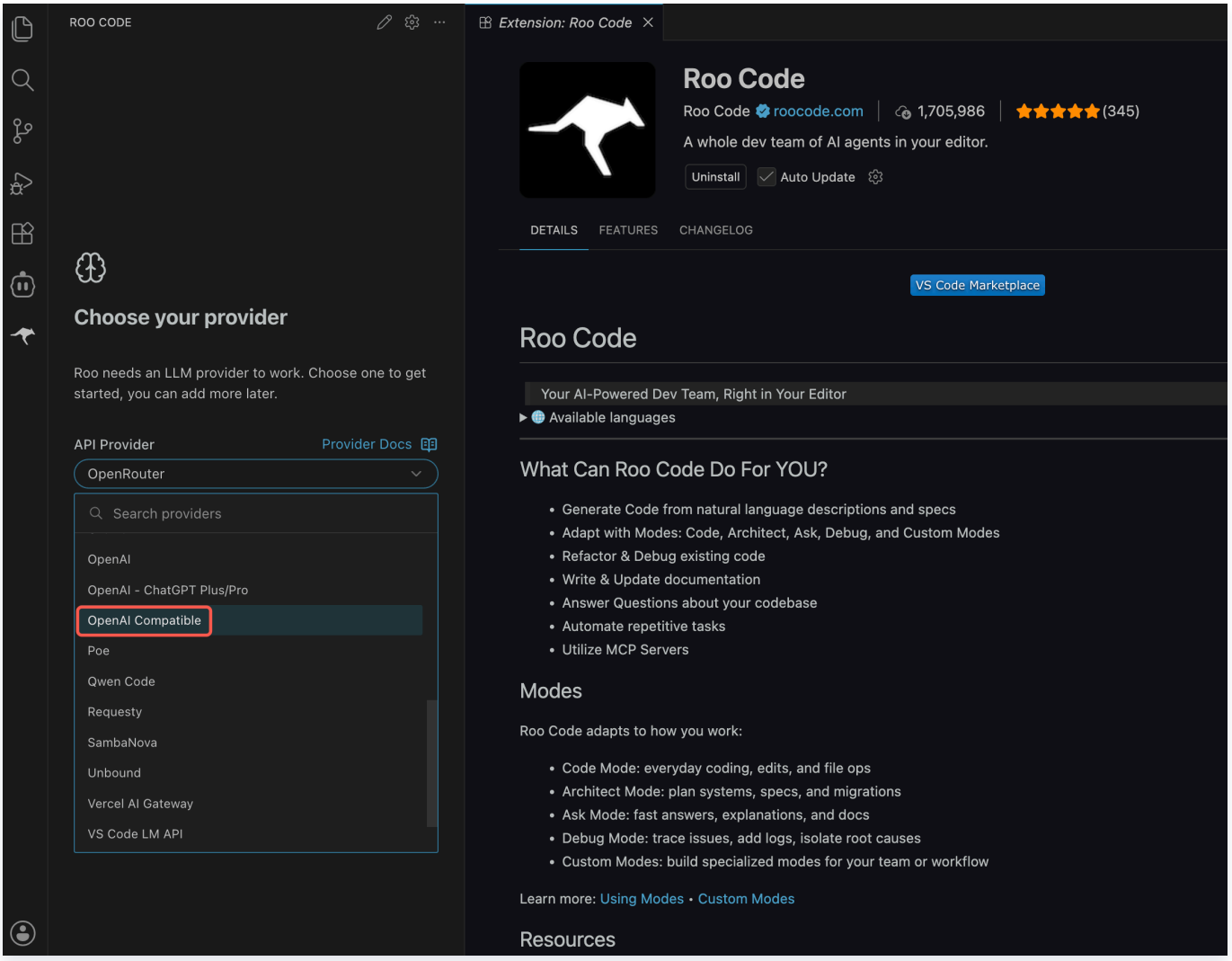


Configuring Roo Code

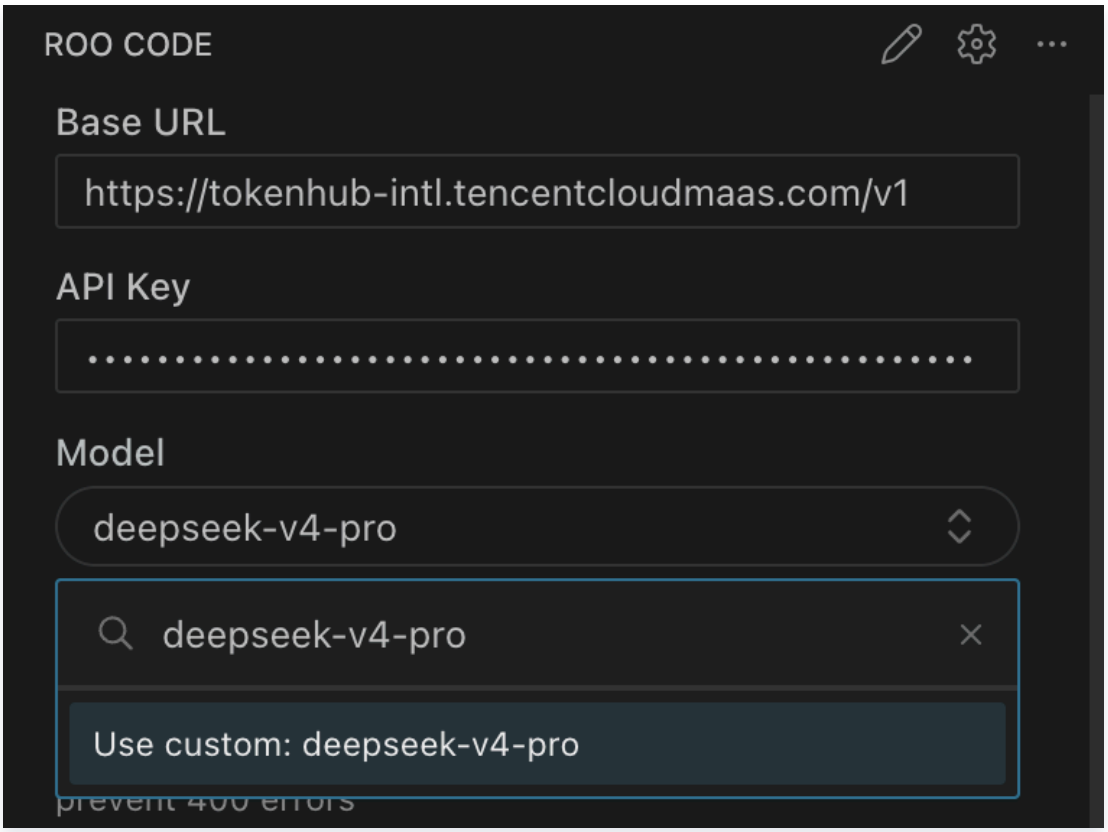
1. Open VS Code and select the Roo Code icon in the left sidebar.

> **Note:**
> If you are launching Roo Code for the first time, a mode selection page (such as Code, Architect, and Ask) will appear first. Select a mode as needed and complete the initial setup before proceeding with the subsequent steps.

2. API Provider: OpenAI Compatible.



3. Begin configuring the following information. Note that if you have used Roo Code before, you can directly click the Settings button in the upper-right corner of the main page to add or modify configurations.



   - **OpenAI Base URL**: `https://tokenhub-intl.tencentcloudmaas.com/v1`
   - **API Key**: Enter your API Key.
   - **Model**: Enter `deepseek-v4-pro` (using custom).
4. After completing the configuration, you can interact with DeepSeek V4 Pro in Roo Code.