Cursor官方指引文档（中文版）

文档说明：这是对Cursor官方指引文档的翻译，该文档写得相当清晰简洁，是新手入门开始学习Cursor，了解Cursor是什么，有哪些功能的最佳选择。但据我了解，很多人不知道这个文档的存在，或者因为语言的壁垒看这份文档非常费劲。即使有翻译插件的帮助，因为涉及到的技术名词和格式很难处理，插件翻译效果不佳，所以我为知识星球「AI编程：从入门到精通」的读者翻译了这份Cursor的官方指引文档。

译者：AI进化论-花生（Powered by Cursor），你可以通过以下两个视频平台和我的知识星球「AI编程：从入门到精通」学习更多Cursor使用技巧。

B站：https://space.bilibili.com/14097567

YouTube：https://www.youtube.com/@AIchain

星球专属GPTs——Cursor101：https://chatgpt.com/g/g-EhEcLTH48-cursor101

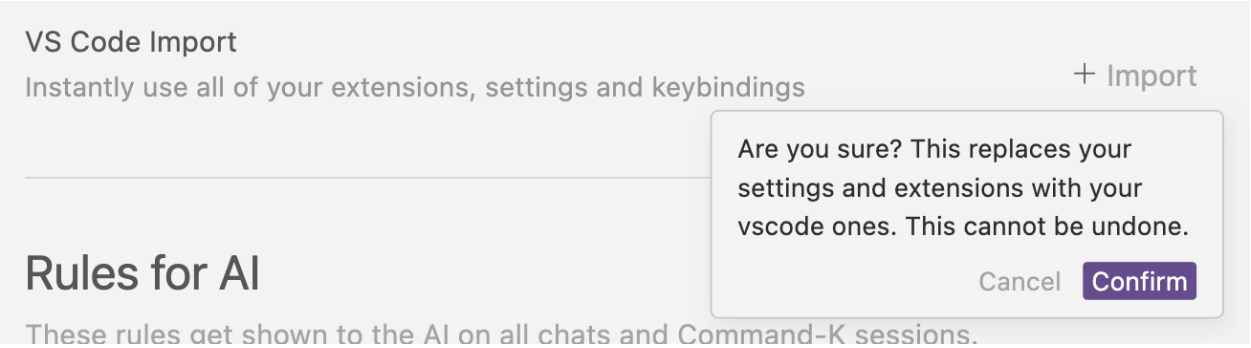
Get Started

Cursor和VS Code的关系

Cursor 是 VS Code 的一个分支，这使我们能够专注于优化与 AI 的编程方式，同时提供熟悉的文本编辑体验。

导入扩展、主题、设置和按键绑定

您可以通过一键将 VS Code 配置导入 Cursor。请导航至 `Cursor Settings` > `General` > `Account`。



保持更新

我们会定期将 Cursor 基于最新版本的 VS Code 进行重构。

为什么选择独立应用而非扩展？

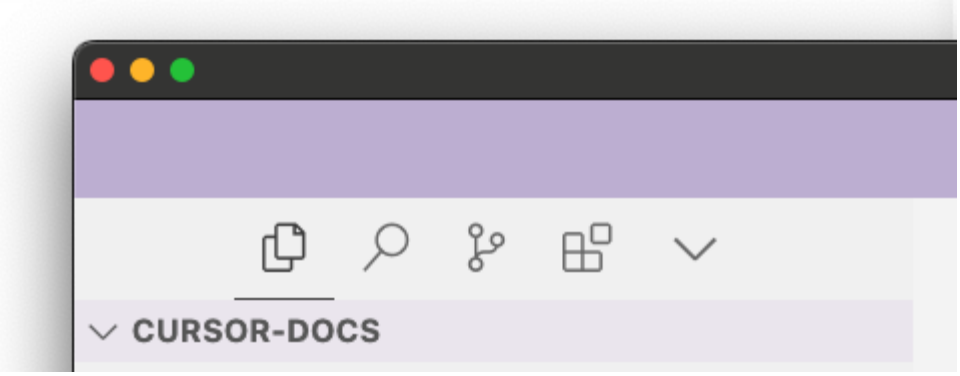
作为独立应用，Cursor 对编辑器的用户界面有更大的控制权，从而实现更高程度的 AI 集成。我们的一些功能，例如 Cursor Tab 和 CMD-K，无法作为现有编码环境的插件来实现。

设置

您可以通过点击右上角的齿轮按钮、按 `Ctrl/⌘ + Shift + J`，或使用 `Ctrl/⌘ + Shift + P` 并输入 `Cursor Settings` 打开 Cursor 特定的设置面板。

您可以使用 `Ctrl/⌘ + Shift + P` 打开 VS Code 特定设置，然后输入 `VS Code Settings`。

为什么 Cursor 中的活动栏是横向的？



活动栏默认是横向的，以节省聊天空间。如果您更喜欢正常的竖向活动栏，可以在 VS Code 设置中将 `workbench.activityBar.orientation` 设置为 `vertical`，然后重启 Cursor。

Cursor定价与用量

有关定价的更多信息，请访问 Cursor 定价。

订阅级别

Cursor 提供多个订阅级别以满足您的需求。

- Hobby
  - 14 天 **Pro 试用**
  - 50 次慢速 `premium` 模型使用
  - 200 次 `cursor-small` 使用
  - 2000 次 **completions** 使用
- Pro
  - 每月 500 次快速 `premium` 模型使用
  - 无限次慢速 `premium` 模型使用
  - 无限次 `cursor-small` 使用
  - 无限次 **completions** 完成
  - 每月 10 次 `Claude Opus` 使用
- Business
  - 使用信息与 `Pro` 层级相同

- 额外福利，详细信息见 **定价页面**

高级模型

GPT-4、GPT-4o 和 Claude 3.5 Sonnet 均被视为 `premium` 模型。

Pro 试用

所有新用户均会收到 14 天 Pro 试用，享有所有 Pro 功能。试用结束后，未升级的用户将恢复至 Hobby 计划。

快速及慢速请求

默认情况下，Cursor 服务器会尽量为所有用户提供快速 `premium` 模型请求。不过，在高峰期间，已用尽快速 `premium` 额度的用户将被转移到慢速池中，基本上是一条等待快速 `premium` 请求的用户队列。

该队列是公平的，Cursor 会尽力保持队列尽可能短。然而，如果您需要更多快速 `premium` 额度且不想等待，您可以在 **设置页面** 添加更多请求。

检查您的使用情况

您可以在 **Cursor Settings** 页面上查看您的使用情况。您也可以在 Cursor 应用中通过 `Cursor Settings` > `General` > `Account` 访问此页面，并为 Pro 用户按“Manage Subscription”，或为 Business 用户按“Manage”。

Cursor 的使用情况每月重置，以您的订阅开始日期为准。

可选的基于使用的定价

您可以通过访问您的 **设置页面** 选择可选的基于使用计费的定价，适用于超出计划的请求。

基于使用的定价详情：

- 基于使用的定价按日历月份计算（不一定与您的账单周期相同，大约在每月第二或第三天计费）
- 如果您立即取消请求或出现错误，我们将不计入
- 您可以配置硬性限制，每月不会产生超过该限制的费用（适用于基于使用的定价）
- 目前，基于使用的定价仅适用于 Claude 3 Opus 以及一些长上下文聊天模型。

Tab（Cursor代码自动补全功能）

概述

Cursor Tab 是我们原生的自动补全功能。它是一个更强大的 Copilot，能够精确建议整个代码改动并具备出色的记忆能力。

Cursor Tab 由自定义模型驱动，可以：

- 提议光标周围的编辑，而不仅仅是插入额外代码。
- 同时修改多行内容。
- 基于您最近的更改和 linter 错误给出建议。

免费用户可以无成本获得 2000 次建议。Pro 和 Business 计划用户可以获得无限次建议。

用户界面

当 Cursor 仅添加额外文本时，完成的建议将以灰色文本显示。如果建议修改了现有代码，则会以差异弹窗形式显示在您当前行的右侧。

您可以按 `Tab` 接受建议，按 `Esc` 驳回建议。要逐字部分接受建议，请按 `Ctrl/⌘ →`。要拒绝建议，只需继续输入，或使用 `Escape` 来取消/隐藏建议。

每当您进行按键或光标移动时，Cursor 会尝试基于您的最近更改做出建议。然而，Cursor 并不会总是显示建议；有时模型会预测没有更改需要被做出。

Cursor 可以在您当前行的上面一行到下面两行之间进行更改。

切换

要启用或禁用该功能，将鼠标悬停在应用右下角状态栏上的“Cursor Tab”图标上。

从 GitHub Copilot 迁移

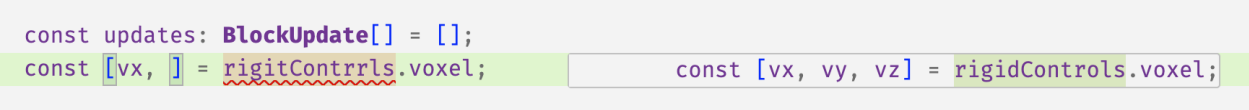
Tab 改进

Cursor 和 GitHub Copilot 补全代码的最大区别在于实现方式。

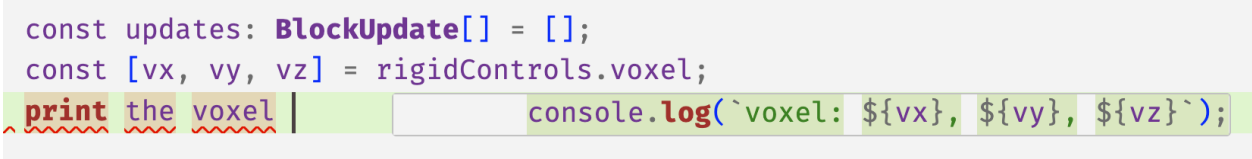
GitHub Copilot 只能在光标位置插入文本，无法编辑周围代码或删除文本。

Cursor 不仅可以在光标处插入文本，还能做到更多：

- 多字符编辑



- 基于指令的编辑



此外，Cursor 还在上下文窗口中维持您最近更改的历史记录，因此它能知道您想做什么。

从 GitHub Copilot 迁移

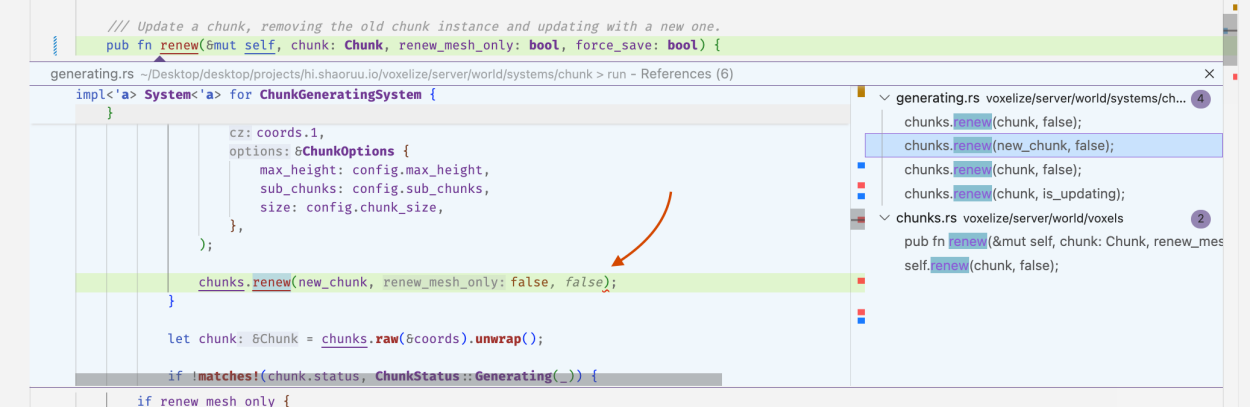
由于 Cursor 默认与 GitHub Copilot 一起使用，您可能同时安装了 GitHub Copilot 和 Cursor。我们建议在使用 Cursor 时关闭 GitHub Copilot。

默认情况下，Cursor 优先于 GitHub Copilot。如果您希望使用 GitHub Copilot，可以在设置中 **禁用 Cursor**。

高级功能

Peek 中的 Tab

您还可以在“转到定义”或“转到类型定义”的 Peek 视图中使用 Cursor Tab。这在例如为函数调用添加新参数时非常有用。



我们特别喜欢在 vim 中结合 `gd` 使用此功能，以便例如修改一个函数定义，然后一次性修复所有用法。

Cursor 预测

Cursor 还可以预测您在接受编辑后会进行什么。如果可用，您可以按 Tab 键进入下一个位置，允许您通过编辑连续跳转。

```
const updatesToServer: BlockUpdate[] = [];
const [vx, vy, vz] = rigidControls.voxel;

for (let i = 0; i < allBlocks.length; i++) {
  const block = allBlocks[i];
  const x = i % perRow;
  const y = 0;
  const z = Math.floor(i / perRow);

  updatesToServer.push({ vx: vx + x, vy: vy + y, vz: vz + z, type: block.id });
}
```

Cursor 预测了下一个位置，并在此处建议了编辑。

部分接受

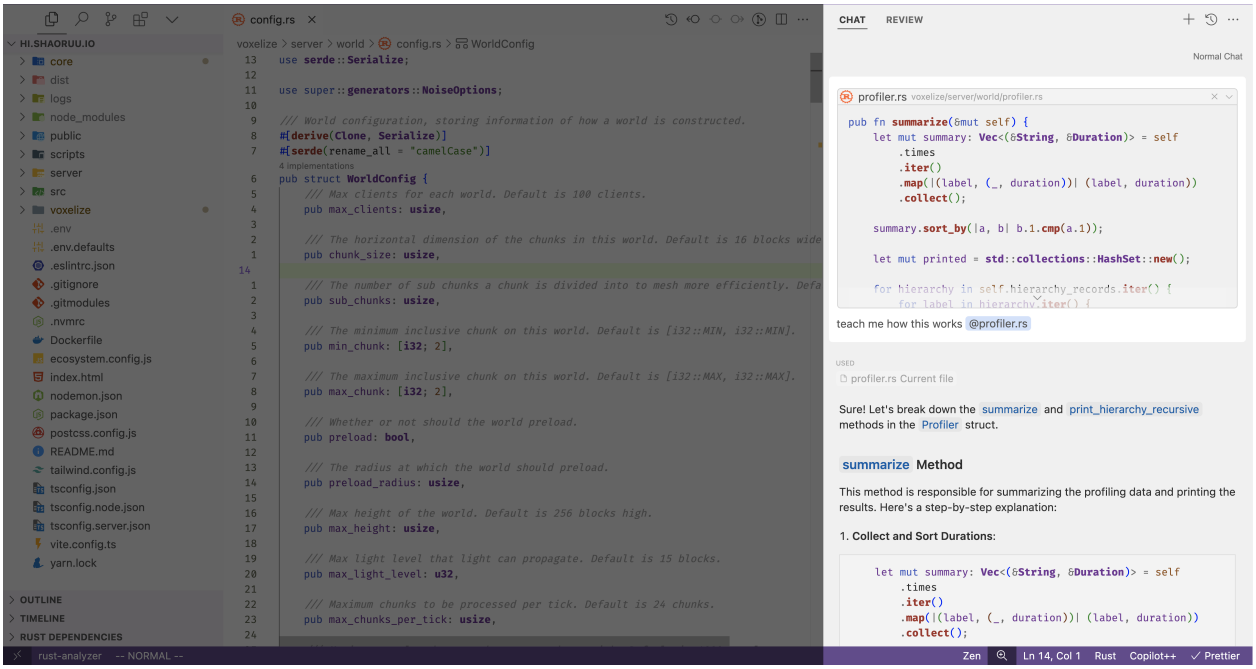
您可以按 `Ctrl/⌘` 和右箭头接受建议的下一个单词（或
将 `editor.action.inlineSuggest.acceptNextWord` 设置为您的优先按键绑定）。

要启用部分接受，请导航至 `Cursor Settings` > `Features` > `Cursor Tab`。

Chat（Cursor聊天对话模式）

概述

Cursor Chat 让您可以在代码库中提出问题或解决问题，所有采用最强大的语言模型，在编辑器中实现。



为了让语言模型提供优质答案，需要了解与您的代码库相关的特定内容——上下文。

Cursor 提供多种内置功能来提供上下文，例如自动包含整个代码库的上下文、在线搜索、文档索引以及用户指定的代码块引用。这些功能旨在消除在与语言模型进行代码交互时繁琐的复制粘贴工作。

默认情况下，Cursor Chat 位于 AI 面板中，面板位于主侧边栏的另一侧。您可以通过按 `Ctrl/⌘ + L` 切换 AI 面板，打开时它将聚焦聊天。要提交查询，请按 `Enter`。

用户和 AI 消息

用户消息包括您输入的文本以及您引用的上下文。您可以回到任何先前的用户消息进行编辑并重新运行查询。这将覆盖之后的所有消息并生成新的消息。

AI 消息是您选择的 AI 模型生成的响应。它们与之前的用户消息成对并列。AI 消息中可能包含解析的代码块，您可以通过 **即时应用** 将其添加到代码库中。

所有用户/AI 消息汇聚在同一线程中称为聊天线程，每个聊天线程都保存在您的聊天记录中。

聊天记录

通过点击 AI 面板右上角的“Previous Chats”按钮，或按 `Ctrl/⌘ + Alt/Option + L`，您可以查看聊天记录。点击任何聊天线程可查看该线程内的消息，您还可以通过点击笔图标修改线程标题，或通过悬停在线程上点击垃圾桶图标删除该线程。

Cursor 线程的标题仅为第一条用户消息的前几个单词。

默认上下文

默认情况下，Cursor Chat 包含当前文件作为上下文。您可以按 `Alt/Option Enter` 提交查询而不包括任何上下文，或在 `Cursor Settings` > `Features` > `Chat` 中启用 `Default to no context`。以无上下文请求开始的聊天将不会在任何消息中添加上下文。

在您输入时，可以在输入框下方的展示框中查看将包含的上下文内容。

添加上下文

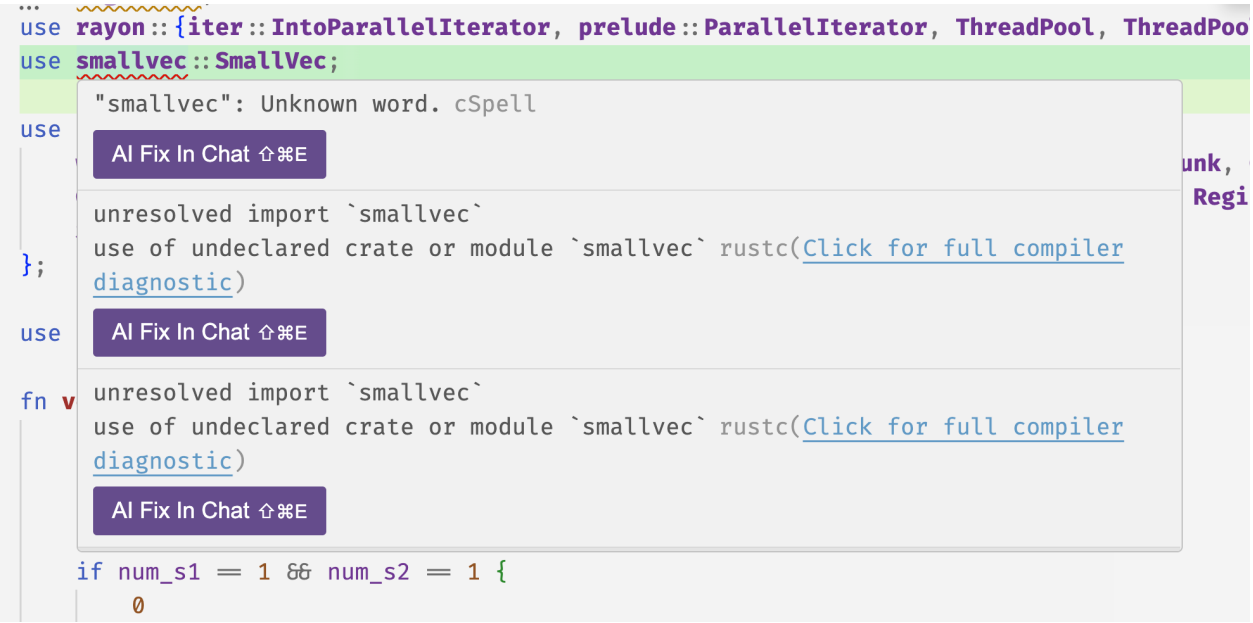
默认情况下，用户消息将包括您输入的文本，以及您引用的上下文。您可以通过 @ 符号为每个消息添加更多自定义上下文，默认情况下，当前查看的文件也会在用户消息中用作上下文。

有关更多信息，请查看 **@ 符号** 的相关页面。

聊天中的 AI 修复

使用聊天中的 AI 修复功能可以方便地修复您代码库中的 lint 错误。为此，您只需将鼠标悬停在编辑器中的错误上，然后点击出现的蓝色 AI 修复按钮。

该功能的快捷键是 `Ctrl/⌘ + Shift + E`。



长上下文聊天（测试版）

您可以通过访问 `Cursor Settings` > `Beta` > `Long Context Chat` 来启用长上下文聊天。启用后，您可以使用 `Ctrl/⌘ .` 切换不同的聊天模式。

长上下文聊天允许您将整个文件夹作为上下文，因为支持的模型具有更大的上下文窗口。有关仅支持长上下文模型的更多信息，请 **查看这里**。

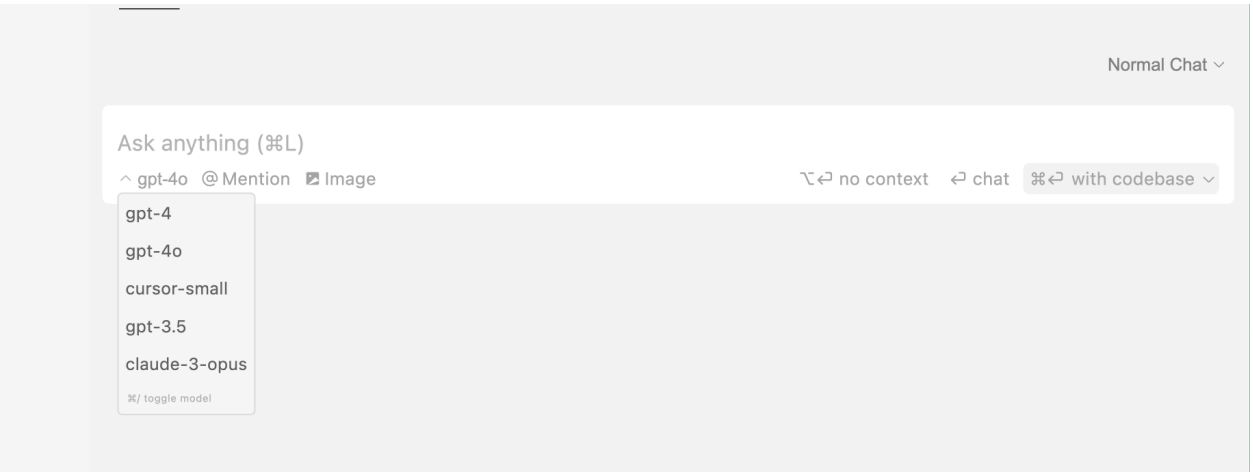
自定义

选择 AI 模型

通过 **模型切换** 选择您首选的 AI 模型，Cursor Chat 将使用该模型生成响应。您可以通过按 `Ctrl/⌘ /` 在模型之间切换。

默认情况下，Cursor Chat 的 AI 模型使用 OpenAI 的 `GPT-4`（具体而言，`gpt-4` 标签指向我们的 GPT4-Turbo 实例）。

您为聊天选择的 AI 模型将保存用于将来的会话，因此每次打开 Cursor Chat 时都无需更改。



内部编辑器

通过点击横向的“更多”按钮，再点击“在编辑器中打开聊天”按钮，Cursor Chat 也可以作为编辑器选项卡使用。此时，Cursor Chat 将表现得像一个常规的编辑器选项卡。
按 `Ctrl/⌘ + L` 将调出聊天选项卡并聚焦于其上。

设置

您可以在 `Cursor Settings` > `Features` > `Chat` 中自定义 Cursor Chat。

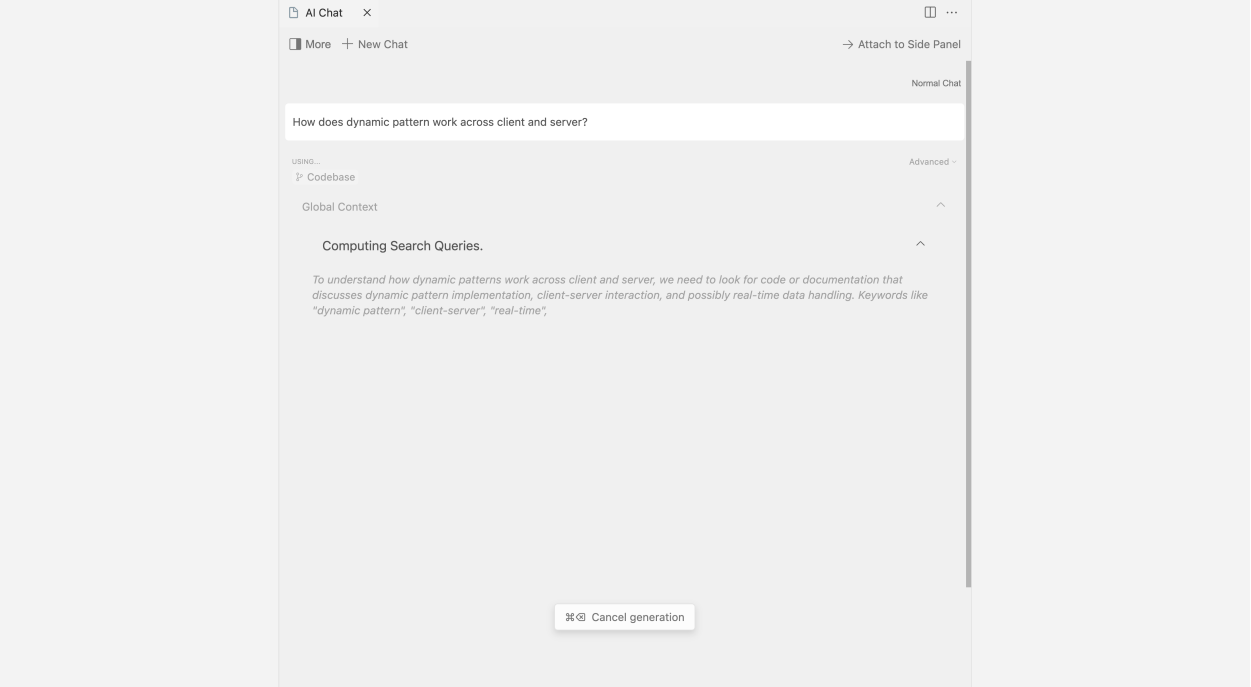
这些设置包括：

- 始终在线搜索答案。
  - 这将使 AI 模型在每个查询时浏览网站以获取最新信息。
- 添加聊天淡入淡出动画。
  - 这将在 AI 消息生成时添加平滑动画。
- 默认无上下文。
  - 这将使 AI 模型仅使用用户消息作为上下文，而不包括任何额外的上下文，例如当前文件。
- 自动滚动聊天。
  - 当在聊天线程底部时，这会自动滚动聊天。
- 聊天窗中的滚动条变窄。
- 开始新聊天时显示聊天历史。

With Codebase (代码库)

Default Codebase Chat (默认代码库聊天)

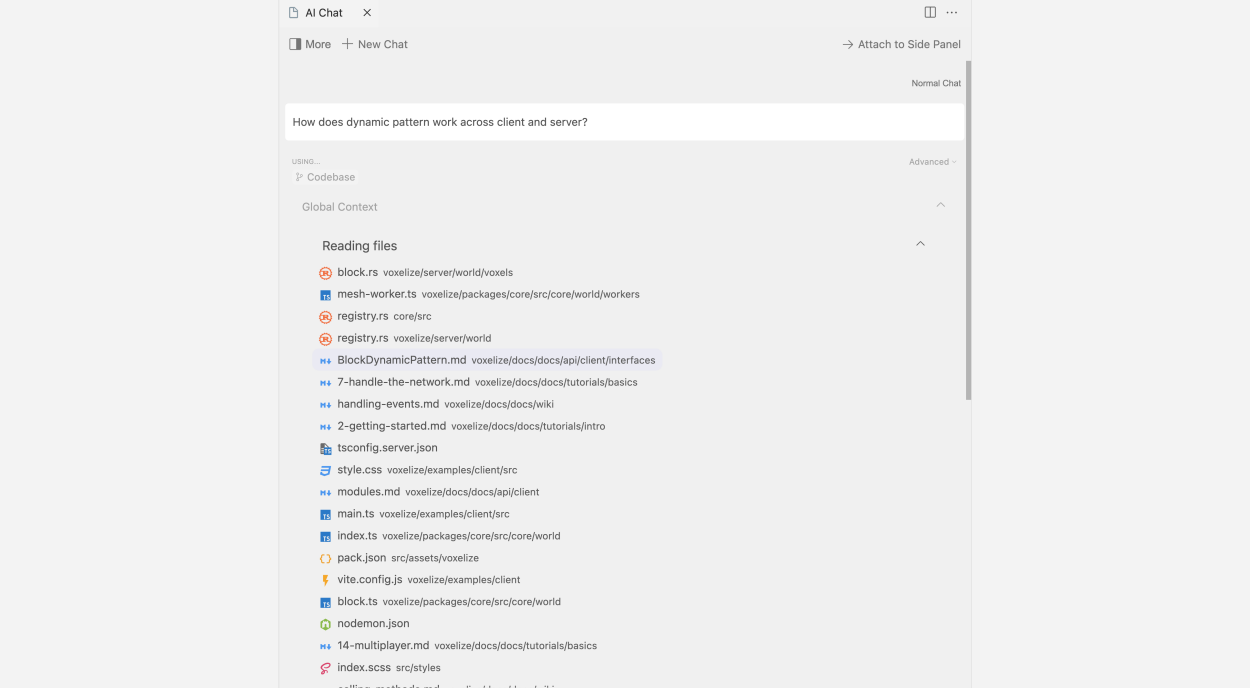
如果代码库未被 索引，Cursor Chat 将首先尝试生成一些搜索查询，以便在您的代码库中进行检索。为了提高准确性，建议使用 嵌入搜索。



Embeddings Search (嵌入搜索)

通过 代码库索引，Cursor Chat 能够准确地根据您的代码库生成响应。

在输入消息后按 `Ctrl/⌘ + Enter`，Cursor Chat 会扫描您的索引代码库以查找相关代码片段。这对于快速将代码片段融入对话上下文非常有用。若需要对代码库搜索的更好控制和更高的准确性，可以使用 `@codebase`。



Advanced Codebase Search (高级代码库搜索)

使用 `@Codebase` 时，Cursor 的代码库聊天会进行更详细的搜索。

有关 `@Codebase` 的更多信息，请**点击这里**。

Apply (应用)

Cursor 的 `Apply` 功能允许您快速将聊天中的代码块建议整合到您的代码中。

Apply Code Blocks (应用代码块)

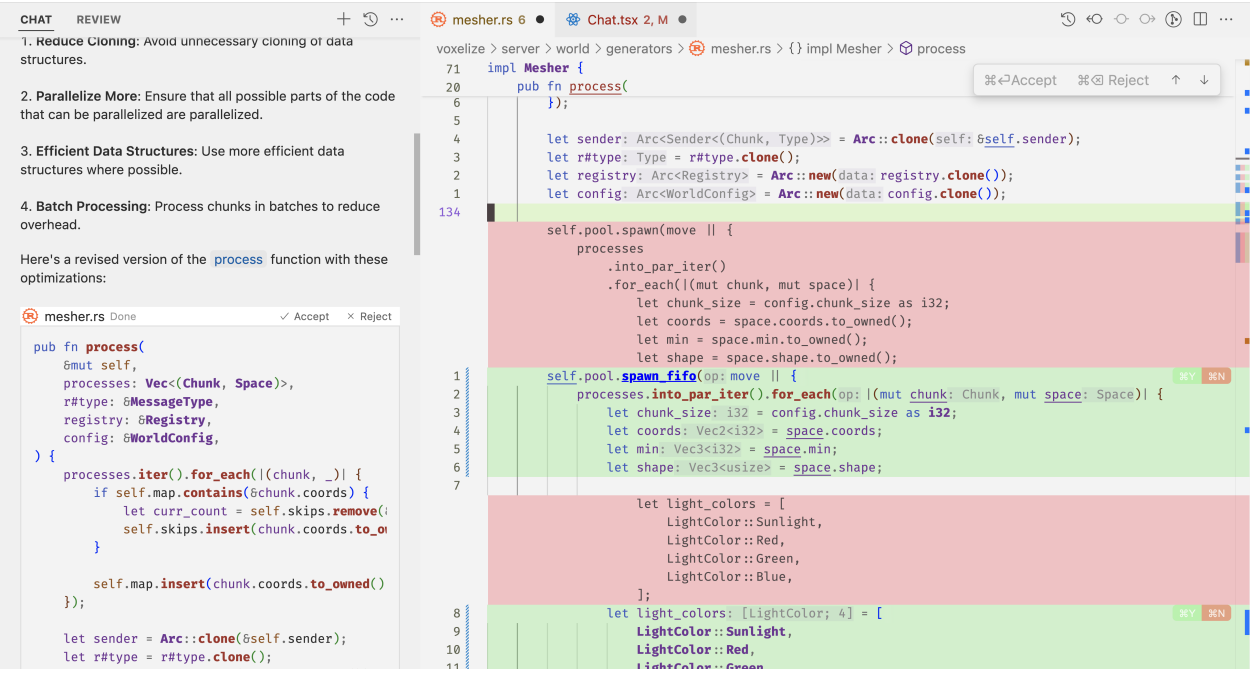
要应用代码块建议，您可以点击每个聊天代码块右上角的播放按钮。

这将编辑您的文件，将 Chat 生成的代码整合到其中。由于您可以在 Chat 中提供更多上下文并与模型进行多次互动，我们建议在进行更复杂的 AI 驱动代码更改时使用 Chat 和 Apply 结合的方法。

Accept or Reject (接受或拒绝)

一旦应用了代码块，您可以查看更改差异并选择接受或拒绝。您也可以直接点击聊天代码块右上角的"Accept"或"Reject"按钮。

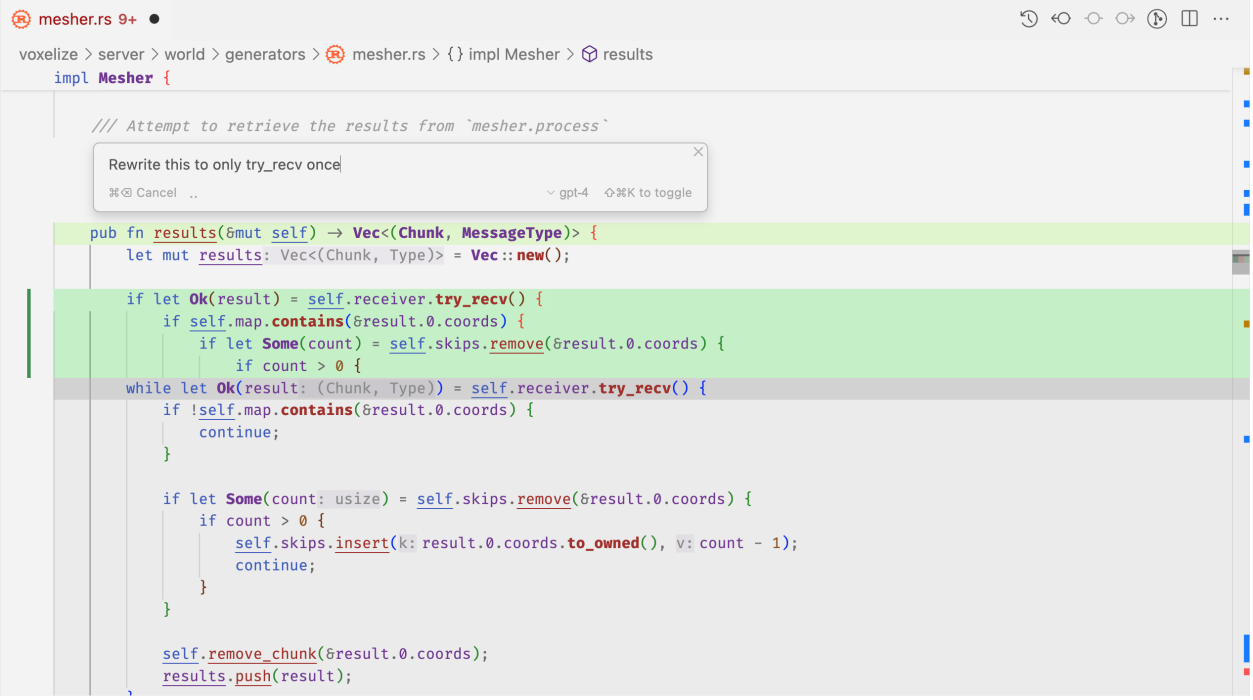
`Ctrl/⌘ Enter` 接受，`Ctrl/⌘ Backspace` 拒绝。



Cmd K

Cmd K 概述

Cmd K，也称为 Windows/Linux 上的 “Ctrl K”，允许您在编辑器窗口生成新代码或编辑现有代码。



Prompt Bars（提示框）

在 Cursor 中，按 `Ctrl/Cmd K` 时出现的条称为“提示框”。它类似于聊天的 AI 输入框，您可以正常输入，或使用 @ 符号 引用其他上下文。

Inline Generation（原地生成）

如果在按 `Ctrl/Cmd K` 时未选择任何代码，Cursor 将根据您在提示框中输入的提示生成新代码。

Inline Edits（原地编辑）

对于原地编辑，您可以直接选择要编辑的代码，并在提示框中输入。

Follow-up Instructions（后续提示）

在每次生成后，您可以通过在提示框中添加更多指令并按 `Enter` 来进一步细化提示，以便 AI 基于您的后续指示进行再生成。

Default Context（默认上下文）

默认情况下，Cursor 会尝试寻找不同类型的有用信息以提升代码生成，除了您手动包含的 @ 符号。

额外的上下文可能包括相关文件、最近查看的文件等。在收集后，Cursor 会根据相关性对上下文项进行排序，并将最重要的项保留在大型语言模型的上下文中。

Quick Question（快速提问）

如果您在提示框中按 `Option/Alt Enter`，Cursor 将回答您关于所选内容及附加上下文的任何问题。

此对话的内容可进一步用于后续生成，因此您可以在 Cursor 给出答案后简单输入“do it”以生成代码。

Terminal Cmd K（终端的Cmd+K）

在内置的 Cursor 终端中，您可以按 `Ctrl/⌘ K` 在终端底部打开提示框。此提示框允许您描述在终端中希望执行的操作，Terminal Cmd K 将生成相应的命令。您可以通过按 `esc` 来接受该命令，或立即通过按 `Ctrl/⌘ + Enter` 来执行该命令。

默认情况下，Terminal Cmd K 会将您的最近终端历史记录、您的指令以及提示框中的其他内容视为上下文。

Context（上下文）

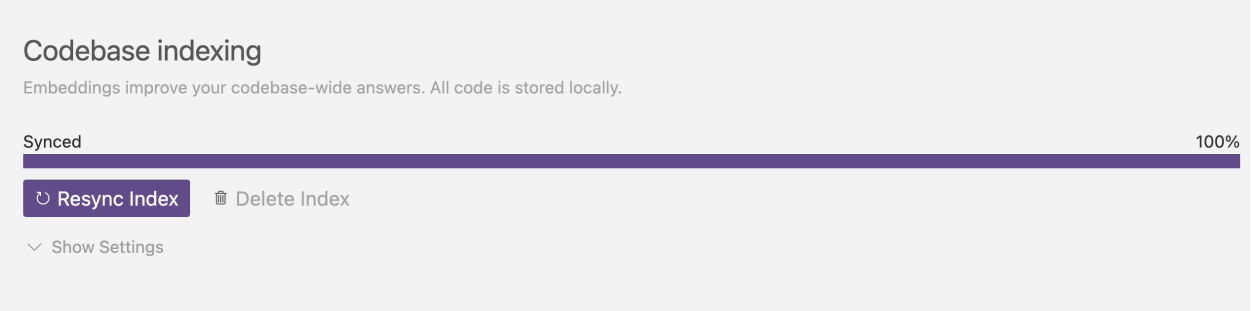
Codebase Indexing（代码库索引功能）

Index your Codebase

为了通过 `@codebase` 或 `Ctrl/⌘ Enter` 来获得更好且更准确的代码库答案，您可以对代码库进行索引。在后台，Cursor 会为您代码库中的每个文件计算 embeddings，以提高代码库答案的准确性。

您的代码库索引将自动与您最新的代码库更改同步。

您可以在 `Cursor Settings` > `Features` > `Codebase Indexing` 处查看代码库索引状态。



Advanced Settings（高级设置）

默认情况下，Cursor 会索引您代码库中的所有文件。

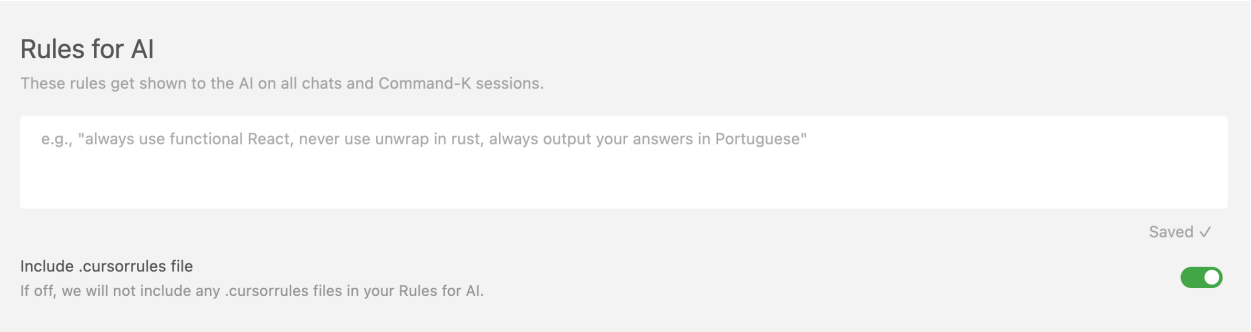
您还可以展开 `Show Settings` 部分以访问更多高级选项。您可以决定是否希望为新代码库启用自动索引，并配置 Cursor 在索引期间忽略的文件，以及您的 .gitignore 设置。

如果您在项目中有某些大型内容文件，AI 无需读取的文件，**忽略这些文件** 可能会提升答案的准确性。

Rules for AI（Cursor系统级提示词）

您可以通过修改 `Rules for AI` 部分来自定义 Cursor 指令，位置在 `Cursor Settings` > `General` > `Rules for AI`。

这些自定义指令将应用于 Cursor Chat 和 Ctrl/⌘ K 等功能。



`.cursorrules`（Cursor规则文件）

针对特定项目的指令，您可以在项目根目录中的 `.cursorrules` 文件中包含指令。

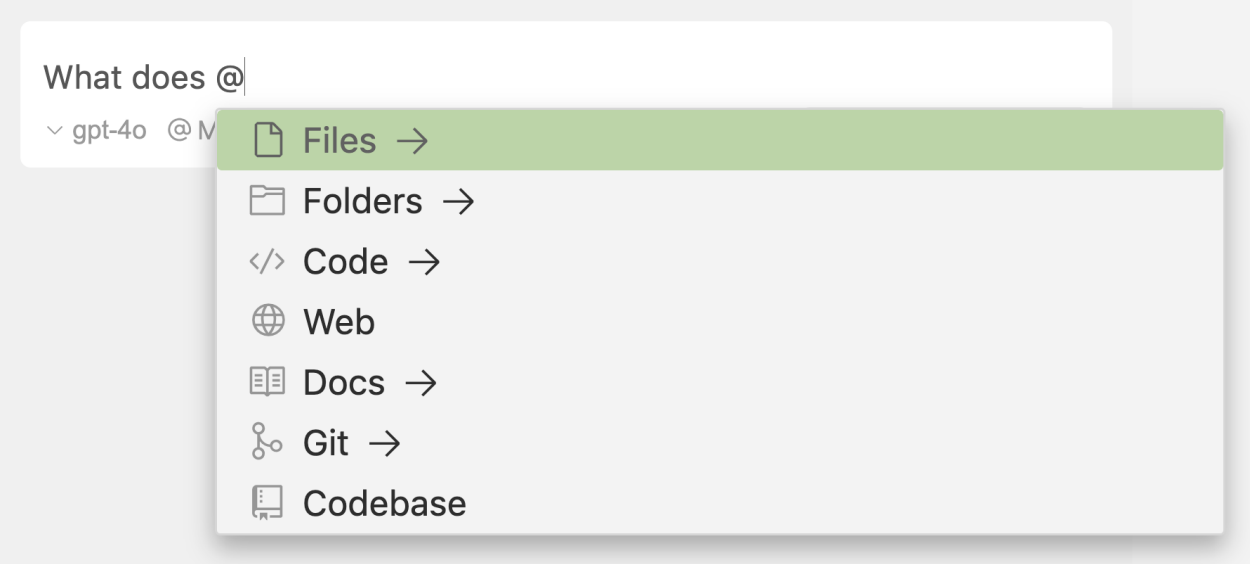
与"Rules for AI"部分相同，`.cursorrules` 文件中的指令将被用于 Cursor Chat 和 Ctrl/⌘ K 等功能。

Basic Usage（基本用法）

在 Cursor 的 AI 输入框中，如 Cmd K、Chat 或 Terminal Cmd K 中，您可以通过输入 `@` 使用 @ 符号。弹出菜单会显示相关建议列表，并自动过滤，仅显示与您的输入最相关的建议。

Keyboard Shortcuts（键盘快捷键）

您可以使用上下箭头键在建议列表中导航，按 `Enter` 选择建议。如果建议是某一类别（例如 `Files`），该类别下的建议将被过滤，仅显示最相关的项目。



Cmd K Keyboard Shortcut（Cmd K 键盘快捷键）

您可以使用上下箭头键在选择的 Cmd K @ 符号列表中导航，按 `Enter` 来扩展或折叠选定的上下文项。对于文件引用，您可使用 `Ctrl/⌘ M` 切换文件读取策略。有关文件读取策略的更多信息，请**点击这里**。

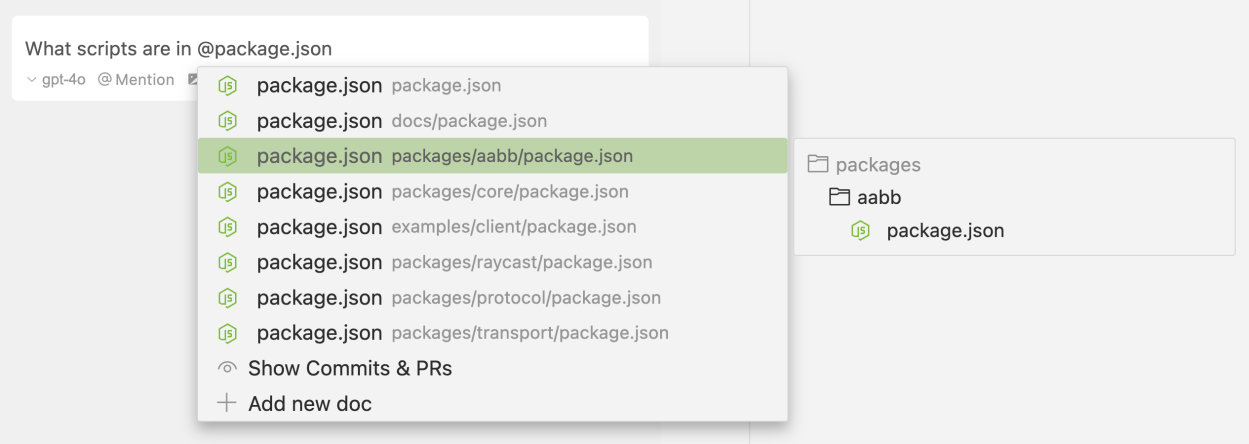
@ Symbols（@符号的用处）

@Files（文件引用）

`@Files`

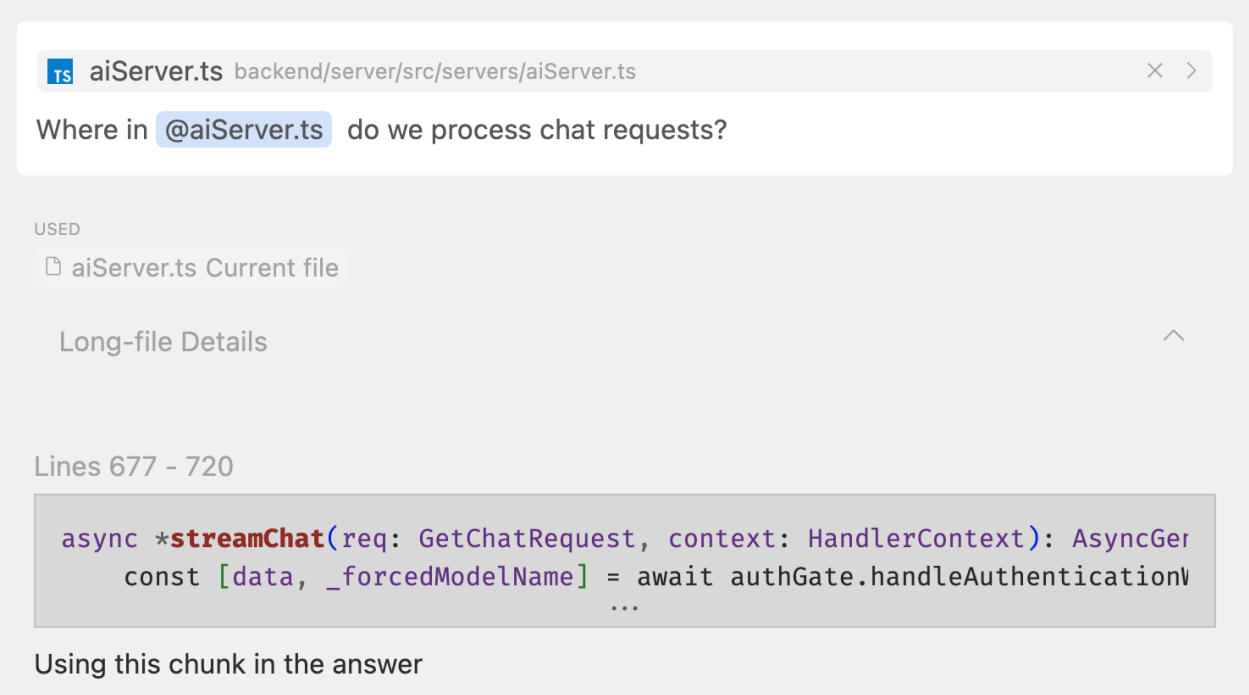
在 Cursor Chat 和 Cmd K 的 AI 输入框中，您可以通过使用 `@Files` 引用整个文件。如果您在 `@` 后继续输入，您将在 `@Code` 策略之后看到文件搜索结果。

为了确保您引用的文件是正确的，Cursor 会显示文件路径的预览。这在您有多个同名文件位于不同文件夹时尤其有用。



Chat Long File References（聊天长文件引用）

在 Cursor 的 Chat 中，如果文件内容过长，Cursor 会将文件分块并根据与查询的相关性重新排序。

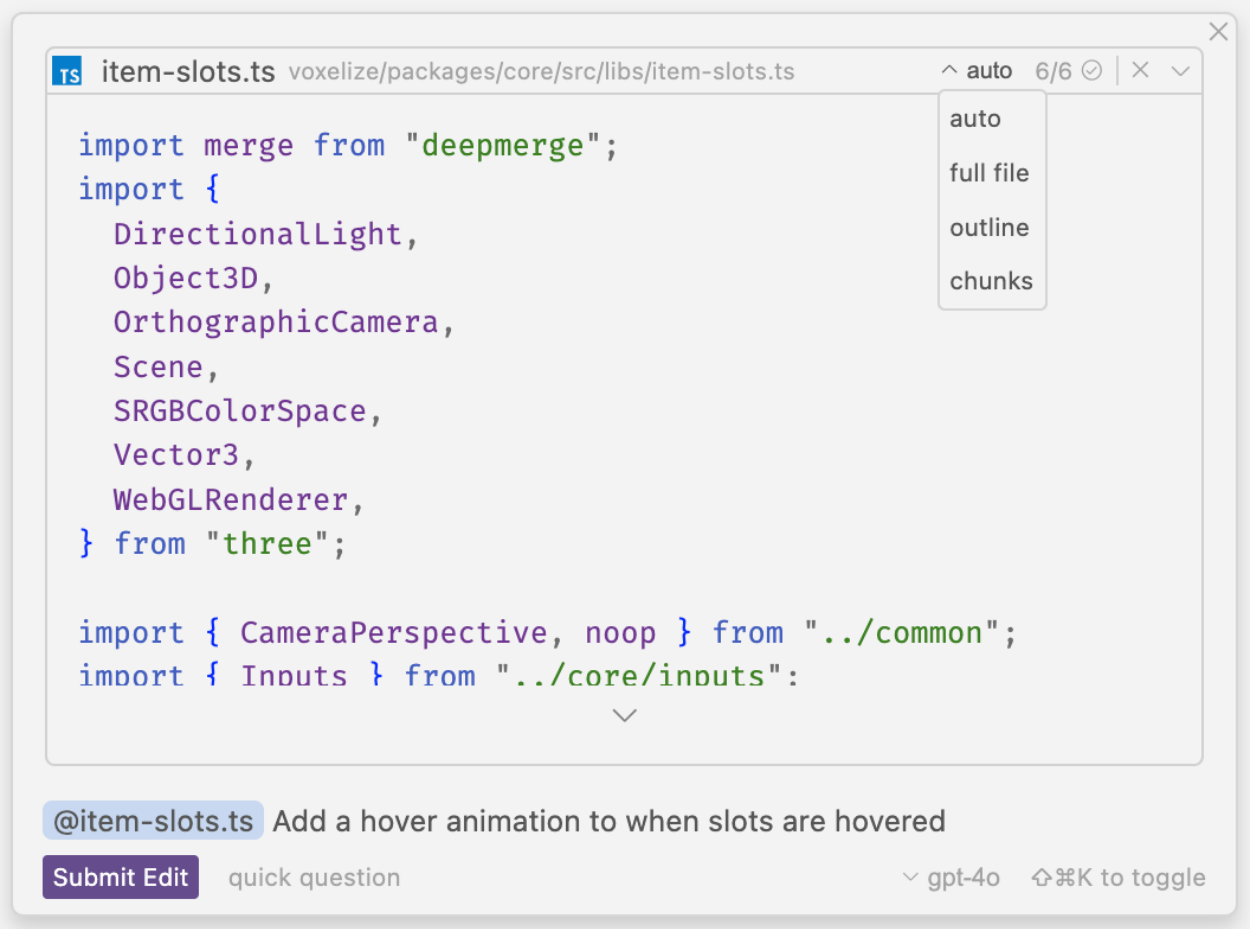


Cmd K Chunking Strategy（Cmd K 分块策略）

对于 Cmd K，Cursor 根据内容长度使用不同的文件引用策略。

- auto
  - 根据文件大小自动选择三种读取策略之一

- full file
  - 整个文件用作上下文。
- outline
  - Cursor 解析文件的轮廓，并将该信息作为上下文。
- chunks
  - Cursor 将文件分块为较小的部分，并选择最相关的部分。



Drag and Drop（拖放功能）

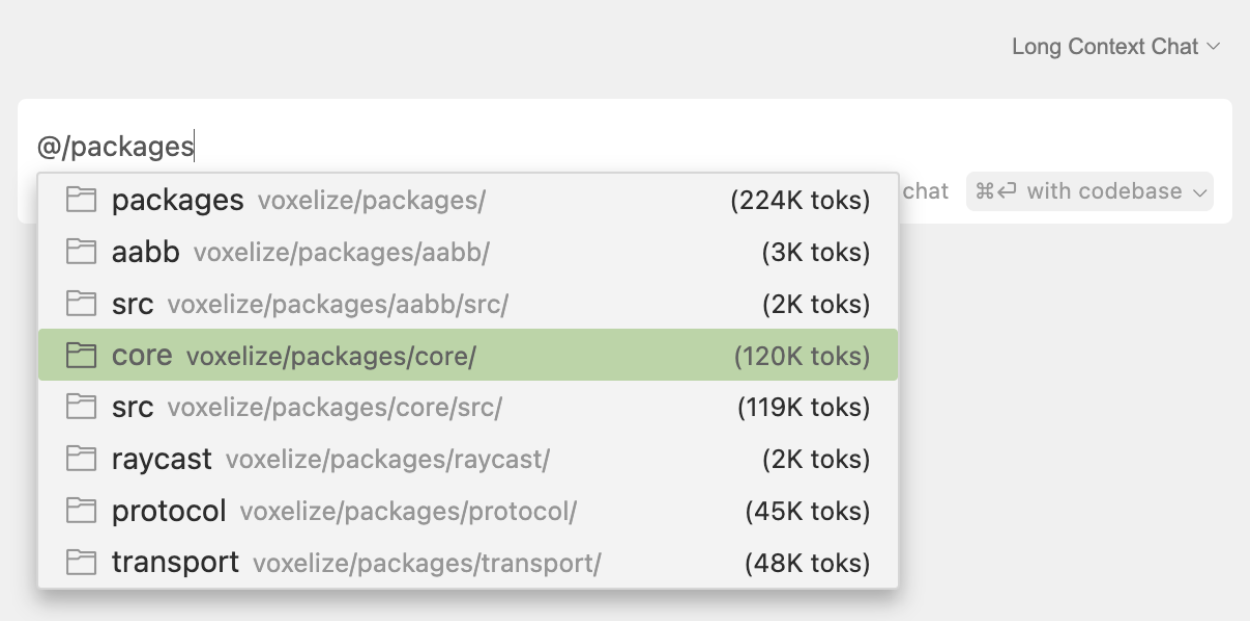
您也可以将文件从主侧边栏拖放到 Chat 或 Cmd K 中，作为上下文添加。

@Folders（文件夹引用）

当前， `@Folders` 仅在 Cursor Chat 中受支持。

`@Folders`

您可以在 Cursor 中引用整个文件夹作为上下文。 `@Folders` 对于 **长上下文聊天** 特别有用，您可以为 AI 提供大量上下文。



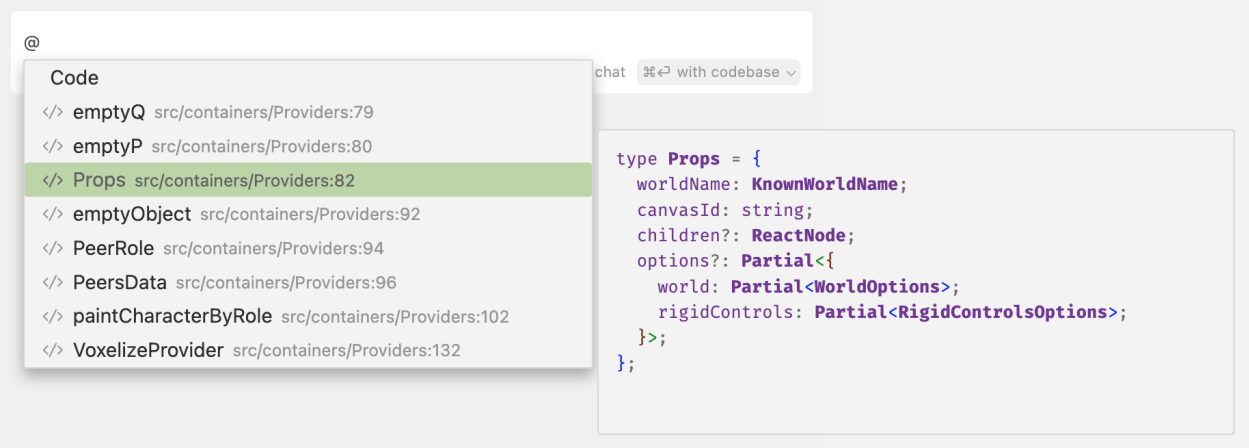
@Code（代码引用）

`@Code`

要引用特定代码段，您可以使用 `@Code` 符号。

Code Preview（代码预览）

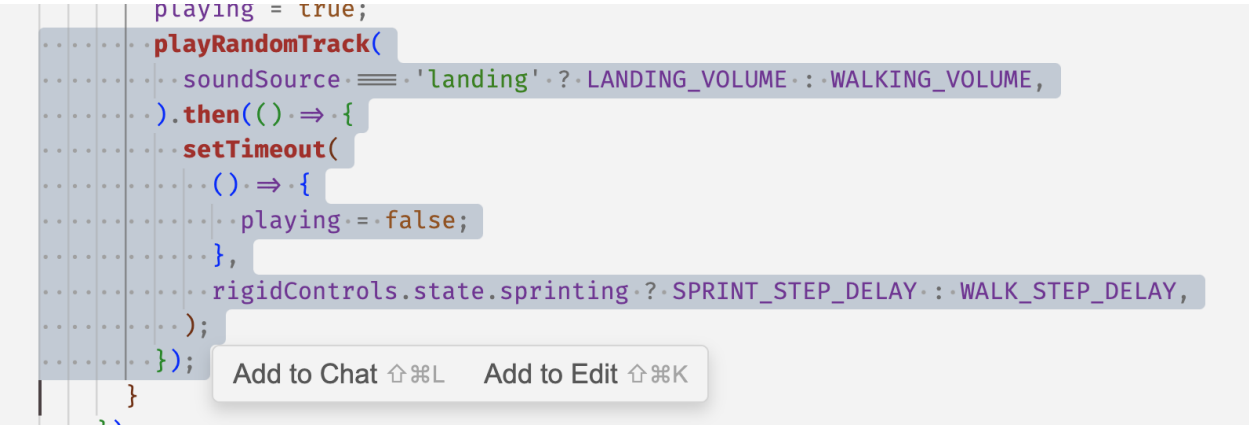
类似于 `@Files` 符号，Cursor 会显示代码内容的预览，以便您确认引用的代码是正确的。



From the Editor（从编辑器中）

要将代码片段添加为上下文，您可以选择要引用的代码，然后点击"添加到聊天"（`Ctrl/⌘ Shift L`）或"添加到编辑"（`Ctrl/⌘ Shift K`）。

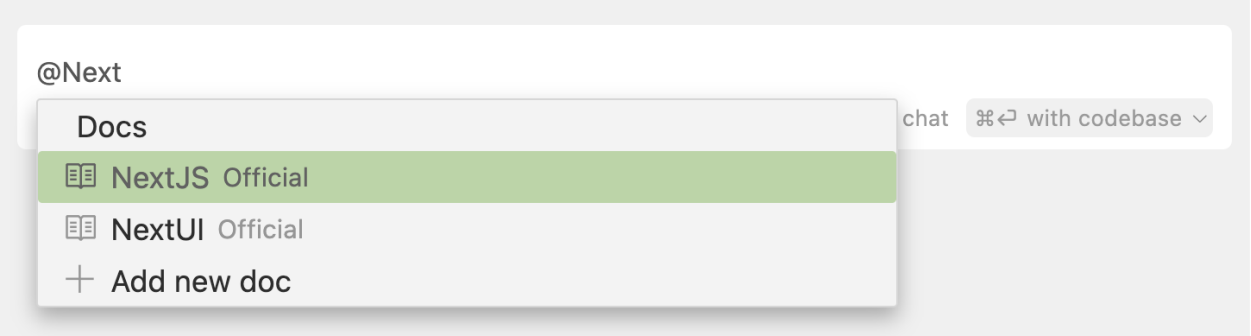
这些操作将选定的代码片段添加到聊天输入框或当前活动的 Cmd K 提示框中。



要将选定的代码添加到新聊天中，您可以按 `Ctrl/⌘ L`。

@Docs（文档引用）

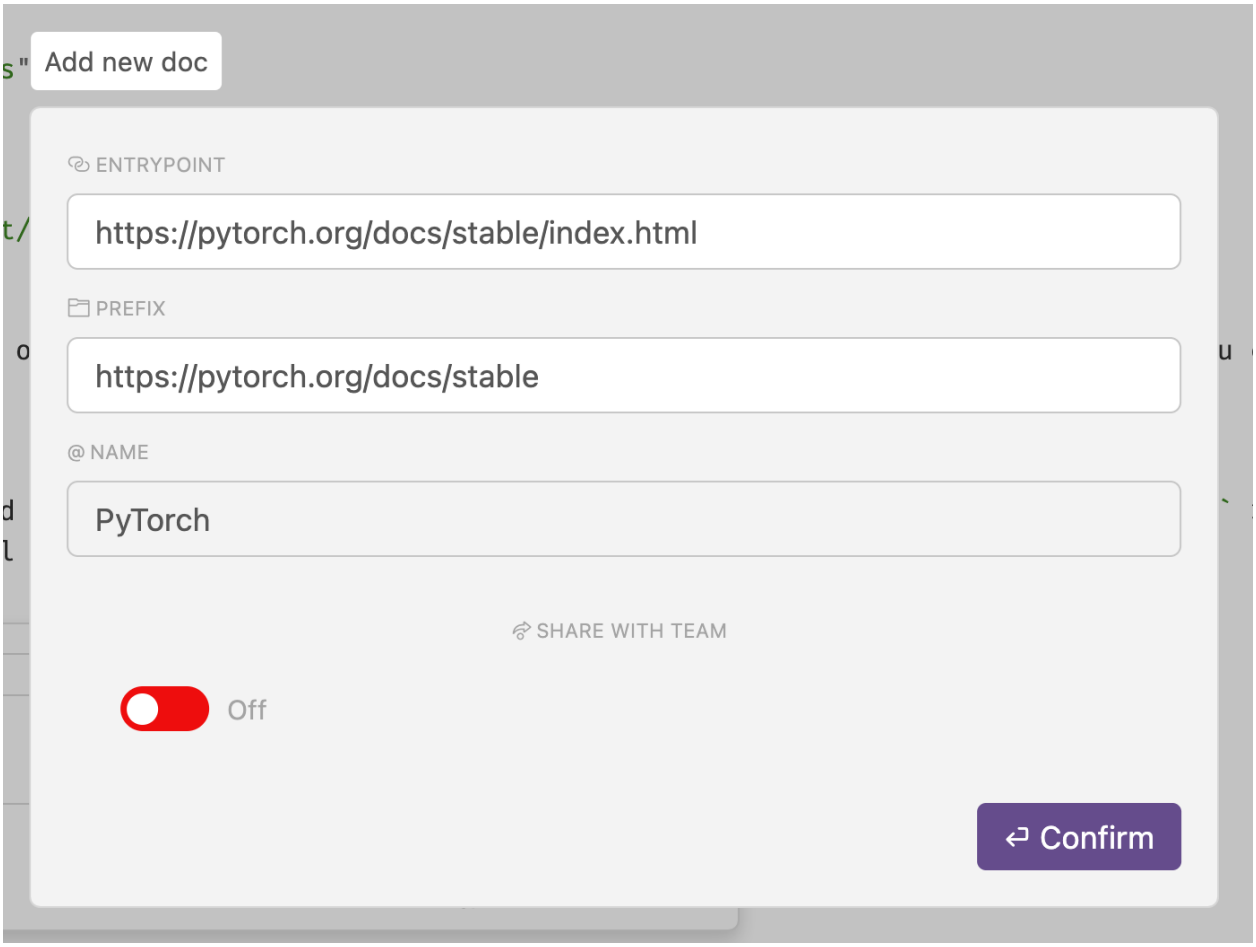
`@Docs`



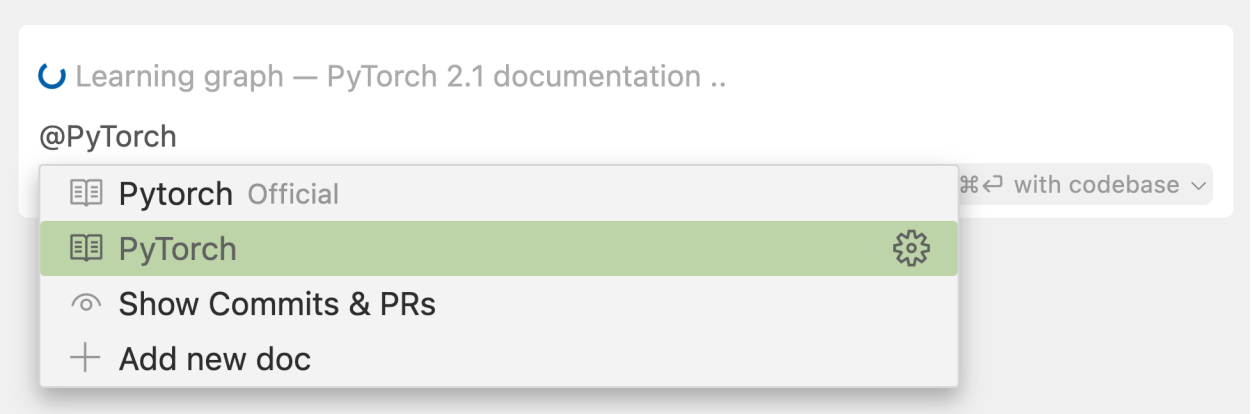
Cursor 提供了一组经过爬取、索引并准备好作为上下文的第三方文档。您可以通过使用 `@Docs` 符号访问这些文档。

Add Custom Docs(添加自定义文档)

如果您希望爬取和索引未提供的自定义文档，可以执行 `@Docs` > `添加新文档`。在您粘贴所需文档的 URL 后，将出现以下模态：

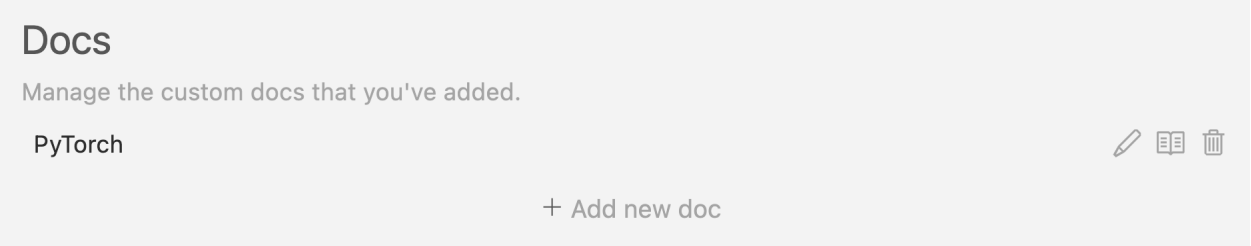


Cursor 会索引并学习该文档，您将能够像使用其他文档一样将其作为上下文。



Manage Custom Docs(管理自定义文档)

在 `Cursor Settings` > `Features` > `Docs` 下，您将看到已添加的文档。您可以在此处编辑、删除或添加新文档。

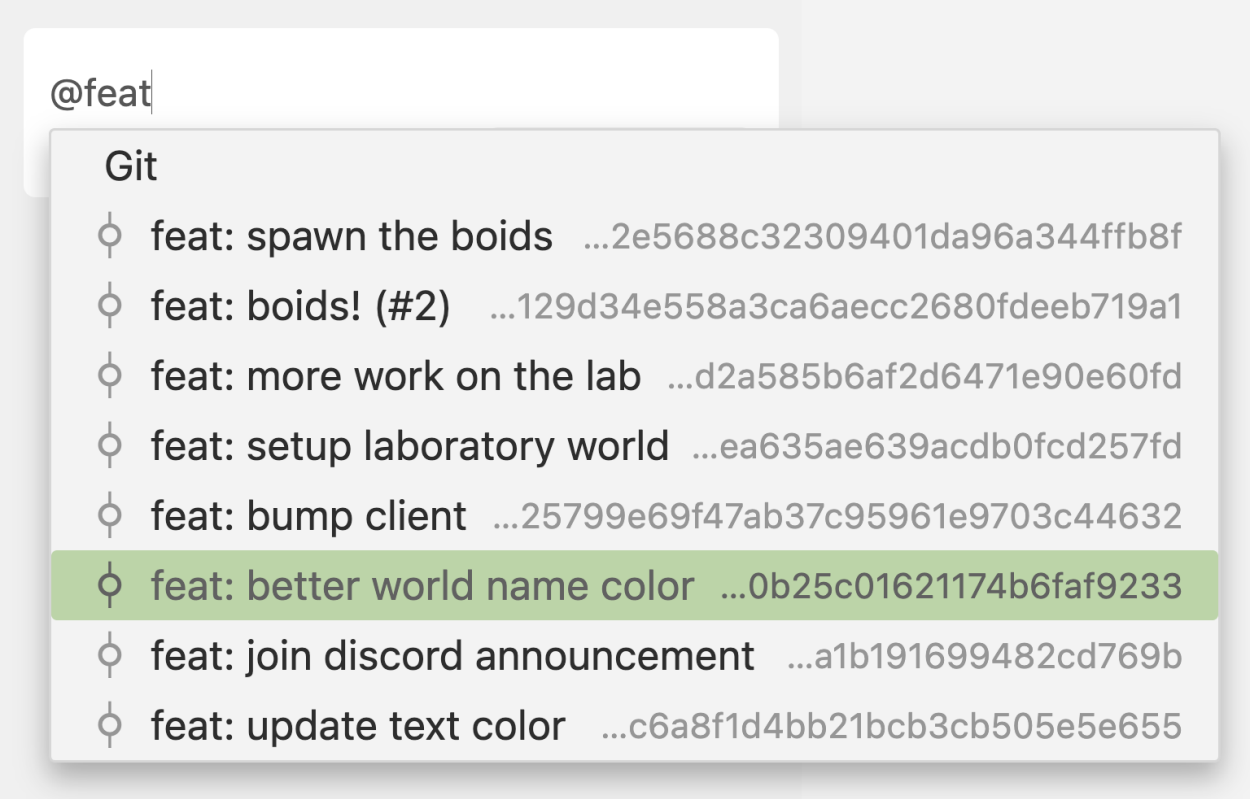


@Git（Git引用）

当前， `@Git` 仅在 Cursor Chat 中支持。

`@Git`

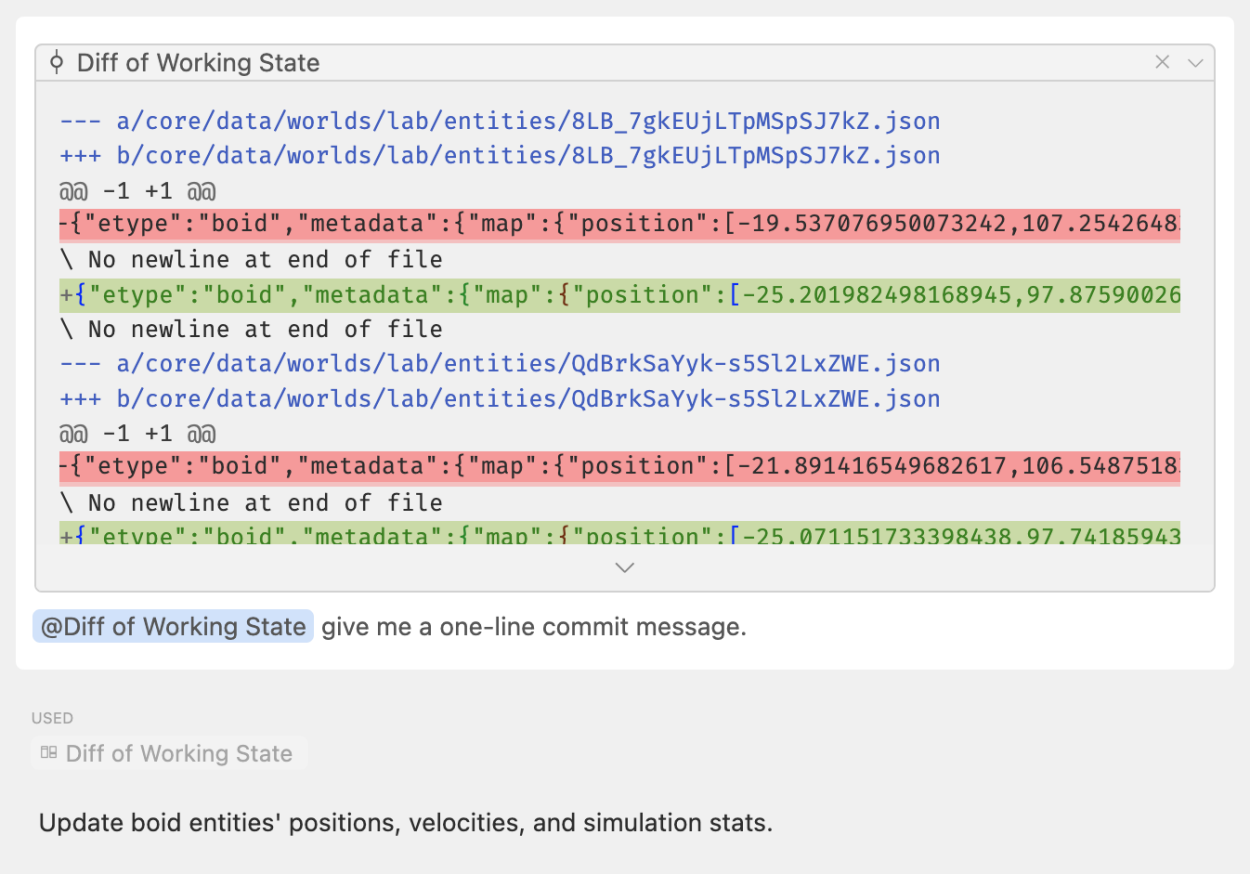
您可以在 Cursor 的 Chat 中使用 `@Git` 将 Git 提交、差异或拉取请求添加到您的提示中。



Common Use Cases（常见用例）

`@Git` 的一个常见用例是允许 Cursor 的 AI 扫描差异，寻找可能由差异导致的 bug 或问题。

您还可以使用 `@Diff of Working State` 根据当前的差异生成提交消息。



@Codebase（代码库引用）

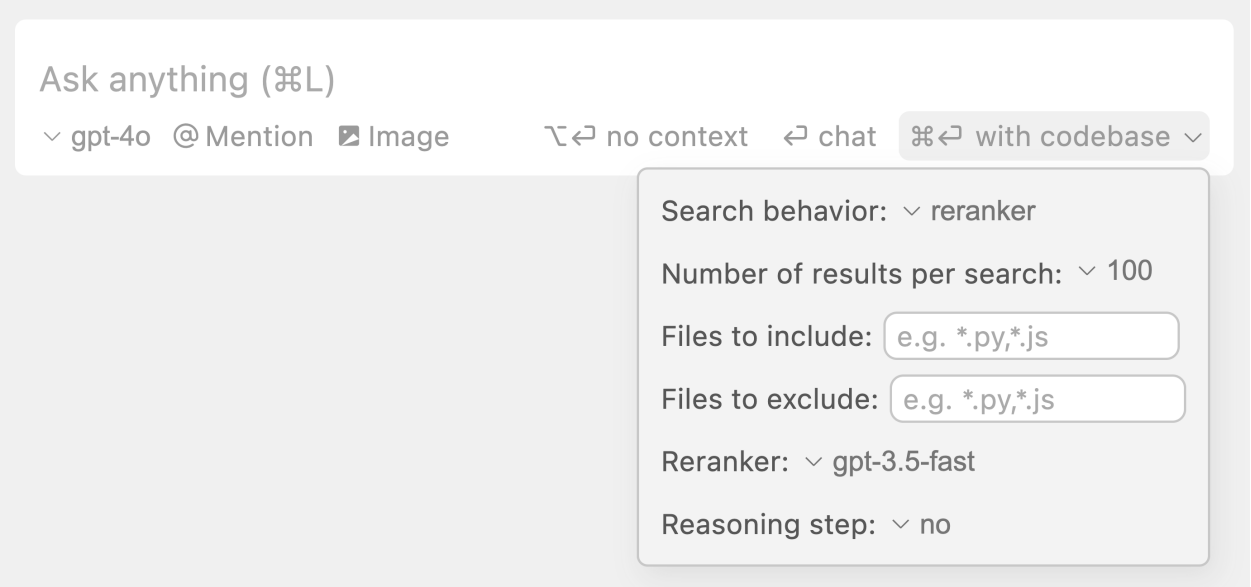
`@Codebase`

通过 `@Codebase`，Cursor Chat 将执行以下步骤以找到相关代码片段：

- Gathering: 扫描您的代码库以寻找重要文件/代码块

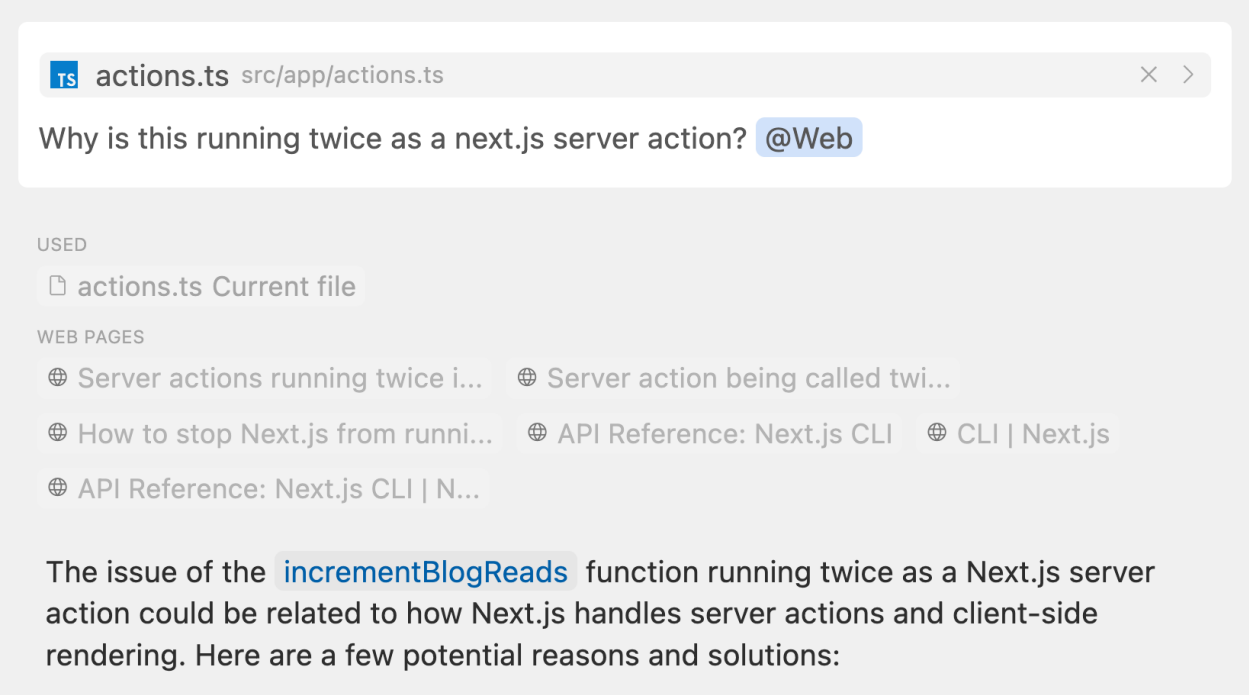
- Reranking: 根据与查询的相关性将上下文项重新排序
- Reasoning: 针对上下文制定使用计划
- Generating: 生成相应响应

另一种提交高级代码库查询的方法是点击 `Ctrl/⌘ + Enter` 按钮旁的下拉菜单，选择 `reranker` 作为搜索行为。该选项仅在不使用 `@Codebase` 时可用，否则 `@Codebase` 将优先。



@Web（网络引用）

`@Web`



通过 `@Web`，Cursor 根据您提供的查询和上下文构建搜索查询，在网络上搜索相关信息作为附加上下文。这对于查找最新信息特别有用。

Always On（始终开启）

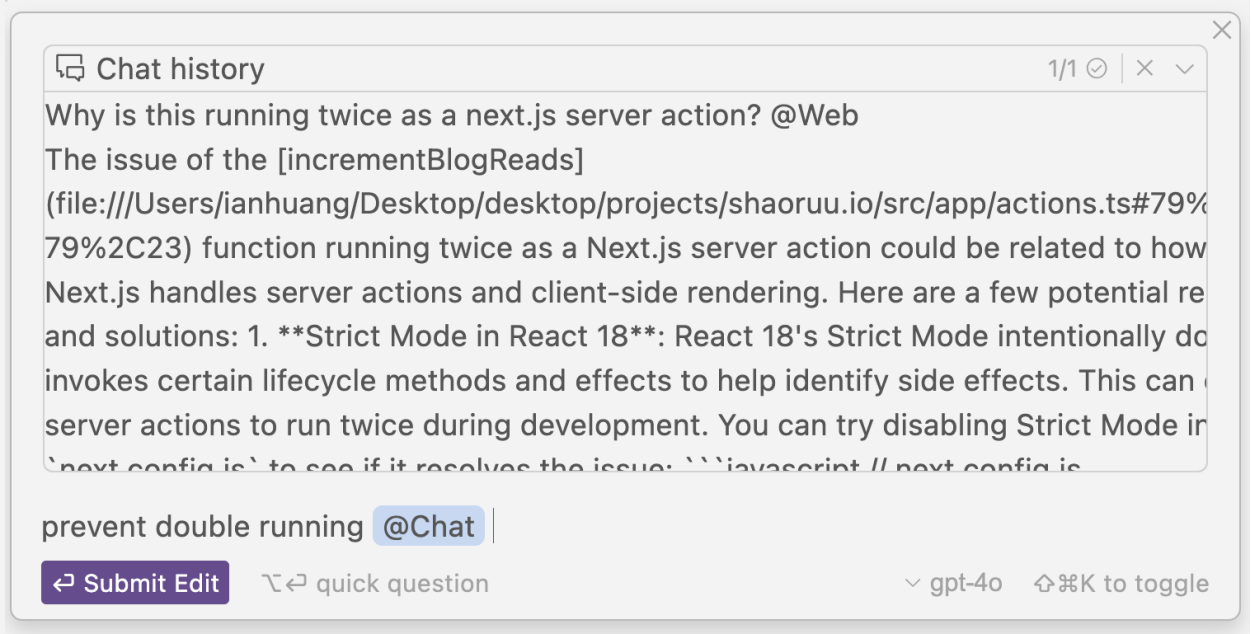
您可以通过在 `Cursor settings` > `Features` > `Chat` 下启用"始终搜索网络"以使 Cursor 在每次查询时自动进行网络搜索。这相当于在每个查询中使用 `@web`。

@Chat（聊天引用）

此功能目前仅适用于 Cmd K。

`@Chat`

您可以通过在 Cmd K 内使用 `@Chat` 将当前聊天消息添加为上下文。当您与 AI 进行对话希望将其应用于编辑或生成代码时，这个功能非常有用。

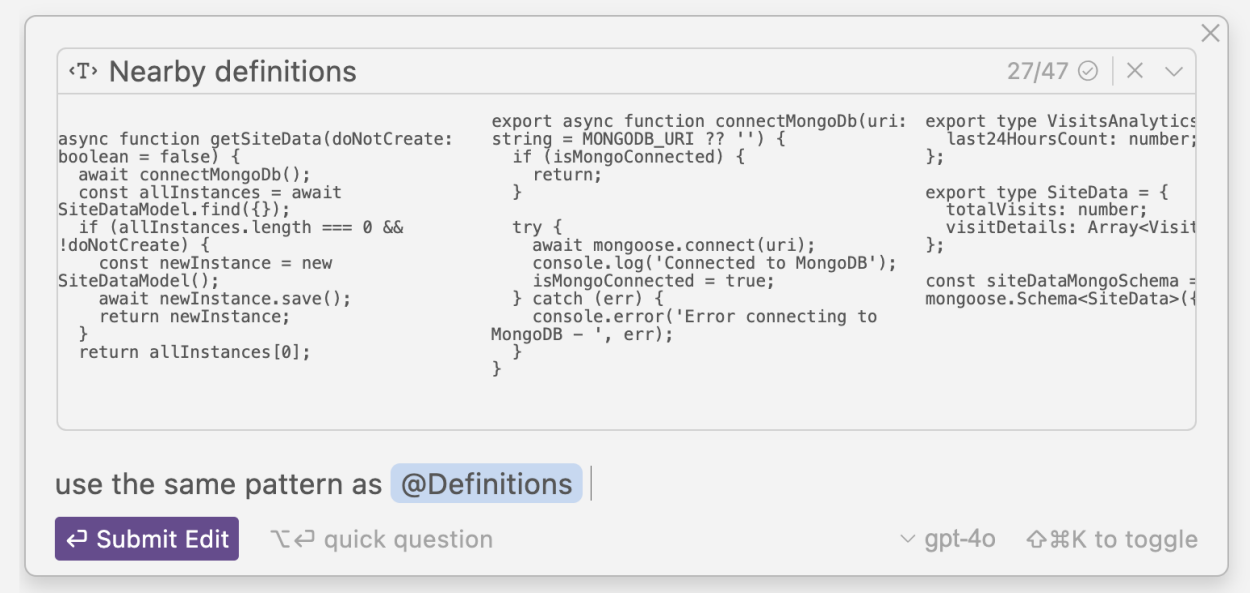


@Definitions（定义引用）

此功能目前仅适用于 Cmd K。

`@Definitions`

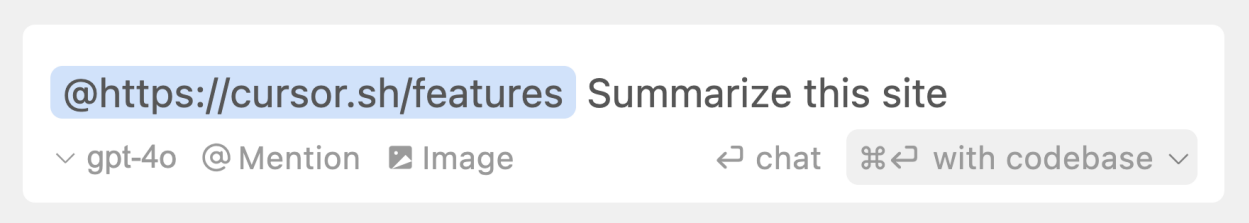
`@Definitions` 符号将所有附近的定义添加到 Cmd K 作为上下文。



Paste Links（粘贴链接）

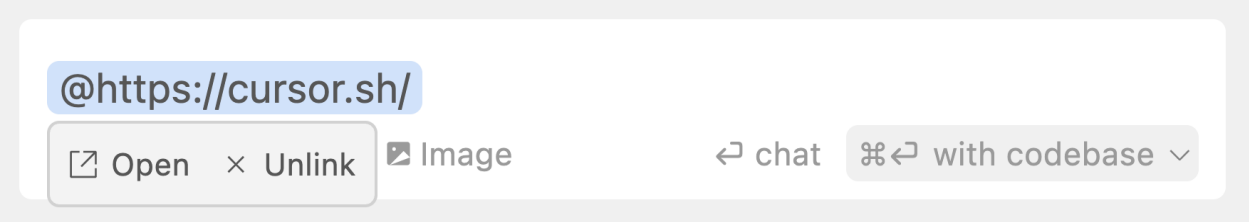
`@https://your-link.com`

为了让 Cursor 在回应之前访问链接，您可以输入 `@` 并粘贴链接。



Remove Links（移除链接）

默认情况下，我们会自动解析链接，将其转换为 `@Links` 在 Cursor Chat 中。如果您希望将链接视为纯文本，可以点击链接，然后单击 `Unlink`。



Ignore Files（忽略文件）

要忽略不希望被包含在 Cursor 功能中的文件（例如代码库索引），您可以在项目根目录下创建一个 `.cursorignore` 文件。它的工作方式与 Git 中的 `.gitignore` 文件完全相同。

`.cursorignore` 文件会遵循 `.gitignore` 的设置。如果您已经有 `.gitignore`，文件将默认被忽略。如果您想忽略其他额外的文件，可以将其添加到 `.cursorignore` 文件中。

示例 `.cursorignore` 文件

忽略特定文件

```
# 忽略dist目录下的所有文件
dist/

# 忽略所有.log文件
*.log
```

```
# 忽略特定文件config.json
config.json
```

仅包含特定文件

仅在`app`目录中包含`*.py`文件。请注意，这与`.gitignore`的语法相同。

```
# 忽略所有文件
*
# 不忽略app
!app/
# 不忽略app内的目录
!app/*/
!app/**/*/
# 不忽略python文件
!*.py
```

故障排除

忽略文件的语法有时会有些混淆。`.cursorignore`文件遵循与`.gitignore`完全相同的语法，因此如果您尝试使用某个忽略文件而没有按预期工作，建议使用Google搜索该问题，将`cursorignore`替换为`gitignore`。可能会有人遇到类似问题，而StackOverflow上会有很好的答案。

一个常见的例子：如何忽略所有文件，除了那些带有`.php`扩展名的文件（仅添加`*`后跟`!*.php`不起作用，因为gitignore文件探测器不会深入并发现子目录中的任何`.php`文件）。

高级

AI模型

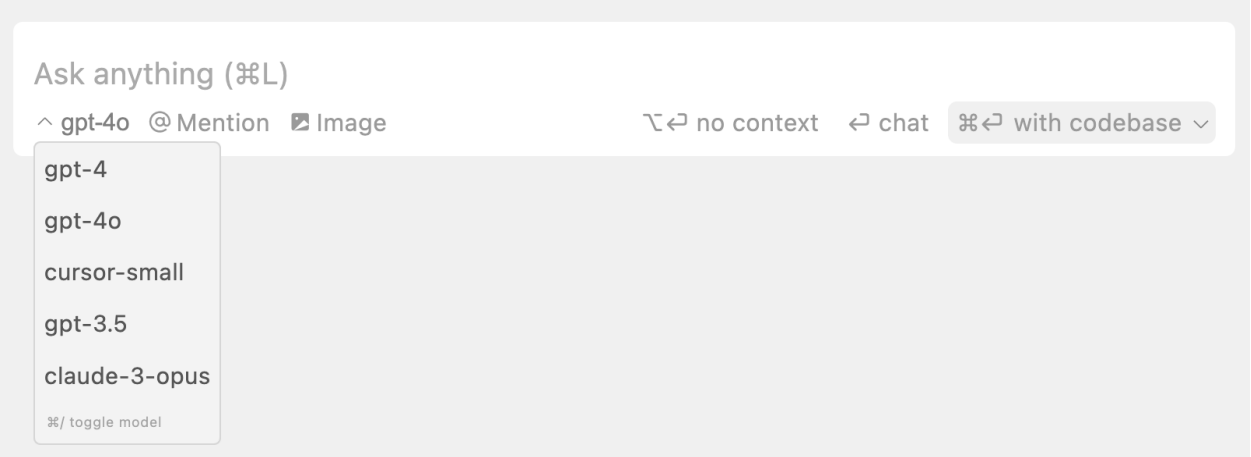
通过Cursor Chat、Ctrl/⌘ K及终端的Ctrl/⌘ K，您可以轻松在不同模型之间切换。

模型下拉菜单

在AI输入框下方，您会看到一个下拉菜单，允许您选择要使用的模型。默认情况下，Cursor准备了以下模型可供使用：

- `GPT-4o`
- `GPT-4`
- `Claude 3.5 Sonnet`
- `cursor-small`
    - `cursor-small`是Cursor的自定义模型，虽然智能程度不如`GPT-4`，但速度更快，用户可以无限制访问。

您可以在`Cursor Settings` > `Models` > `Model Names`下添加其他模型。



仅限长上下文的模型

在长上下文聊天中，模型选择限制为支持长上下文的模型：

- `gpt-4o-128k`
- `gemini-1.5-flash-500k`
- `claude-3-haiku-200k`
- `claude-3-sonnet-200k`
- `claude-3-5-sonnet-200k`

模型X使用什么上下文窗口？

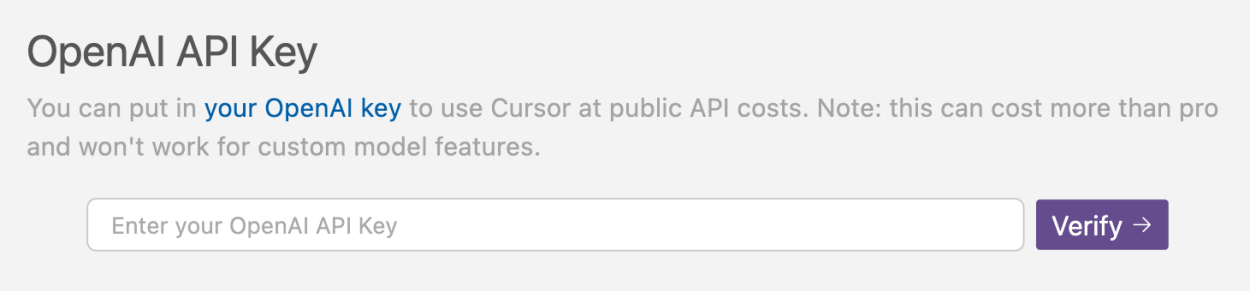
在聊天中，我们当前限制约20,000个tokens（如果模型不支持这么多上下文，则更少）。对于cmd-K，我们限制大约10,000个tokens，以平衡TTFT和质量。长上下文聊天使用模型的最大上下文窗口。

自定义API密钥

OpenAI API密钥

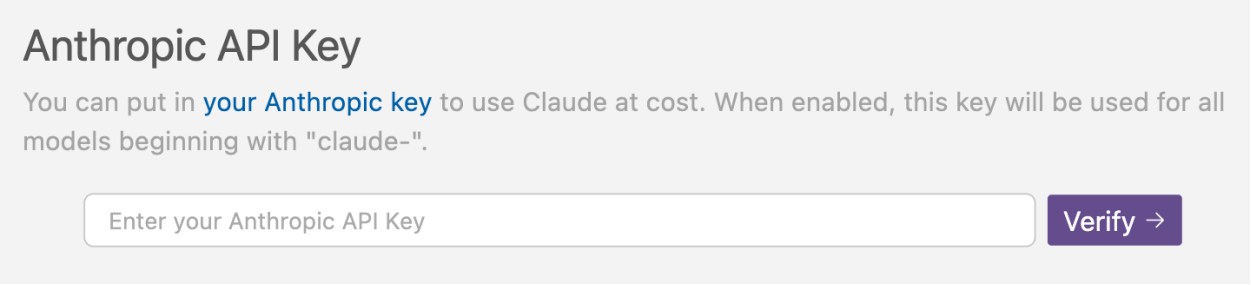
Cursor允许您输入自己的OpenAI API密钥，以便以您自己的成本发送任意数量的AI消息。

您可以从这里获取自己的API密钥。要使用自己的API密钥，请前往`Cursor Settings` > `Models` > `OpenAI API Key`并输入您的API密钥。然后，单击"验证"按钮。一旦您的密钥被验证，您的OpenAI API密钥将被启用。



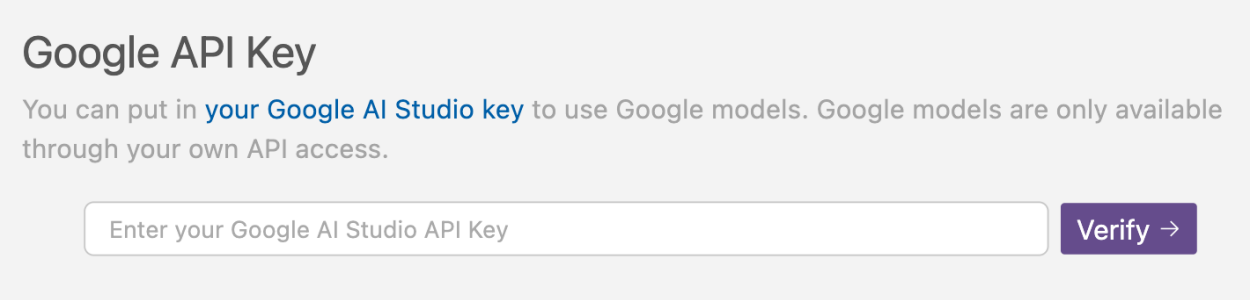
Anthropic API密钥

与OpenAI类似，您还可以设置自己的Anthropic API密钥，以便您以自己的成本使用基于Claude的模型。



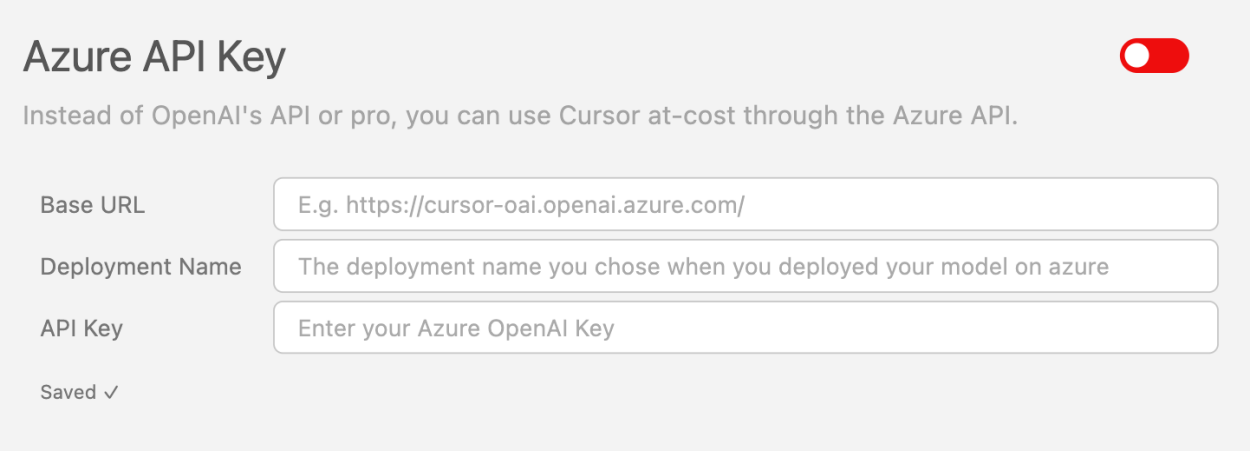
Google API密钥

对于Google API密钥，您可以设置自己的API密钥，以便您以自己的成本使用如`gemini-1.5-flash-500k`等Google模型。



Azure集成

最后，您还可以设置自己的Azure API密钥，以便您以自己的成本使用Azure OpenAI模型。

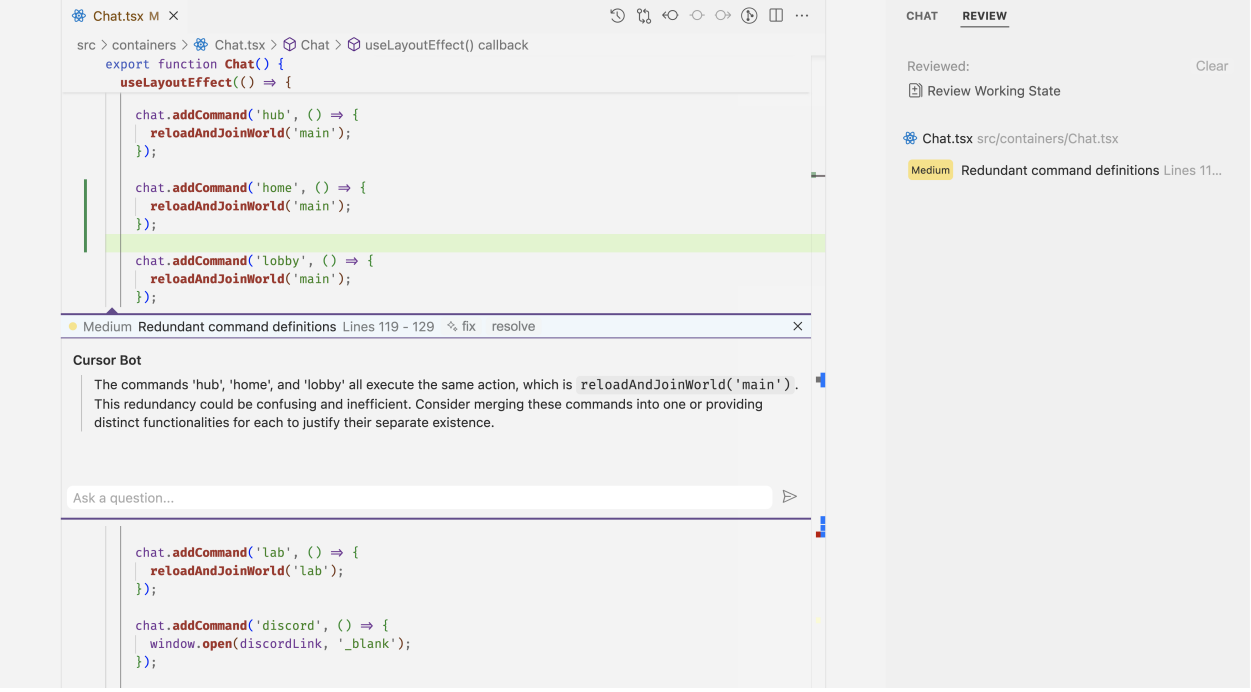


我的API密钥会被存储还是离开我的设备？

您的API密钥不会被存储，但它会随着每个请求发送到我们的服务器。所有请求都通过我们的后端进行路由，因为在那里我们进行最终的提示构建。

AI审查（测试版）

AI审查是一个允许您审查代码库中最近更改以捕获潜在错误的功能。



您可以单击各个审查项目，以在编辑器中查看完整上下文，并与AI进行聊天以获取更多信息。

自定义审查说明

为了使AI审查更有利于您，您可以提供自定义说明，以便AI集中注意力。例如，如果您希望AI关注性能相关的问题，可以输入：

`focus on the performance of my code`

这样，AI审查在扫描您的更改时将专注于代码的性能。

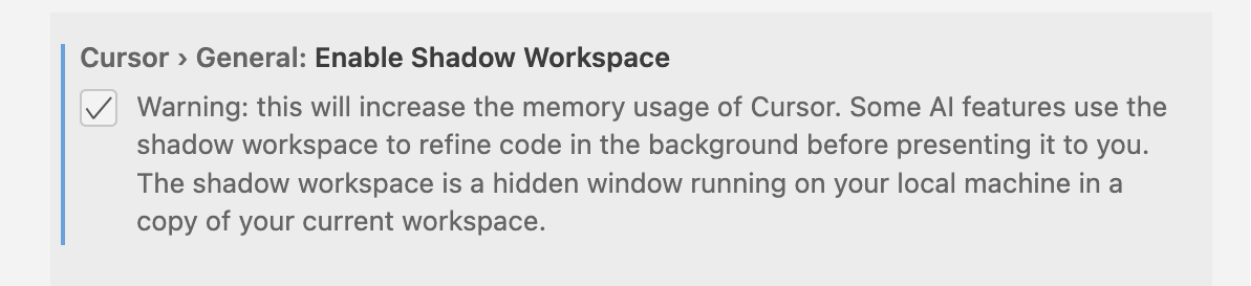
审查选项

目前，您有几个选择供您审查：

- `Review Working State`
  - 这将审查您的未提交更改。
- `Review Diff with Main Branch`
  - 这将审查您当前工作状态与主分支之间的差异。
- `Review Last Commit`
  - 这将审查您最近的提交。

Shadow Workspace（影子工作区）

Shadow Workspace是一个可选设置，您可以配置它以改善AI生成代码的质量。只有部分功能使用它。



如果您启用了Shadow Workspace，后台AI可以请求他们编写代码的lint。这会在您计算机上本地创建一个隐藏窗口，以确保您的编码体验不受影响。

Shadow Workspace将增加Cursor的内存使用量，因此我们建议仅在您有足够的RAM时才启用此功能。

您可以在我们的博客文章中阅读有关Shadow Workspace的更多信息。

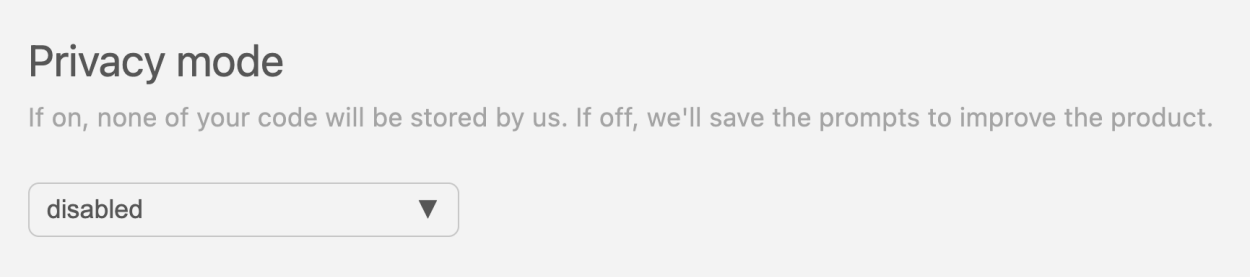
隐私

隐私常见问题

什么是隐私模式？

启用`Privacy Mode`后，您的代码将不会被我们或任何第三方存储。否则，我们可能会收集提示、代码片段和遥测数据，以便改善Cursor。您可以在这里阅读更多关于隐私模式的信息。

您可以在入职时或在`Cursor Settings` > `General` > `Privacy Mode`下启用`Privacy Mode`。



请求是否总是通过Cursor后端路由？

是的！即使您使用API密钥，您的请求仍然会通过我们的后端进行！那里是我们进行最终提示构建的地方。

索引代码库是否需要存储代码？

不需要！如果您选择索引代码库，Cursor将把代码库的小块上传到我们的服务器以计算嵌入，但所有明文代码在请求的生命周期结束后将消失。

嵌入和关于您的代码库的元数据（哈希、模糊文件名）存储在我们的数据库中，但您没有任何代码。

您可以在我们的安全页面上阅读更多关于此的信息。

故障排除

常见问题

我在更新日志中看到更新，但Cursor不会更新。

如果更新非常新，可能尚未向您推出。我们实施分阶段发布，这意味着我们先向少数随机选定的用户发布新更新，然后再发布给所有人。通常，从第一个用户开始，达到所有用户大约需要5小时。对于较大的更新，这可能需要更长时间。

我在Cursor中使用GitHub登录时遇到问题/如何从Cursor中注销GitHub？

您可以尝试使用命令调色板中的`Sign Out of GitHub`命令，快捷键为`Ctrl/⌘ + Shift + P`。

我无法使用GitHub Codespaces。

不幸的是，我们尚不支持GitHub Codespaces。

我在连接Remote SSH时出现错误。

目前，我们不支持SSH连接到Mac或Windows机器。如果您不是在使用Mac或Windows机器，请在论坛中向我们报告您的问题。提供一些日志将有助于获得更好的帮助。

在公司代理后，Cursor Tab和Cmd K不起作用。

Cursor Tab和Cmd K默认使用HTTP/2，这使我们能够以更低的延迟使用更少的资源。有些公司代理（例如某些配置中的Zscaler）会阻止HTTP/2。要解决此问题，您可以在设置中将`"cursor.general.disableHttp2": true`进行设置（`Cmd/Ctrl + ,`，然后搜索`http2`）。

我刚订阅了Pro，但应用中仍然处于免费计划。

请尝试在Cursor设置弹出窗口中注销然后重新登录。

我的使用量何时再次重置？

如果您订阅了Pro，可以在仪表板中单击`Manage Subscription`，计划续订日期将在顶部显示。

如果您是免费用户，可以查看您收到了我们第一封邮件的日期。您的使用量将于每个月从该日期重置。

如何卸载Cursor？

您可以遵循此指南卸载Cursor。将每次出现的“VS Code”或“Code”替换为“Cursor”，并将“.vscode”替换为“.cursor”。

故障排除指南

以下是对大多数Cursor故障排除场景有帮助的信息列表：

01) 问题截图（隐去任何敏感信息）

02) 重现步骤

03) 系统信息来自：

`Cursor` > `Help` > `About`

04) 您是否在使用VPN或Zscaler？

05) 开发者工具控制台错误

通过以下方式打开开发者工具：

`Cursor` > `Help` > `Toggle Developer Tools`

然后点击`Console`，看看是否有相关的错误。

06) 日志

在Windows中，您可以在以下位置找到日志：

`C:\Users\<your-user-name>\AppData\Roaming\Cursor\logs`

这是执行以下操作时打开的文件夹的父文件夹：

- `Ctrl` + `Shift` + `P`（在Cursor中打开命令调色板）
- 输入并选择`Developer: Open Logs Folder`

您还可以在`Cursor` > `Terminal` > `Output`中查看日志，然后点击下拉菜单选择`Window`或其他Cursor特定选项，例如`Cursor Tab`或`Cursor Indexing & Retrieval`。

译者：AI进化论-花生（Powered by ChatGPT&Claude）

B站：https://space.bilibili.com/14097567

YouTube：https://www.youtube.com/@Alchain