# 官方文档

## 国内版本

https://docs.trae.com.cn/docs/troubleshooting

## 国际版本

https://docs.trae.ai/ide/what-is-trae

# 快捷键

## ctrl+,

打开设置

## alt+右箭头

前进

## alt+左箭头

返回

## F12

转到定义

## ctrl+P

打开文件导航

## Ctrl + Shift + F

打开全局搜索

## ctrl+U

与AI对话

## ctrl+I

Editor内AI编码

## ctrl+shift+P

打开命令面板

## ctrl+shift+`

新建终端

## ctrl+N

新建文本文件

## alt+Z

自动换行

## ctrl+/

切换行注释

# 强大的AI助手

## 1、智能问答

在编写代码时，你可以随时与 AI 助手对话，获得代码解释、代码注释和错误修复等帮助。

## 2、实时代码建议

AI 助手能够理解当前代码并在编辑器中实时提供代码建议，提升编程效率。

## 3、代码片段生成

通过自然语言描述你的需求，AI 助手将生成相应的代码片段，甚至能够编写项目级或跨文件的代码。

## 4、从 0 到 1 开发项目

告诉 AI 助手你想开发的程序，AI 助手将根据你的描述提供相关代码或自动创建所需文件。

# 管理项目

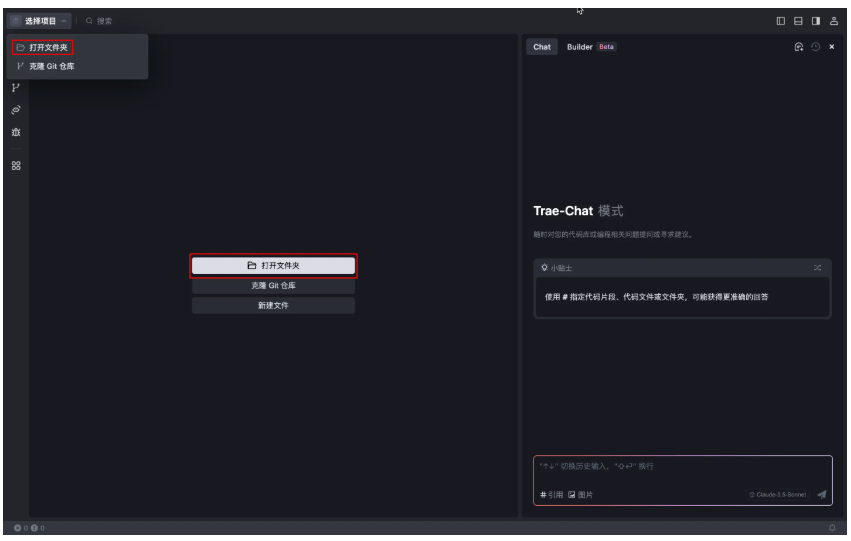
## 工作空间

“工作空间” 通常为一个在 Trae 中打开的文件夹。若你的项目较为复杂，也可以将多个文件夹添加至一个 “工作空间”。

## 创建项目

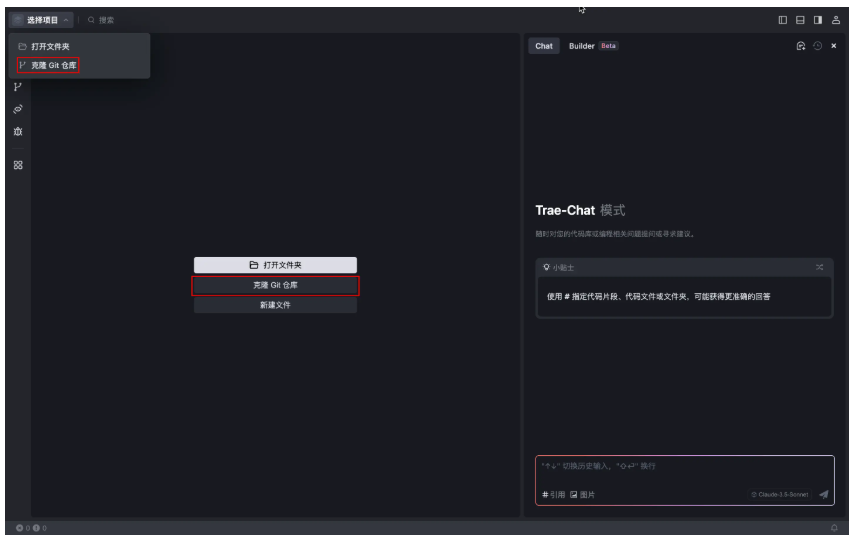
### 方式一：导入本地文件夹

点击左侧面板中央的 打开文件夹 按钮，或在界面左上角点击 选择项目 > 打开文件夹

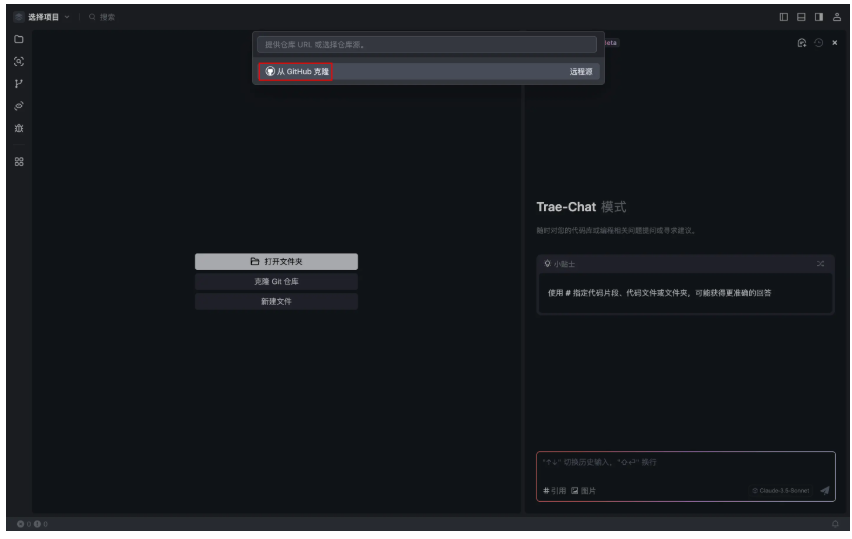


### 方式二：从 GitHub 克隆仓库

点击左侧面板中央的 **克隆 Git 仓库** 按钮，或在界面左上角点击 **选择项目** > **克隆 Git 仓库**



点击 从 GitHub 克隆

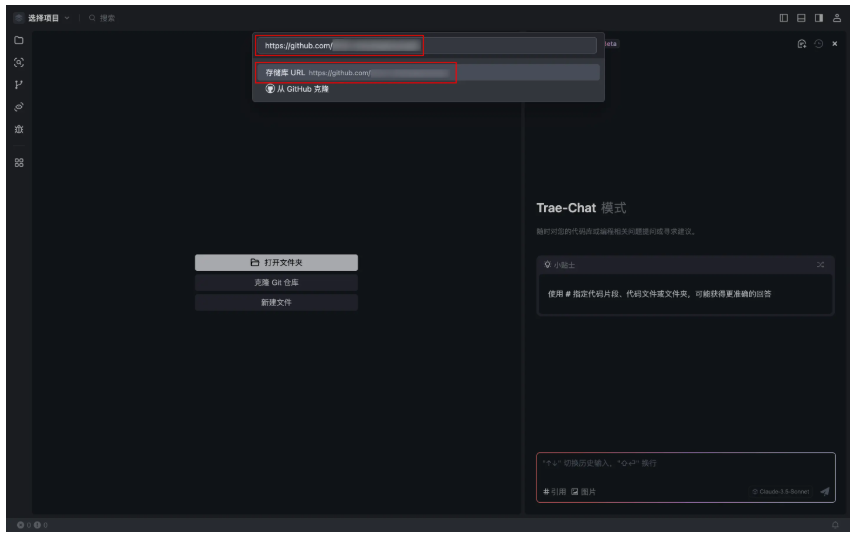


### 方式三：直接从 Git 仓库的 URL 克隆

此方式无需授权 GitHub 登录 Trae，你可以直接使用目标仓库的 URL 来将其克隆至 Trae。

1、点击左侧面板中央的 克隆 Git 仓库 按钮，或在界面左上角点击 选择项目 >克隆 Git 仓库。

2、在 URL 输入框中，输入目标 Git 仓库的 URL，然后点击 存储库 URL {URL}。



# 管理插件

## 从 Trae 的插件市场安装

## 从 VS Code 的插件市场安装

## 禁用插件

## 卸载插件

# 对话

## 侧边对话

## 行内对话（Inline chat）

### 唤起行内对话

你可以通过以下两种方式唤起行内对话：

在编辑器内的光标处，使用快捷键（macOS：Command + I；Windows：Ctrl + I）。

在编辑器内，选中任意代码，然后使用快捷键（macOS：Command + I；Windows：Ctrl + I）或点击悬浮菜单中的 **编辑** 按钮。

### 开始问答

在行内对话输入框中输入你的需求，包括为代码添加注释、解释选中的代码、优化选中的代码等等，然后点击右侧的 发送 按钮或敲击回车键。

### 处理结果

### 重试对话

# 模型

# 智能体

## 基本概念

智能体（Agent）是你面向不同开发场景的编程助手。Trae IDE 提供内置智能体。同时，你还可以创建自定义智能体，通过灵活配置提示词和工具集，使其更高效地帮你完成复杂任务。

## 能力

### 1、自主运行

独立探索你的代码库，识别相关文件并进行必要修改。

### 2、完整的工具访问权限

使用所有可用工具进行搜索、编辑、创建文件及运行终端命令。

### 3、上下文理解

建立对你项目的结构和依赖关系的全面理解。

### 4、多步骤规划

将复杂任务拆分为可执行的步骤，并按顺序逐一处理。

## 智能体的工作流

### 1、需求分析

深入理解任务目标及代码库上下文，明确需求要点。

### 2、代码调研

检索代码库、文档及网络资源，定位相关文件并分析现有实现逻辑。

### 3、方案设计

根据分析结果拆解任务步骤，并动态优化调整修改策略。

### 4、实施变更

照计划在整个代码库中进行必要的代码变更，过程中可能涉及：

（1）新增依赖库推荐

（2）需执行的终端指令

（3）Trae IDE 客户端外的手动操作指引

### 5、交付验收

完成验证后移交控制权，同步汇总所有修改内容。

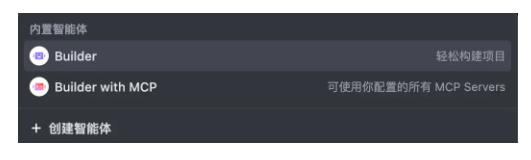
## 内置智能体

### Builder

Builder 可以帮助你从 0 到 1 开发一个完整的项目。根据你的需求，Builder 会调用不同的工具，包括分析代码文件的工具、编辑代码文件的工具、运行命令的工具等等，从而更加精确且有效地处理你的需求。

### Builder with MCP

在Builder的基础上，你配置的所有 MCP Server 都会默认添加至 Builder with MCP，且不可编辑。



# 多模态输入

## 基本概念

你可以在会话中添加图片，例如报错截图、设计稿、参考样式等等，从而更加准确高效地表达需求。

# 代码自动补全

## 自动补全代码

## 根据代码注释补全代码

# 上下文（重点）

## 概览

与 AI 助手对话时，你可以指定上下文供 AI 助手阅读和理解，使 AI 助手的答复更精准。

### 将编辑器内的内容作为上下文

### 将终端中的内容作为上下文

### 使用 # 符号添加上下文

### 管理代码索引

Trae AI 可以为项目构建索引。构建索引后，用户通过 #Workspace 或 #Folder 方式与 AI 助手对话时，AI 助手能够理解用户当前打开的项目并使用整个项目作为上下文进行回答，从而提供更加精准的回答。

### 忽略文件

Trae IDE 默认忽略 .gitignore 文件中所罗列的文件。除此之外， 你可以自主设置 Trae IDE 在索引仓库时需要额外忽略的文件列表，以作为 .gitignore 文件中已定义规则的补充。

## 代码索引

## 规则

### 基本概念

通过制定规则来规范 AI 在 Trae IDE 内的行为。

### 规则类型



### 创建个人规则

### 创建项目规则

### 管理规则

## #符号

### #Code

### #File

### #Folder

### #Workspace

### #Doc

#### 基本概念

### #Web

## 忽略文件

## MCP

### 基本概念

Model Context Protocol (MCP) 是一种协议，它允许大型语言模型（LLMs）访问自定义的工具和服务。Trae 中的智能体作为 MCP 客户端可以选择向 MCP Server 发起请求，以使用它们提供的工具。你可以自行添加 MCP Server，并添加到自定义的智能体中来使用。

在 Trae IDE 中，MCP Server 支持两种传输类型：stdio 传输、SSE 传输。这两者的区别如下：

stdio传输主要用于本地进程与AI模型的交互，它基于操作系统进程间通信的机制，主要应用于本地环境。

SSE传输主要用于远程通信，它是一种基于HTTP的服务器推送技术，允许服务器通过长连接主动向客户端发送数据流。

对于这两种传输类型，我们了解一下就好了，不是开发一个MCP Server，无需太过深入，整个配置过程中，对于传输类型都是无感的。

### 配置系统环境

为确保正常启动 MCP Server，你可能需要安装：

npx：依赖于 Node.js，版本需大于等于18。

uvx：基于 Python 的快速执行工具，需手动安装。

(可选) Docker：容器化平台，用于隔离和运行应用程序，需根据系统版本安装对应版本。若使用 GitHub MCP Server，则需要使用 Docker。

### 安装Node.js

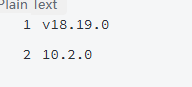
1、下载并安装 Node.js 18 或更高版本。

2、安装完成后，在终端中运行以下命令确认是否安装成功

node -v

npx -v

若安装成功，终端中会输出已安装的 Node.js 的版本号，例如：



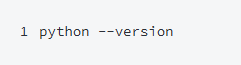
3、重启 Trae IDE，使Node.js生效。

### 安装uvx（Python 工具）

uvx 是 [uv](https://github.com/astral-sh/uv) 提供的命令行工具，用于快速运行 Python 脚本。

1、下载并安装 Python 3.8 或更高版本。

2、安装完成后，在终端中执行以下命令确认是否安装成功。



若安装成功，终端中会输出已安装的 Python 的版本号。

3、执行以下命令，安装 uv（包含 uvx）。安装脚本会自动下载 uv 并放置到系统默认路径中。

Windows 安装命令（PowerShell）

powershell -ExecutionPolicy ByPass -c "irm https://astral.sh/uv/install.ps1 | iex"

4、执行以下命令，验证是否安装成功



若安装成功，终端中会输出已安装的 uvx 的版本号。

### (可选) 安装 Docker

若使用 GitHub MCP Server，则需要使用 Docker。使用以下步骤安装 Docker Desktop。



使用以下操作验证是否安装成功

1、打开终端，输入docker --version命令。如果安装成功，会显示 Docker 的版本信息。

2、输入 docker info 命令，可以查看 Docker 的详细信息。如果能正确显示信息，代表 Docker 已安装并启动成功。

### 配置MCP Server

你可以直接从市场中添加合适的 MCP Server，或手动配置。

### 常用的MCP Server

#### 1、Figma AI Bridge

##### 功能

将Figma设计稿自动转换为前端代码，实现设计与开发的快速衔接

##### 配置步骤

1、在Figma个人设置中生成Personal Access Token

2、在Trae IDE的MCP配置页面添加Figma AI Bridge

3、粘贴Token完成配置

##### 使用场景

示例：通过自然语言指令生成前端代码

"请严格按照我提供的Figma链接内容生成HTML前端页面。UI要严格还原设计稿，需要实现响应式设计"。

##### 特点

1、支持React + Tailwind等多种技术栈

2、可批量更新设计变更

3、生成语义化、响应式代码

#### 2、GitHub MCP

##### 基本概念

GitHub MCP Server是Trae IDE中一个强大的集成工具，它通过Model Context Protocol(MCP)实现了AI智能体与GitHub平台的深度连接，使开发者能够通过自然语言指令直接操作GitHub仓库、管理Issues和处理Pull Requests等开发工作流。

##### 核心功能

###### 1、****仓库管理自动化****

#### 3、Apifox MCP

##### 基本概念

Apifox MCP Server，可以将 Apifox 的接口文档提供给 Cursor 等支持 AI 编程的 IDE，或其他支持 MCP 的 AI 工具。

##### 如何使用

完成配置后，Apifox MCP Server 会自动获取接口文档的数据并缓存在本地，AI 助手可以通过 MCP 读取接口文档数据。

你只要告诉 AI 你想要通过 API 文档做什么即可，示例：

1、"通过 MCP 获取 API 文档，然后生成 Product 及其相关模型的定义代码"

2、"根据 API 文档，在 Product DTO 里添加 API 文档新增的几个字段"

3、"根据 API 文档给 Product 类的每个字段都加上注释"

4、"根据 API 文档，生成接口 /users 相关的所有 MVC 代码"

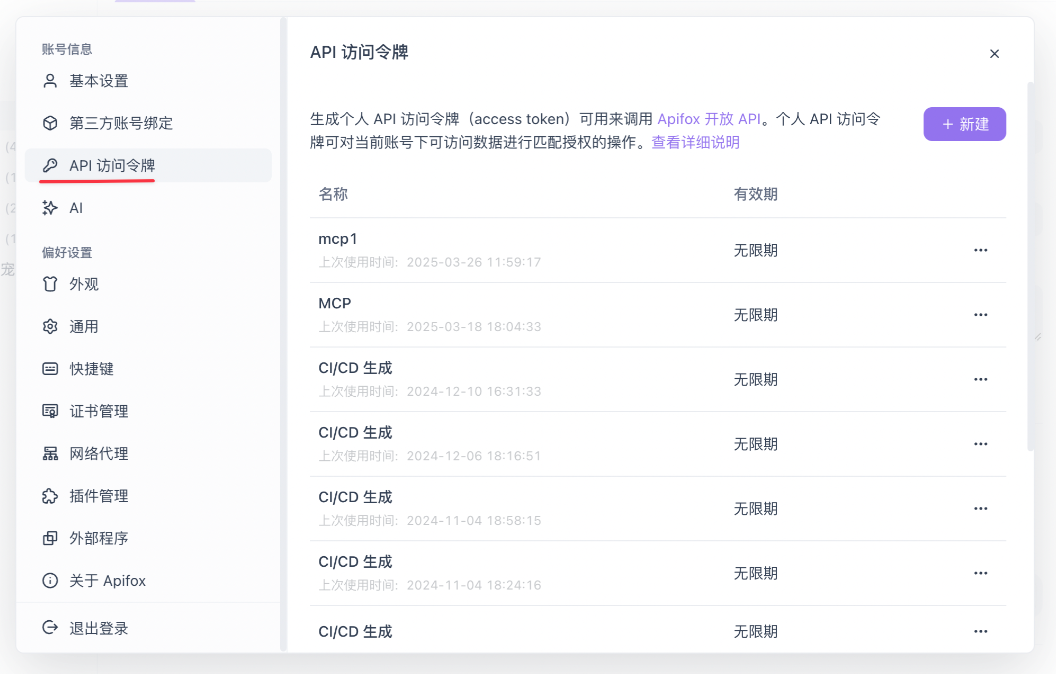
注意：接口文档数据默认缓存在本地，如果 Apifox 内的数据有更新，请告诉 AI 刷新接口文档数据，否则 AI 读到的数据可能不是最新的。

##### 配置方法

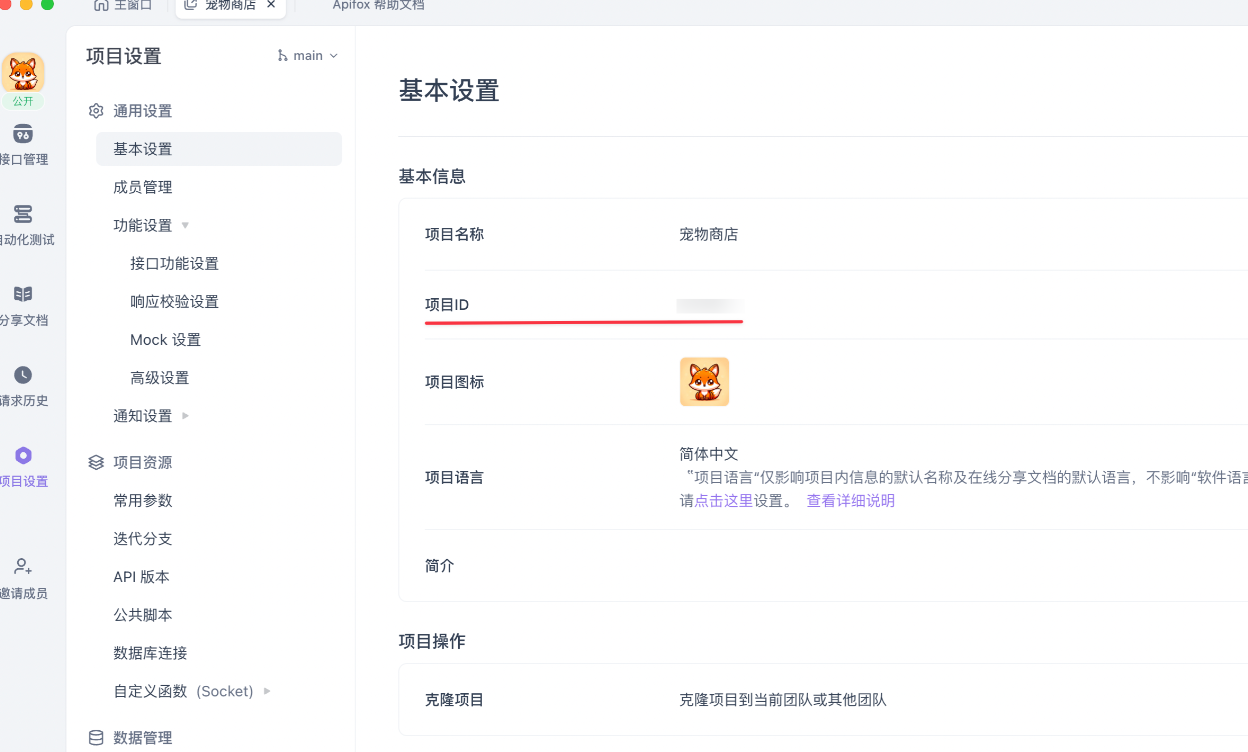
###### 1、前提条件

已安装 Node.js 环境（版本号 >= 18，推荐最新的 LTS 版本）

###### 2、在 Apifox 生成 API 个人访问令牌



###### 3、获取 Apifox 项目 ID



###### 4、添加配置内容

在打开的 mcp.json 文件中添加以下配置，注意替换 <access-token> 和 <project-id> 为你自己的值：



###### 5、验证配置

配置完成后，你可以通过向 AI 询问以验证连接是否正常工作（智能体模式），例如：





#### 4、PostgreSQL

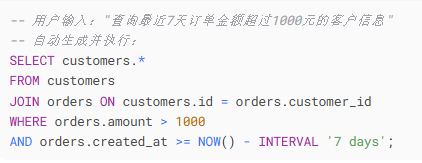
##### 基本概念

PostgreSQL MCP Server是Trae IDE中一个专门用于连接和操作PostgreSQL数据库的Model Context Protocol服务，它通过标准化的协议将AI能力与PostgreSQL数据库深度集成，极大提升了数据库开发和管理效率。

##### 功能

###### 1、自然语言转SQL查询

将开发者的自然语言描述自动转换为优化的SQL语句并执行：



###### 2、数据库结构管理

**表结构查看**："显示users表的结构"

**索引管理**："为orders表的customer\_id字段添加索引"

**外键关系**："列出所有引用products表的外键关系"

###### 3、数据操作自动化



###### 4、性能优化



##### 配置方法

{

"mcpServers": {

"PostgreSQL": {

"command": "npx",

"args": [

"-y",

"@modelcontextprotocol/server-postgres",

"postgresql://postgres:123456@localhost:5432/translate"

],

"env": {}

}

}

}

##### 注意事项

1、如果在数据库中新增了表，那么需要删除MCP配置，重新添加，否则查询不到新添加的表中的数据。

# 源代码管理

# IDE设置

# Trae插件

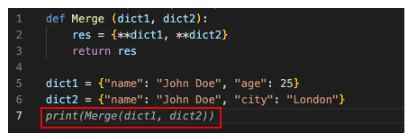
## 基本概念

## AI功能

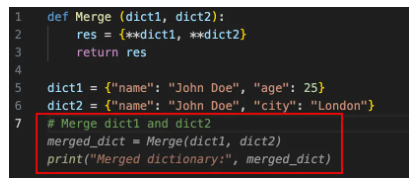
### 代码补全

Trae AI 可以理解当前代码，然后提供建议补全的代码。

**场景一**：在对应代码处，敲击回车键换行。Trae AI 会阅读并理解当前代码，然后自动补全后续代码。



**场景二**：在对应位置添加注释，然后敲击回车键换行。Trae AI 会阅读并理解代码注释，然后直接生成代码。



将鼠标悬浮至自动补全的代码区域，然后采纳或拒清除推荐的代码：

### 代码补全 Pro（Beta）

# Trae中使用MCP和智能体的窍门

https://mp.weixin.qq.com/s/UymLh65FkUumhAR-9DYUwg

# 使用技巧

https://mp.weixin.qq.com/s/RdT8zPFdbFjb0VoFt4CvYQ

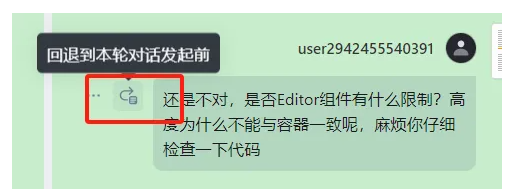
## 1、开启文件路径导航

## 2、回退到之前的某个版本

有时候AI改着改着，你突然发现不对劲，想回退到之前的某个版本，这时分两种情况。

### 清楚的知道该回退到哪个版本

这种情况直接在build模式下，找到你想回退的那个提问，点击提示词左侧的撤销按钮，如下：



撤销以后，所有的改动就都回退到此次改动之前了。

### 不知道回退到哪个版本，需要对比修改内容

如果是不知道回退到哪个版本才是对的，此时可以查看时间线，如下：



时间线会显示你当前打开的文件的历史所有修改，如果你想回退某次修改，右键然后点击还原内容就可以了。

## 3、自动安装vscode官方插件市场插件

## 4、释放Trae占用的C盘空间