

# COMATE 文档



#### 【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

#### 【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

#### 【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

## 目录

目录	2
产品介绍	4
核心能力	4
支持语言、IDE与操作系统	4
快速开始	5
Comate AI IDE 快速入门	5
VSCode快速入门	10
JetBrains快速入门	11
Visual Studio快速入门	13
Xcode快速入门	14
Eclipse快速入门	16
功能介绍	21
概述	21
智能体	26
概述	26
工具	26
消息压缩	27
Checkpoints	27
并行会话	28
模式	28
自定义智能体	29
上下文	39
Skills	39
Rules	49
Memory	55
MCP	56
@关联上下文	76
智能问答	80
代码补全	85
行间命令	90
Mission Mode	94
AI IDE	97
Spec模式	97
预览和元素级调试	101
手动安装插件	105
开放平台	108
企业控制台	113
登录	113
主页	114
权限管理	116

组织管理	116
用户管理	121
知识管理	122
数据统计	130
开放中心	142
企业设置	150
付费订阅	154
企业版	156
开通企业版	156
授权管理	158
企业成员使用	159
混合云帮助手册	161
代码安全	164
解决方案	166
数据安全	170
产品定价	171
常见问题	173
服务协议和隐私协议	177
Baidu Comate用户服务协议	177
Baidu Comate隐私协议	183
Baidu Comate自动续费服务协议	191
发版记录	193
Comate AI IDE 更新日志	193
VSCode端插件更新日志	198
JetBrians端插件更新日志	214
VS端插件更新日志	236
Xcode端插件更新日志	249
Comate 4.0版本正式发布！	249
01 全面重构推理与执行能力	250
02 全面重构 Subagent、MCP	250
03 全面支持 Skills (Tag : New)	251
04 全新 Explore Subagent	251
05 全新 Slash	251
06 速度更快、Token更省	252
07 全新视觉：让复杂变简单	252

# 产品介绍

文心快码 (Baidu Comate) 是百度公司倾力打造的一款创新性 AI 编程助手。它深度融合了先进的人工智能技术，为开发者提供贯穿软件开发全流程的智能编码辅助，旨在创造更懂你的下一代编程体验。无论您是编程初学者还是经验丰富的开发者，文心快码都能成为您值得信赖的编程伙伴。

您可以通过在 [VSCode](#)、[JetBrains IDEs](#) 等主流 IDE 中安装 Comate 插件，或直接使用功能强大的独立 [AI IDE](#) 来体验文心快码。

## 核心能力

### 智能体

文心快码内置强大的编程智能体，能够理解复杂的编程任务，并与开发者协同完成。

**多智能体协同**：能够自动将复杂任务拆解，并唤起多个分工明确的子智能体协同处理。通过隔离不同任务的上下文，高效完成更大规模、更复杂的软件工程任务。

**自定义智能体**：为特定技术领域或业务场景提供深度优化，弥补通用模型在专业领域的不足。同时支持用户按需定制，满足个性化的编程需求。

**领域知识与工具集成**：通过 **自定义规则 (Rules)** 与 **MCP** 配置，可轻松接入企业或个人的领域知识库及第三方工具，增强智能体的专业能力。

### 代码补全

**代码补全**：在您编码时，能够精准理解代码上下文，实时预测并生成从单行到整个代码块的补全内容。

**超级代码补全**：能够智能预测您下一步可能需要编写代码的位置，并提供代码改写建议。

**行间命令**：在代码编辑区内提供丰富的快捷指令，支持一键完成代码解释、添加函数注释、生成单元测试等高频操作。

**Inline Chat (行内对话)**：通过快捷键在代码编辑区唤起对话框，使用自然语言即可对当前选定的代码进行重构、优化或调试。

### 知识增强

文心快码通过自研的上下文引擎，在代码补全与对话中能自动检索并融合当前代码库与相关技术文档知识。这使得生成的代码不仅语法正确，而且更贴合项目规范与开发要求。

### 模型

文心快码采用百度自研的文心代码大模型作为核心引擎，同时灵活支持接入 DeepSeek、ChatGLM、Kimi 等业界优秀的开源模型，以满足不同场景下的性能与成本需求。

### AI IDE

除了作为插件使用，文心快码还提供独立的 AI IDE。它深度整合了代码编辑、智能问答和终端功能，实现了无缝联动，让开发体验更加连贯。此外，IDE 还内置了 [前端实时预览](#)、[元素级调试](#) 等特色功能。

### 企业特性

针对企业级应用场景，文心快码推出了完善的企业版解决方案。支持 SaaS、混合云及私有化等多种部署模式。可与企业的统一认证和用户系统无缝对接，提供更强的数据安全保障。内置研发效能度量工具，帮助企业管理者洞察团队开发效率。开放丰富的自定义功能，满足企业特定的开发规范与业务需求。

## 支持语言、IDE与操作系统

支持 Java、Python、Go、C#、C/C++、JavaScript、TypeScript、PHP、Ruby、Rust、Scala、Kotlin 等主流编程语言。

JetBrains IDEs

- IDE 版本：IntelliJ IDEA、PyCharm、GoLand、WebStorm、Android Studio 等 2021.1 及以上。
- 操作系统：Windows 7 及以上、macOS、Linux。

#### Visual Studio Code

- IDE 版本：1.70 及以上。
- 操作系统：Windows 7 及以上、macOS、Linux。

#### Visual Studio

- IDE 版本：Visual Studio 2019 与 2022 版本。
- 操作系统：Windows 10 及以上。

#### Comate AI IDE

- 操作系统：Windows 10/11 (x64/arm64)，macOS 11.0 或更新版本，Linux glibc >=2.28，glibcxx >=3.4.25(如 Ubuntu 20+，Fedora 36+)。

#### 其他场景

- 远程开发: 完整支持 Remote SSH、Docker、WSL 等远程开发环境。
- Web IDE: 支持在 VS Code WebIDE 中使用，并可通过 Open VSX 插件市场下载。
- HUAWEI DevEco Studio: 兼容 JetBrains IDEs 的配置要求。

[下载：文心快码Comate](#)

## 快速开始

### Comate AI IDE 快速入门

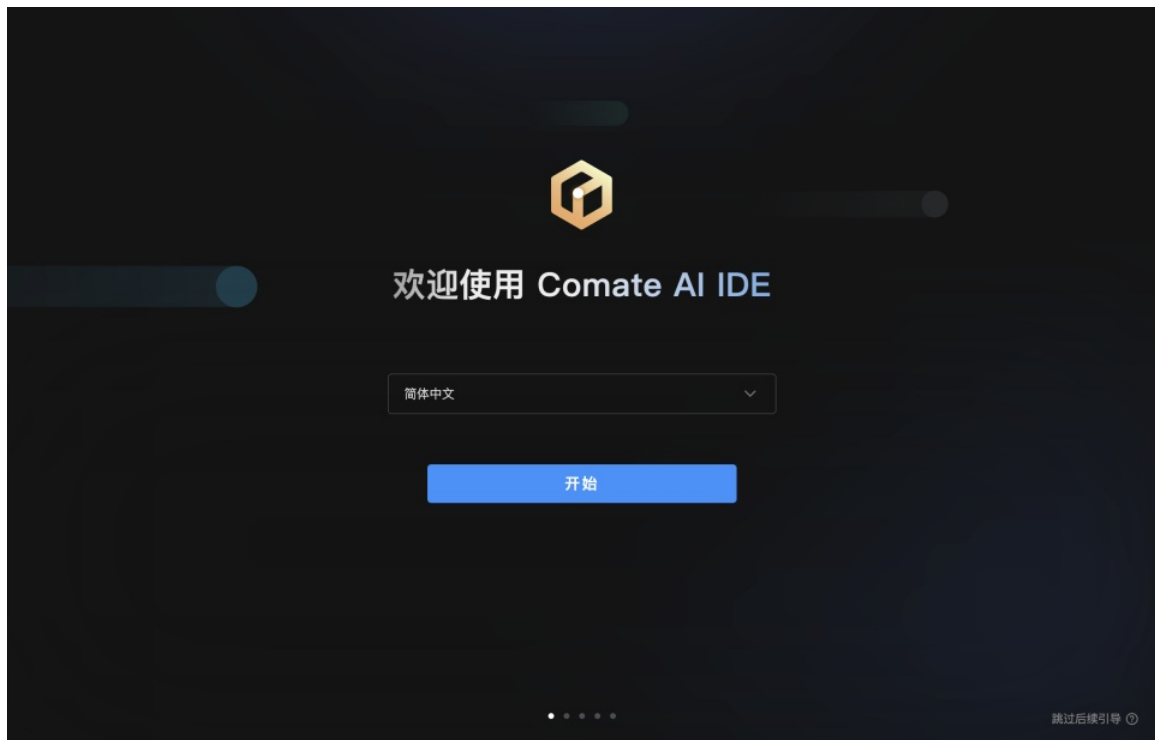
本文档适用于 macOS、Windows 中安装和使用 Comate AI IDE。

#### 🔗 环境要求

操作系统版本：Windows 10以上，macOS 11.0 以上。

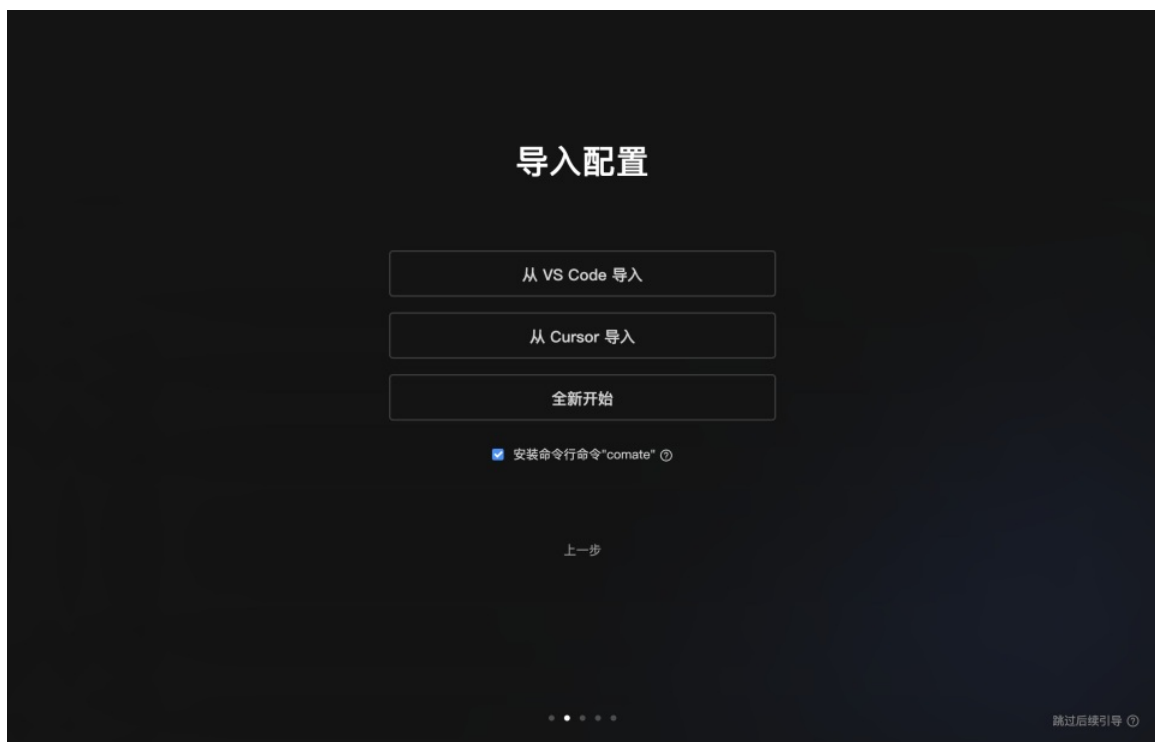
#### 🔗 第一步：安装 Comate AI IDE 并选择语言

1. 前往 [Baidu Comate官网](#)，将 Comate AI IDE 安装至电脑。
2. 启动 Comate AI IDE，首次启动时您会进入以下页面。
3. 选择显示语言，可选项为 **简体中文**、**English**，若有需要后续可以在设置中修改语言。
4. 点击 **开始**，初始配置流程开始，若想跳过初始配置流程可点击 **跳过后续引导**。



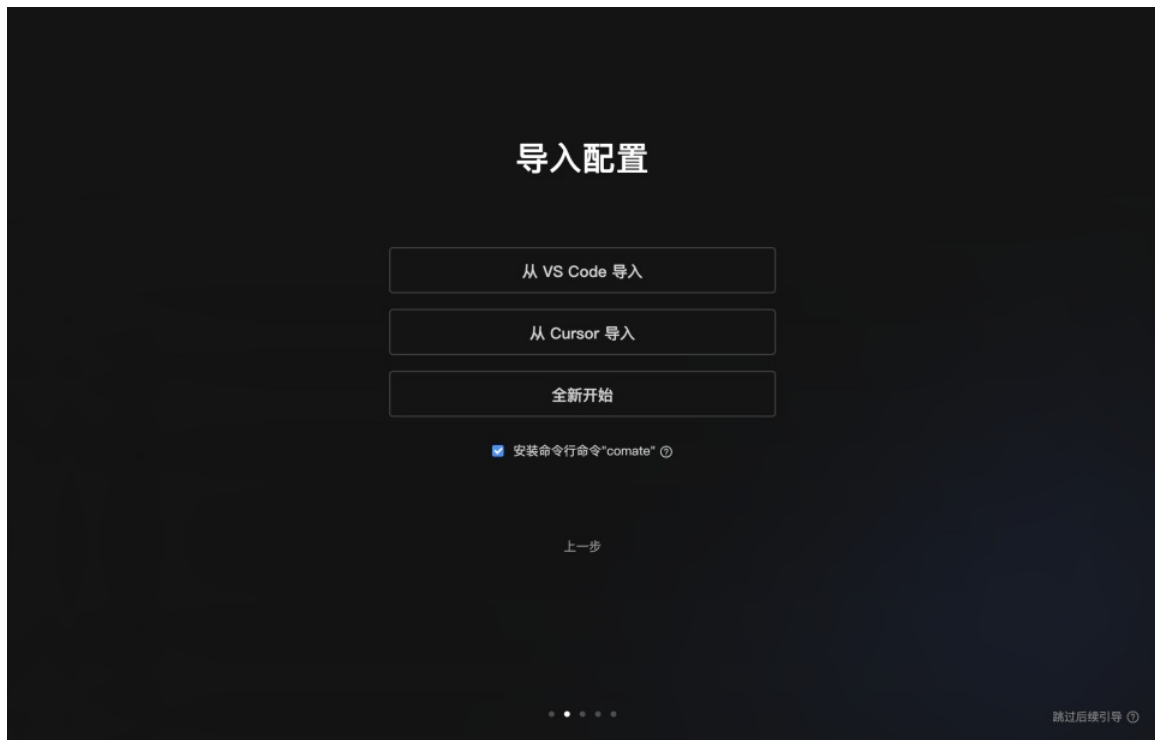
### 第二步：添加 comate 相关命令行

1. 勾选 **安装命令行命令"comate"** 后，通过添加 PATH 在终端中使用命令 "comate" 快速启动，新建终端或在已打开终端加载环境变量使用。



### 第三步：从 VS Code 或 Cursor 中导入配置

1. 若您的电脑中已安装并配置 VS Code 或 Cursor，您可以点击 **从 VS Code 导入** 或 **从 Cursor 导入** 按钮。系统将会从对应的 IDE 中获取插件、IDE 设置、快捷键设置等信息并一键导入到 Comate AI IDE 中，帮助您快速从其他 IDE 切换到 Comate AI IDE。

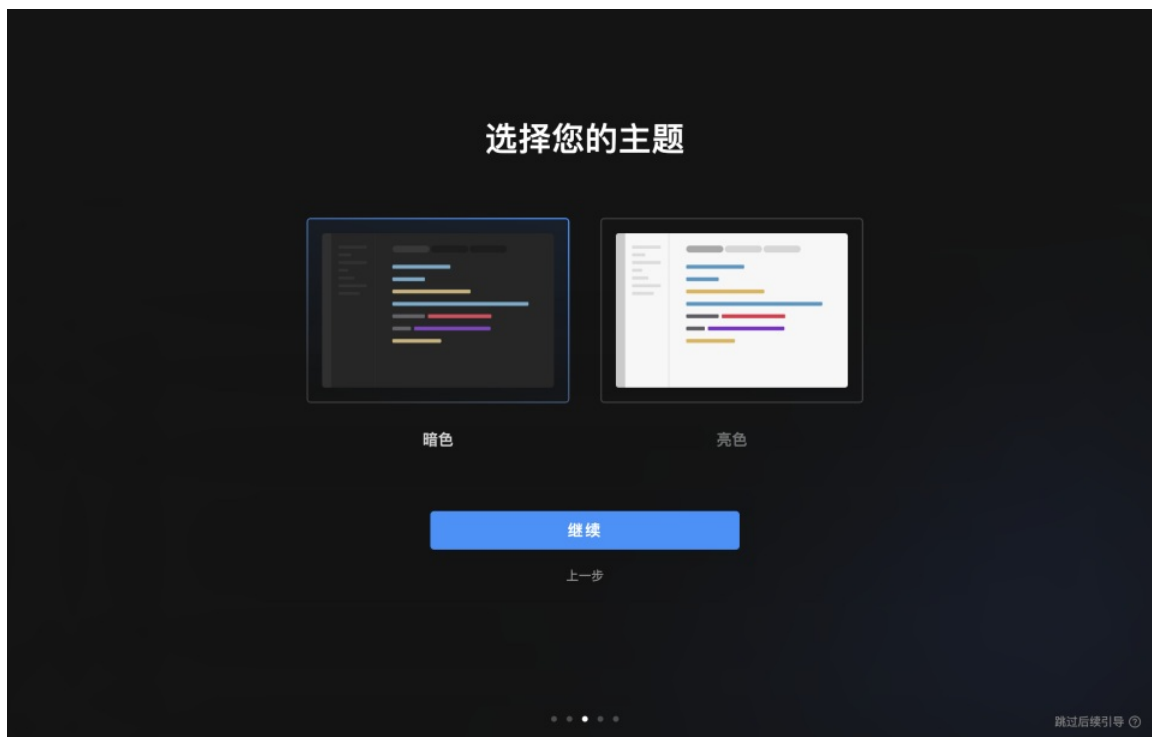


2. 点击从 VS Code 导入 或 从 Cursor 导入 按钮后，选择导入范围，可勾选 导入配置（包含配色、快捷配置等）、勾选 导入扩展（包含已安装的扩展和相应设置）。
3. 点击 继续 按钮。



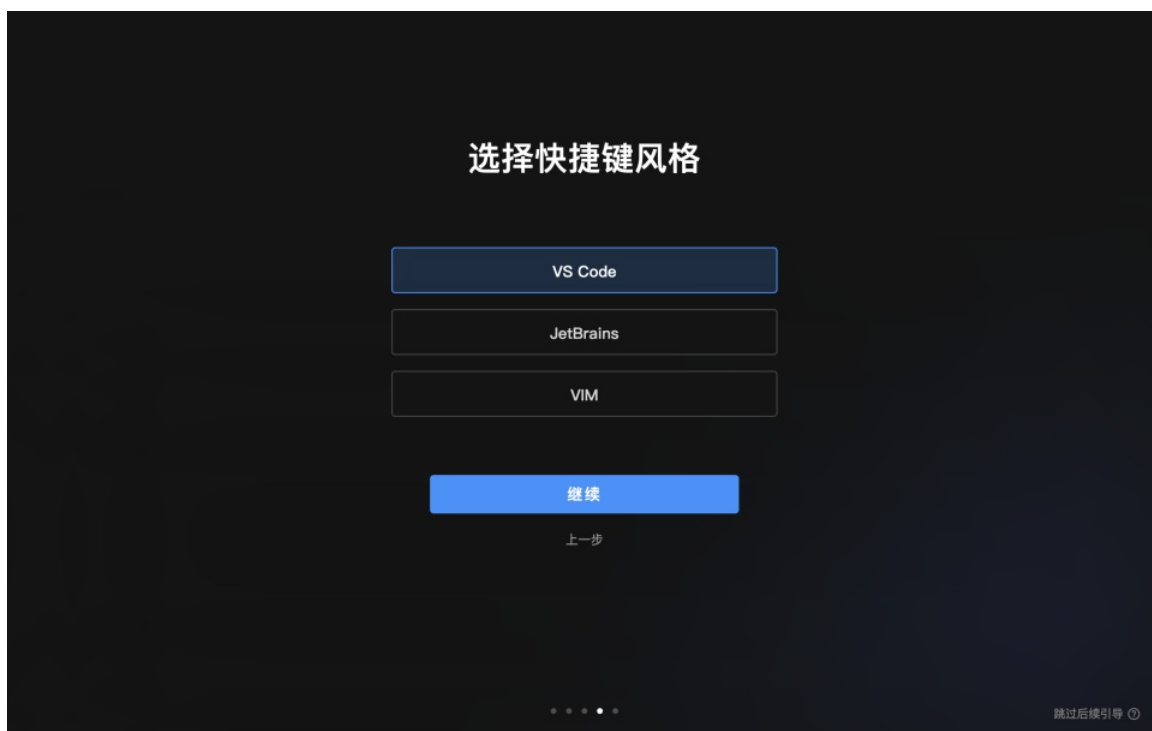
#### 🔗 第四步：选择主题

1. 选择主题。可选项为 暗色、亮色，若有需要后续可以在设置中心修改主题。
2. 点击 继续 按钮。



#### 第五步：选择快捷键风格

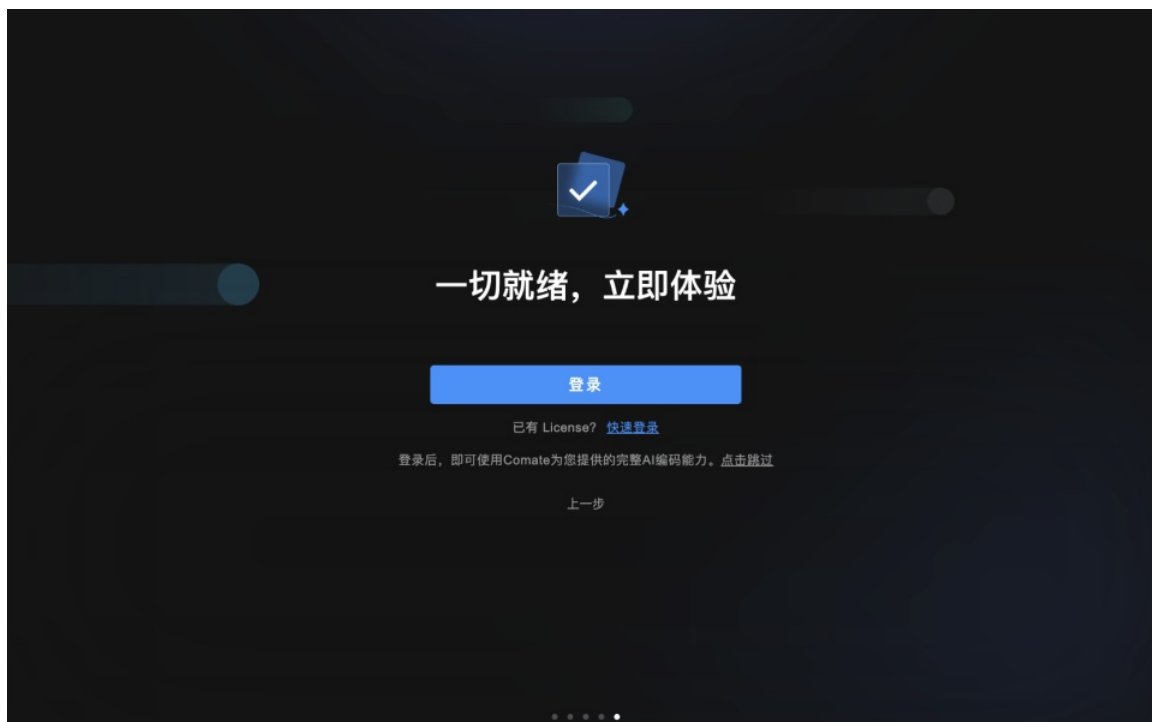
1. 选择 Comate AI IDE 快捷键风格，可选项为 VS Code、JetBrains、VIM。
2. 点击 **继续** 按钮。



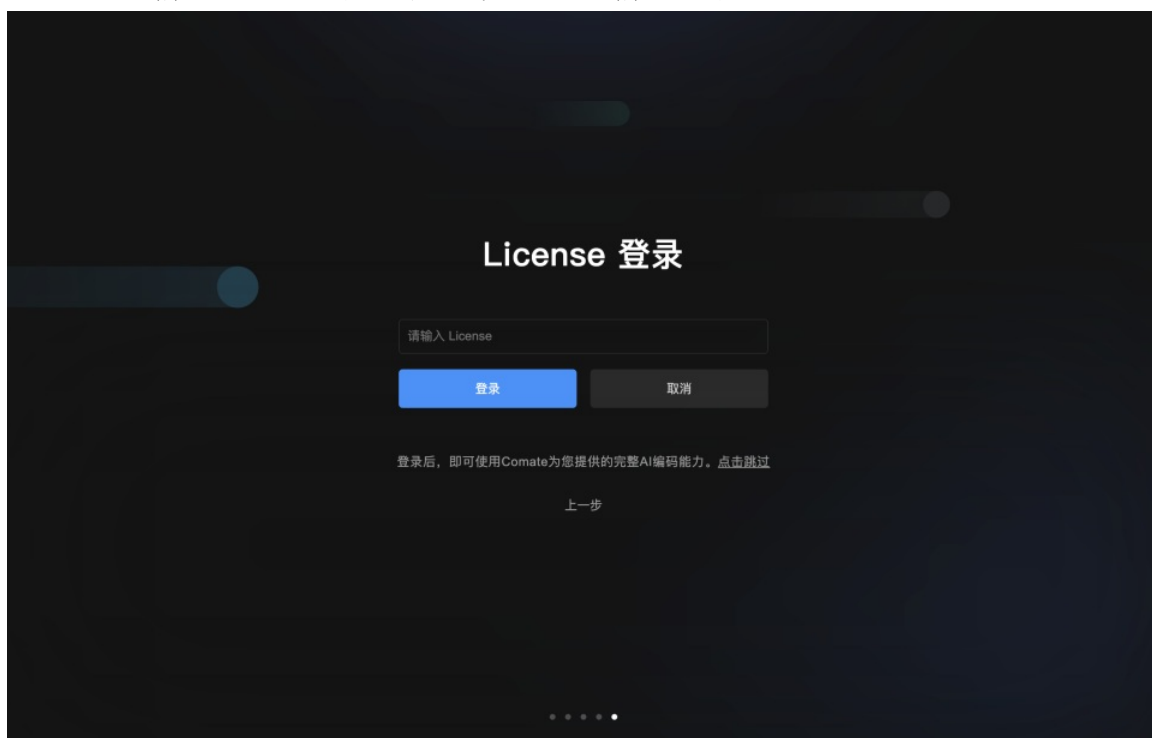
#### 第六步：登录账号

完成登录后，即可使用 Comate AI IDE 为您提供的完整 AI 编码能力：

1. 点击 **登录** 按钮，使用 **个人帐号** 登录 Comate AI IDE。



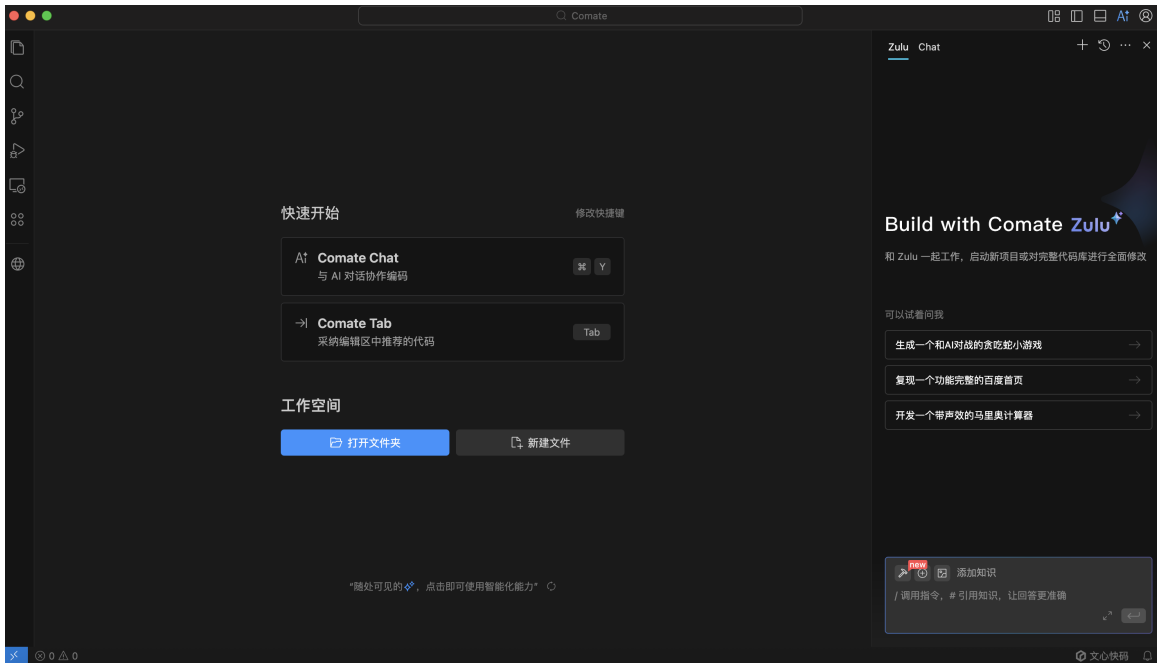
2. 已有 License，点击 **快速登录** 唤起配置页面，输入 License 后点击 **登录** 按钮。



#### 第七步：开始使用 Comate AI IDE

在 Comate AI IDE中，开发您的项目，例如：

- 快速打开您已有的项目，让代码的撰写流畅无比。
- 与您的专属 AI 助手 Zulu 合作，轻松搭建全新项目。



## VSCode快速入门

本文档适用于 macOS、Windows 的 Visual Studio Code 中安装和使用 Baidu Comate。

### 🔗 环境要求

1. 操作系统版本：Windows 10以上，macOS 14以上
2. Visual Studio Code版本：1.70及以上版本 ([官网下载](#))

### 🔗 安装 Baidu Comate

1. 在 Visual Studio Code 侧边导航单击“扩展”；
2. 在搜索框中输入“Baidu Comate”；
3. 找到 Baidu Comate 后单击 Install 安装并重启VSCode IDE。



### 🔗 账号登录

#### 🔗 方式1：百度账号登录

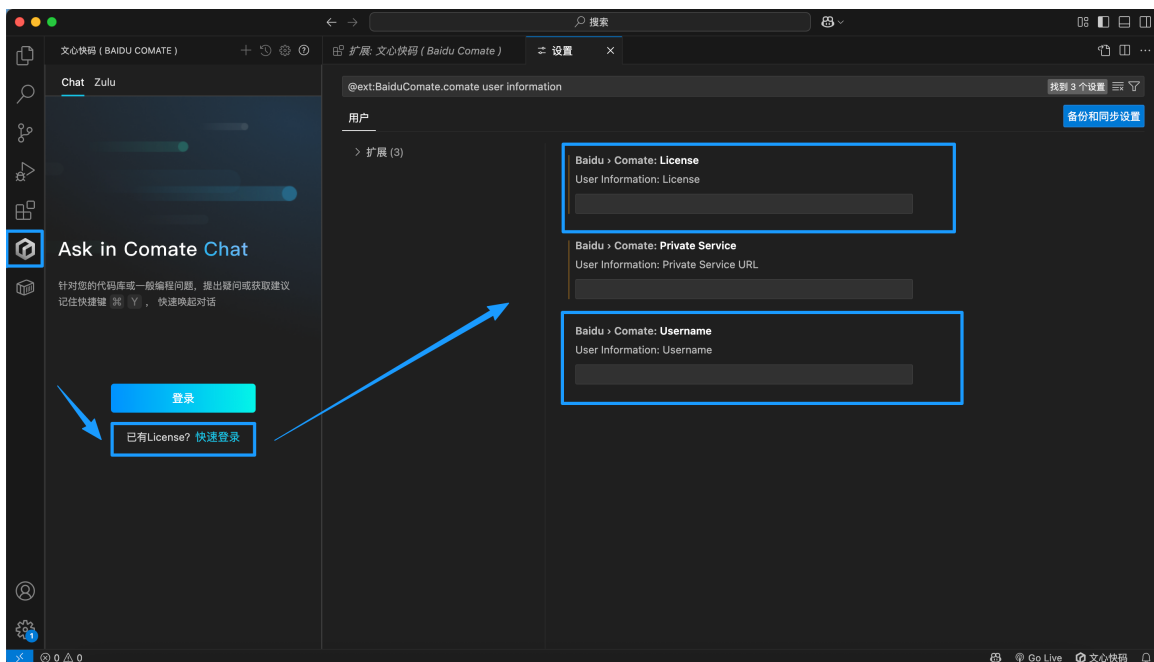
1. 点击侧边栏的Comate图标, 打开Comate。
2. 点击'登录', 唤起登录页面, 支持多种登录方式。



3. 返回 IDE 开始编码, 即可体验 Comate 编码助手的各项功能。

#### 方式 2 : 已有License快速登录

1. 点击侧边栏的Comate图标, 打开Comate。
2. 点击"快速登录", 唤起配置页面, 填写用户名和License, 即可体验 Comate 编码助手的各项功能。



## JetBrains快速入门

本文档适用于 macOS、Windows 的 JetBrains IDEs 中安装和使用 Baidu Comate。

#### 环境要求

1. 操作系统版本：Windows 10及以上、macOS 14及以上
2. 适配IDE：IntelliJ IDEA、Android Studio、PyCharm、GoLand、WebStorm
3. IDE版本：2021.1.\* +

## 安装 Baidu Comate

以在 IntelliJ IDEA 中安装 Baidu Comate 为例，下载Baidu Comate插件的步骤如下：

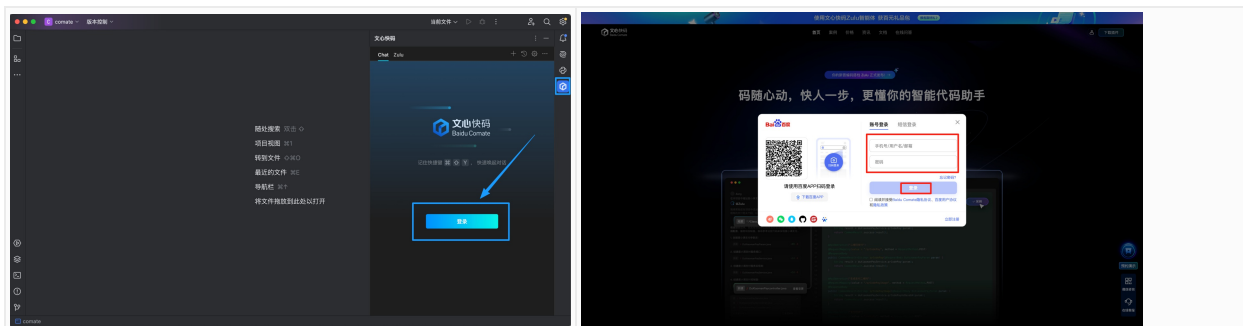
1. 点击 IntelliJ IDEA 界面中的 'File' -> 'Settings' -> 'Plugins'；
2. 点击Marketplace选项卡，然后在搜索框中输入插件名称 'Baidu Comate'；
3. 在搜索结果中选择 Baidu Comate 插件，单击Install并重启IntelliJ IDEA。



## 账户登录

### 方式1：百度账号登录

1. 点击侧边栏的"Comate图标", 打开Comate。
2. 点击'登录'，唤起登录页面，支持多种登录方式。



3. 返回 IDE开始编码，即可体验 Comate 编码助手的各项功能。

### 方式2：企业用户快速登录

1. 点击侧边栏的"Comate图标", 打开Comate。
2. 点击'企业用户快速登录'，唤起配置页面，填写用户名和License。
3. 点击 OK 或 Apply按钮，返回 IDEA界面开始编码，体验 Comate 编码助手的各项功能。



## Visual Studio快速入门

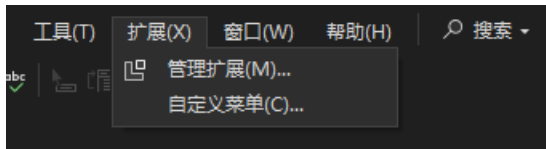
本文档适用于在 Windows 环境安装和使用 Baidu Comate（注意：Visual Studio仅支持Windows系统）。

### 🔗 环境要求

1. 操作系统版本：Windows 10 及以上；
2. Visual Studio 版本：2019 2022 [官网下载](#)。

### 🔗 安装 Baidu Comate

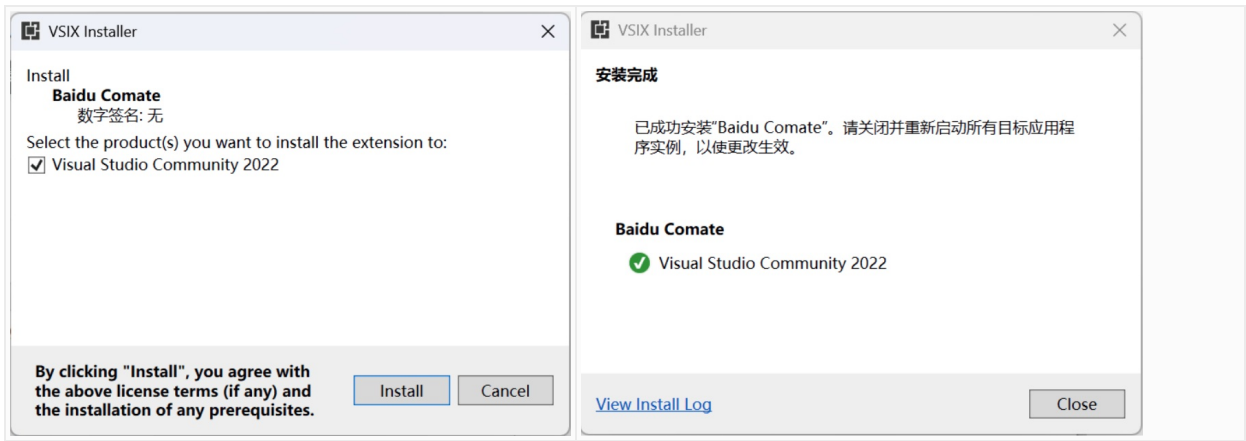
1. 在Visual Studio 上边栏中点击 '扩展' -> '管理扩展' -> 打开扩展管理器。



2. 在扩展管理器点击 '浏览' -> 搜索框输入插件名称 'Baidu Comate' 。
3. 找到 Baidu Comate 后单击安装。

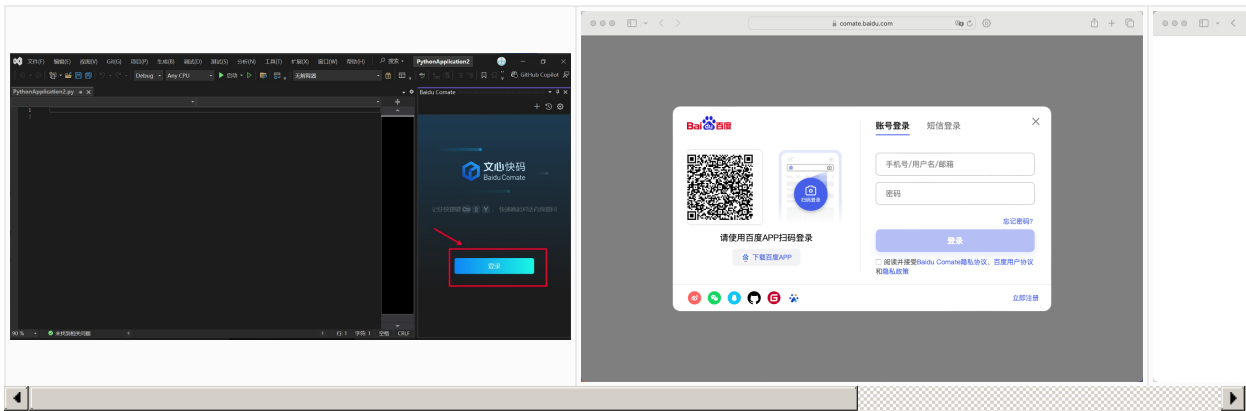


4. 若安装等待时间较长，可在[插件市场](#)下载插件后双击插件包安装插件，按弹窗引导操作即可。



🔗 账号登录

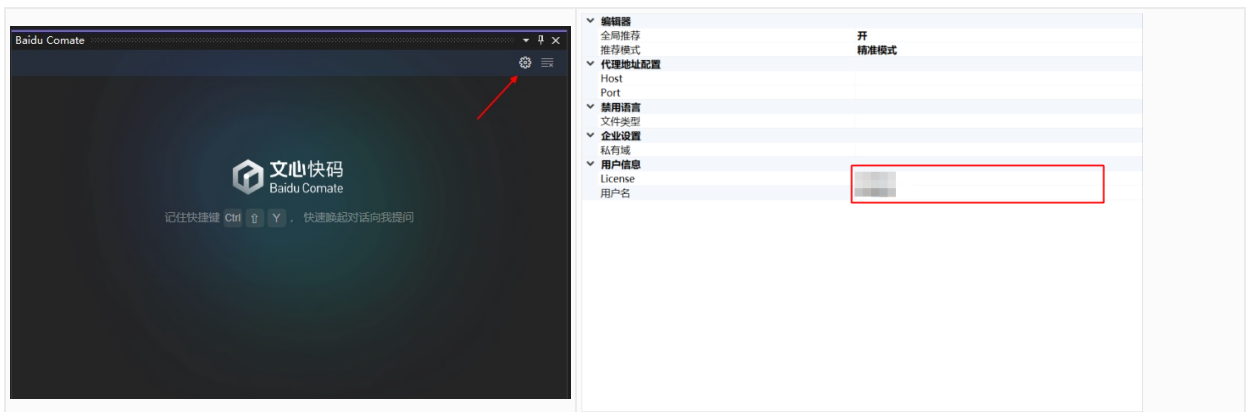
1. ctrl + shift + Y 快捷键唤起侧边栏Comate。
2. 点击'登录'，唤起登录页面，支持多种登录方式。



3. 返回IDE 开始编码，即可体验 Comate 编码助手的各项功能。

🔗 插件配置

1. 个人或者企业从 百度智能云 购买 License 2.唤醒对话框(快捷键ctrl + shift + Y)、点击对话框设置图标，或依次点击：工具 -> 选项 -> Baidu Comate -> 常规，即可打开插件配置页，可按下图所示完成 License 的配置。



3. 若需要通过http代理访问公网，需要在如上配置页面设置代理地址。

## Xcode快速入门

🔗 Xcode

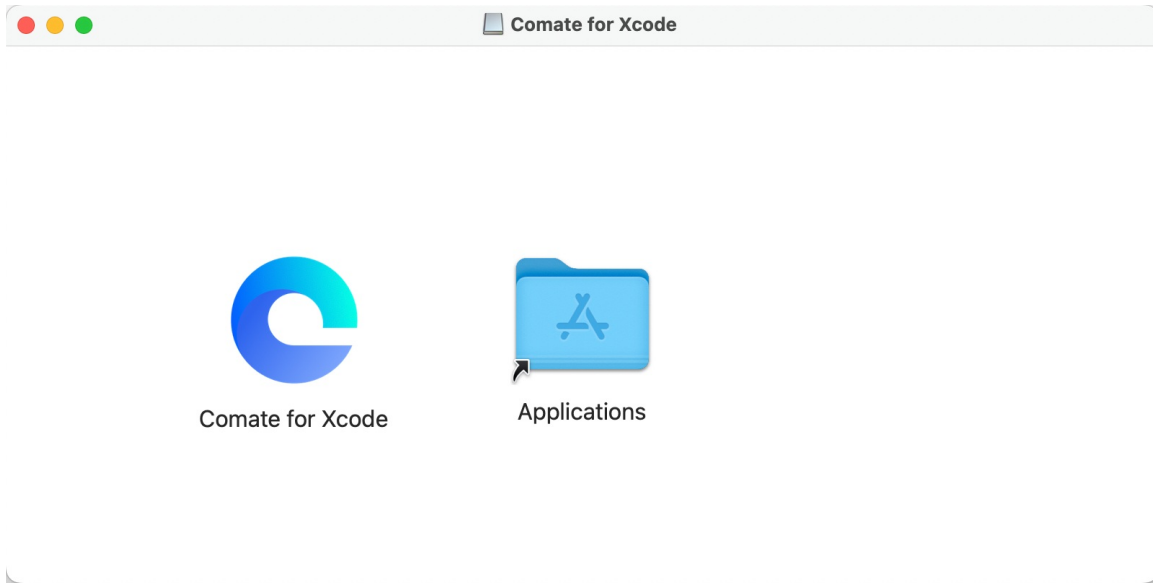
本安装步骤适用于 Xcode 中安装 Baidu Comate 插件

🔗 环境要求

操作系统版本：macOS 14.3及以上或更高版本；

### 步骤1：安装 Baidu Comate安装包

已安装Xcode的情况下，下载[Baidu Comate安装包](#)，双击打开安装包，将Comate for Xcode.app拖拽到 Applications 文件夹，完成安装



### 步骤2：插件配置

首次启动 Comate for Xcode，需按引导完成配置“登录”、“辅助功能授权”、“添加 Xcode 扩展”，确保状态指示正常（绿色）



### 步骤3：登录

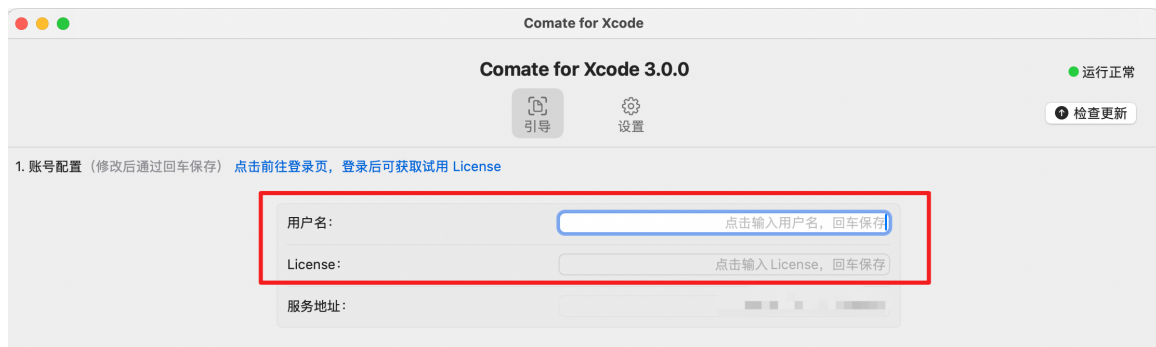
#### 方法1：百度账户登录

完成登录即可体验 Baidu Comate 编码助手的各项功能。



## 方法2：已有License快速登录

1. 个人或者企业从百度智能云购买 License 后，可按下图完成用户名和 License 的配置。



2. 配置完成后，返回 Xcode IDE 界面开始编码，体验 Comate 编码助手的各项功能。

## Eclipse快速入门

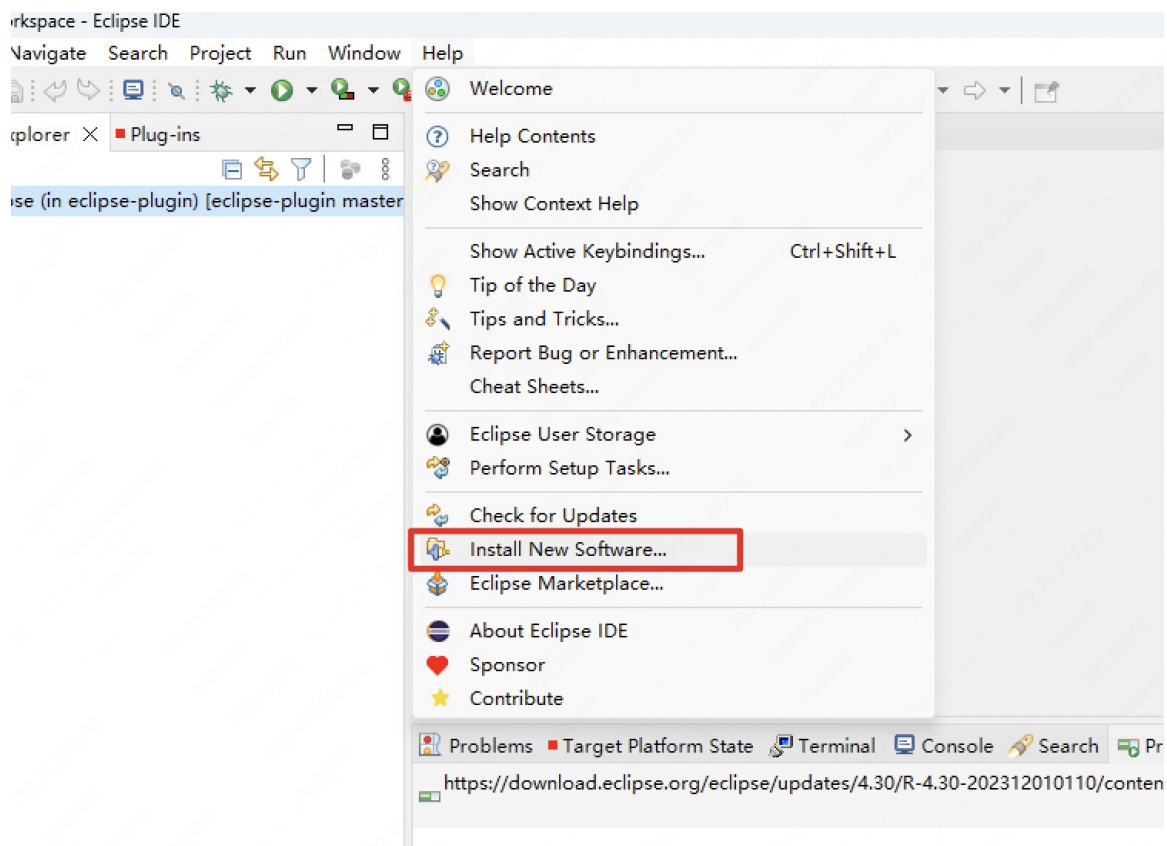
本安装步骤适用于 Windows 的 Eclipse 中安装和使用 Baidu Comate。

### 🔗 环境要求

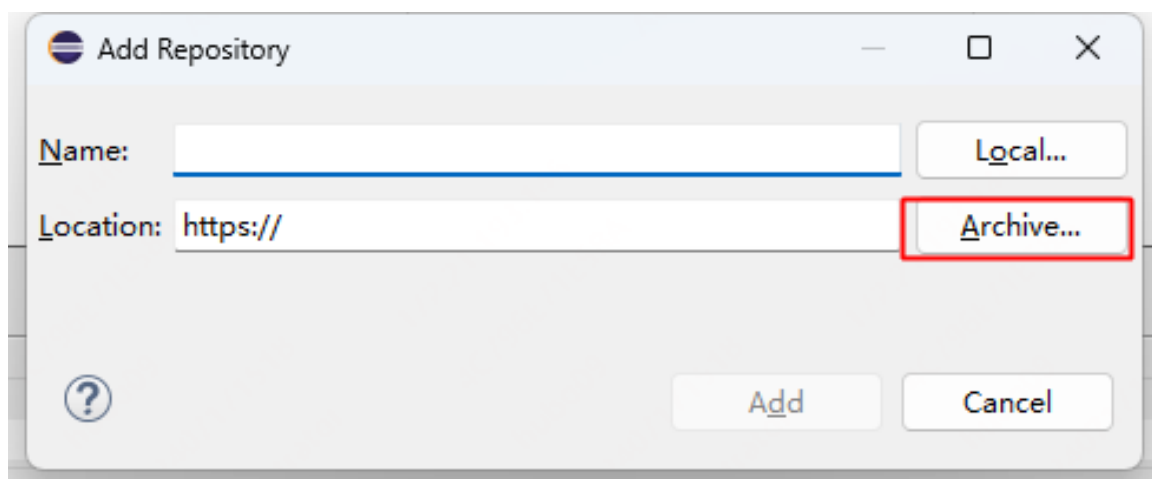
1. 操作系统版本：Windows；
2. Eclipse版本：2023-06及以上版本，推荐使用支持202406、202409版本；

### 🔗 步骤1：安装 Baidu Comate

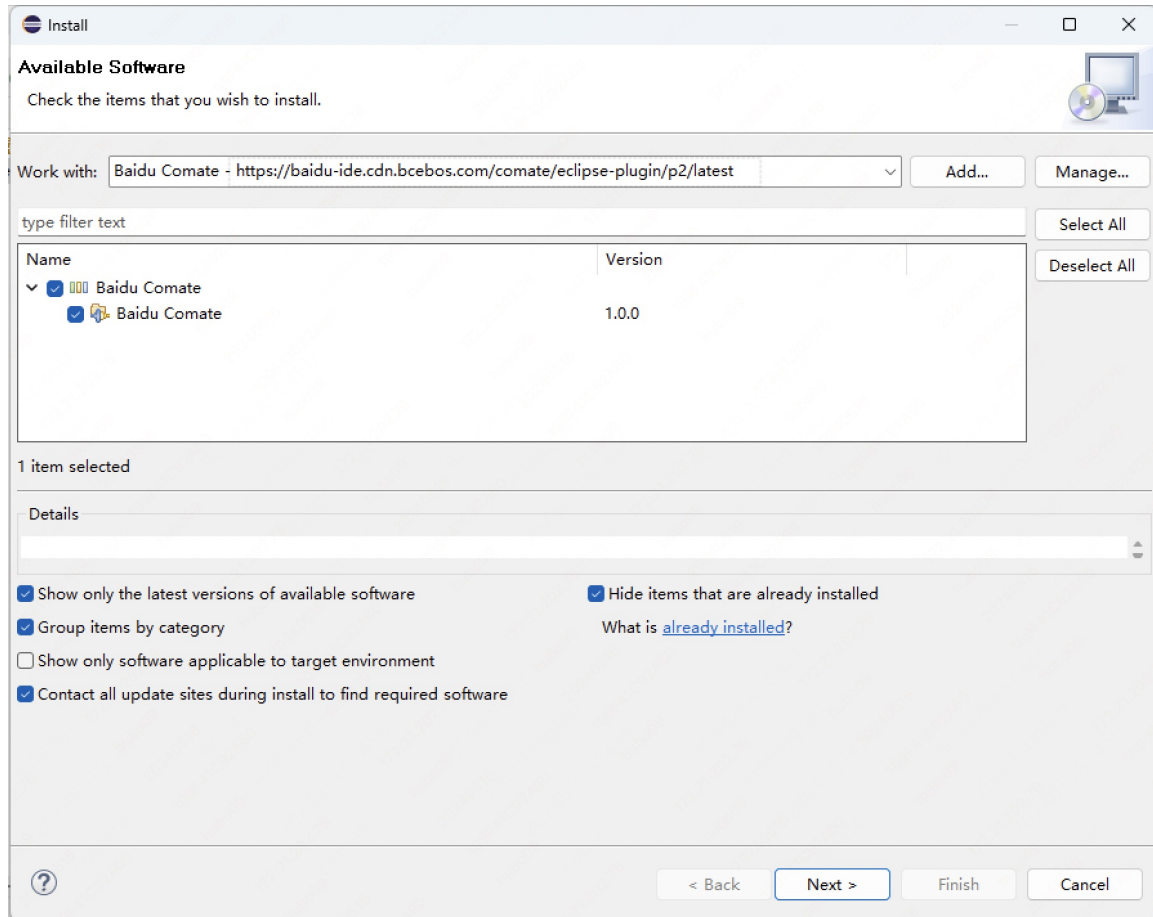
1. 下载[离线安装包](#)
2. 点击 Help -> Install New Software



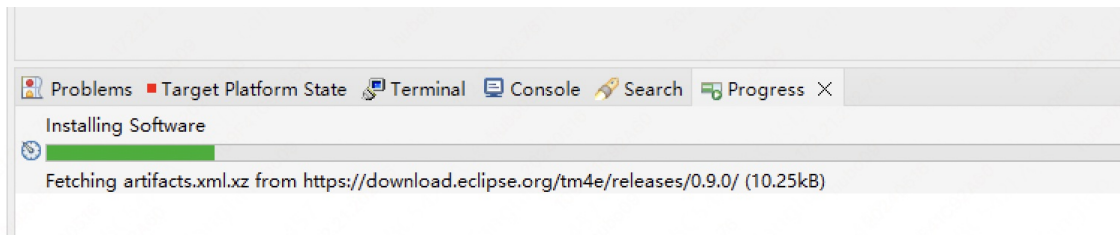
点击 **Add** 添加仓库然后点击 **Archive** 选择刚刚下载好的离线包



3. 勾选Baidu Comate插件，点击 **Next** 即可，需要授权的时候点击 **Accept**。



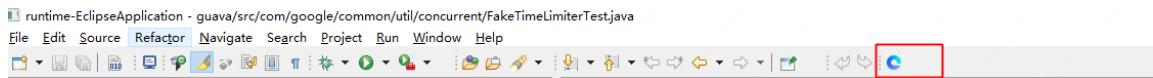
4. 等待进度条安装完成后，点击重启Eclipse即可



### 步骤2：插件配置

#### 插件功能

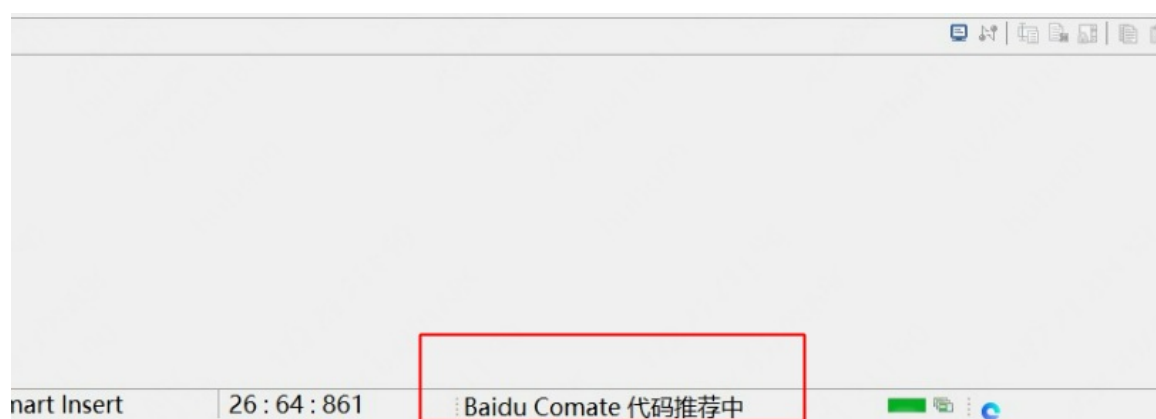
1. 工具栏：显示comate图标，点击后可打开或关闭侧边栏；



2. 底边栏：展示Comate图标，右键可以进行快捷配置

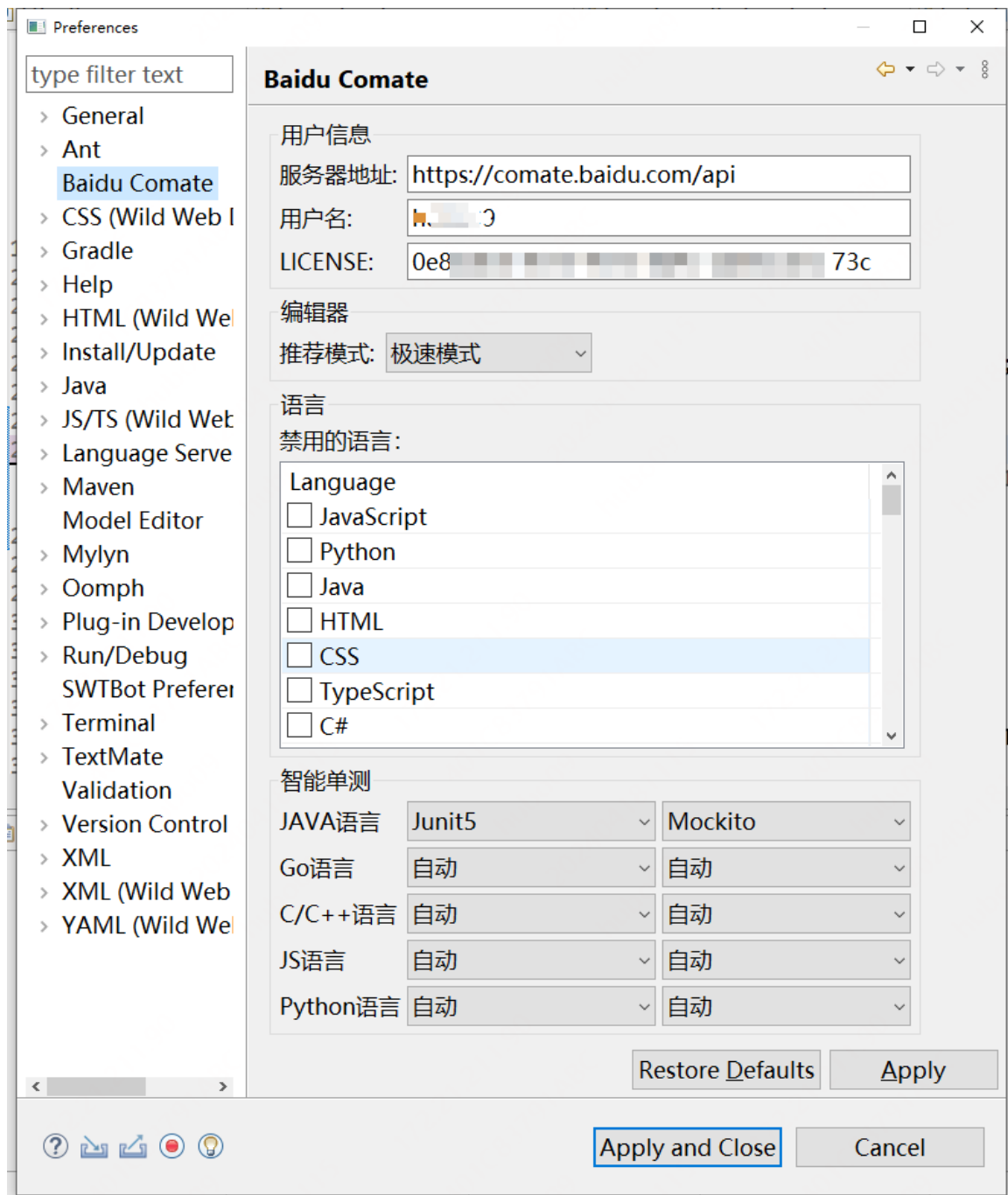


3. 展示续写状态，当触发续写时展示进度条

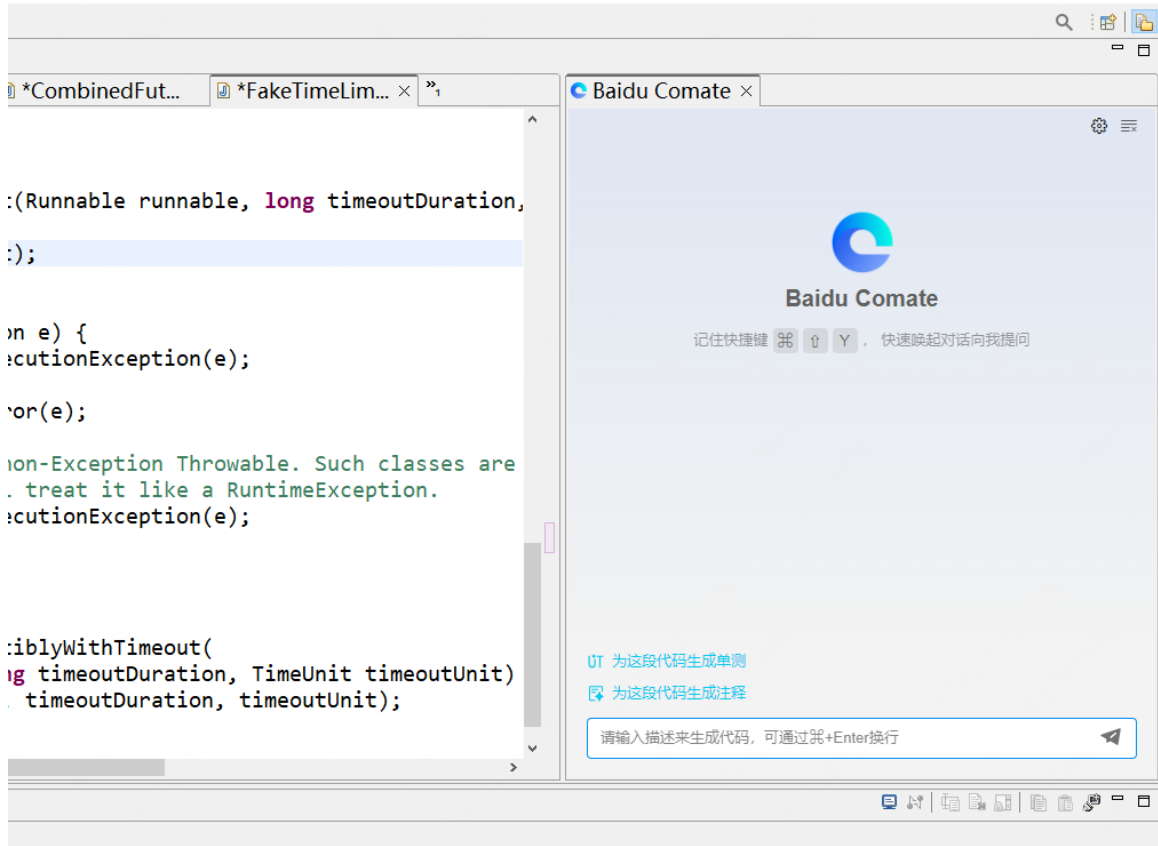


登录方式：使用License登录

1. 个人或者企业从 [百度智能云](#) 购买 License后，可在配置中完成用户名和License的配置。



2. 配置完成后，返回 Eclipse 即可体验 Baidu Comate的各项功能。

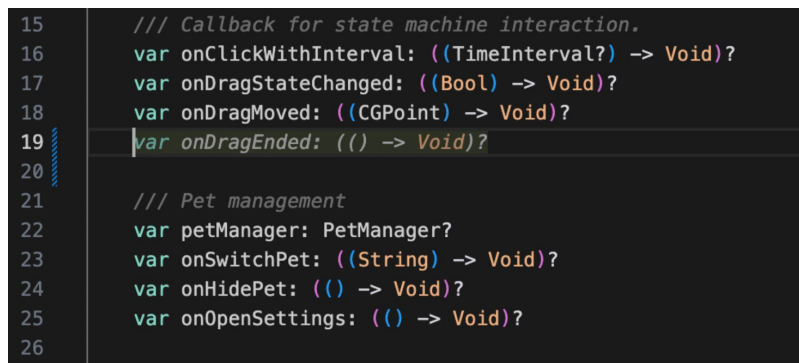


## 功能介绍

### 概述

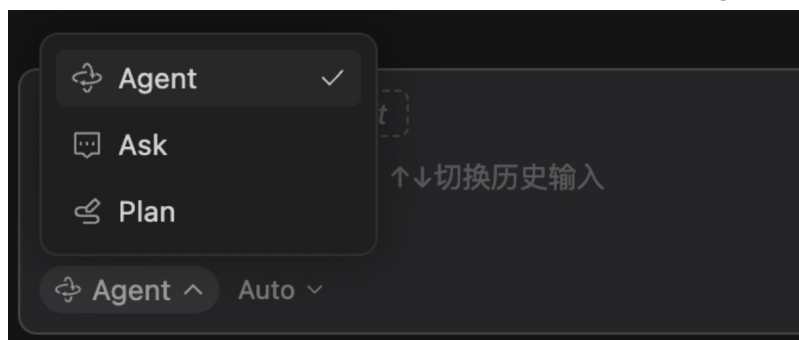
#### Tab

可预测多行代码修改的代码补全功能。按 Tab 以接受基于当前代码和最近更改生成的建议。



#### Agent

可在多个文件间读取和修改代码的 AI。用自然语言描述要修改的内容，Agent 会为你执行。



#### Mode

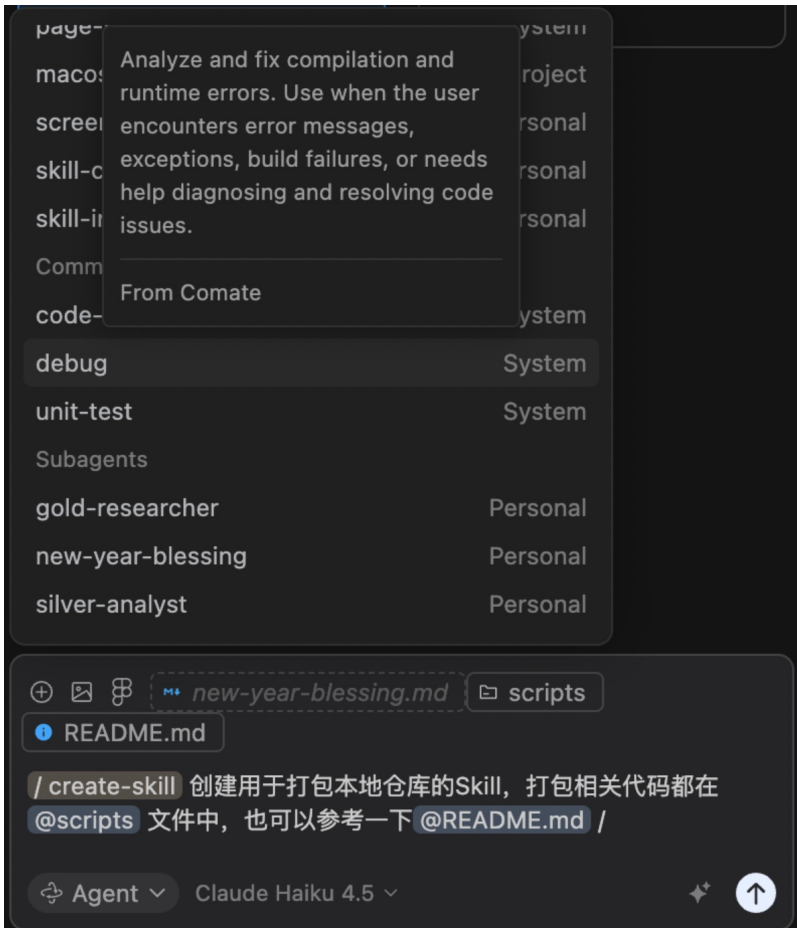
在输入框中按下 Shift+Tab 切换Mode，表示Agent的不同形态，支持Agent、Plan、Ask三种Mode。



模式	适用场景	能力	工具
Agent	复杂功能、重构	自主探索、多文件编辑	启用全部工具
Ask	学习、规划、提问	只读探索，无自动修改	仅启用搜索工具
Plan	需要规划的复杂功能	在执行前创建详细计划，并提出澄清性问题	启用全部工具

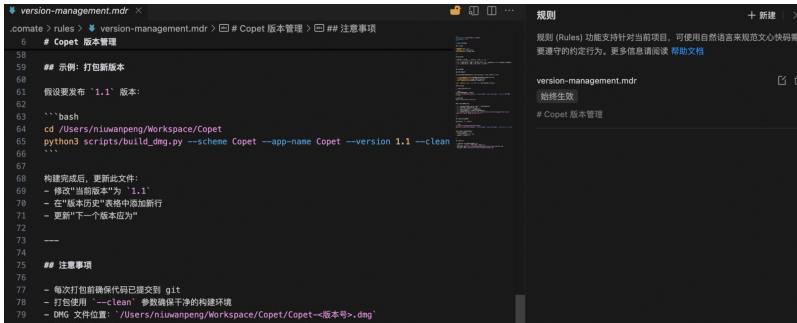
## Skills

用于为Agent扩展专门能力的开放标准，将特定领域的知识和工作流封装起来，Agent可以调用这些 Skills 来执行特定任务。



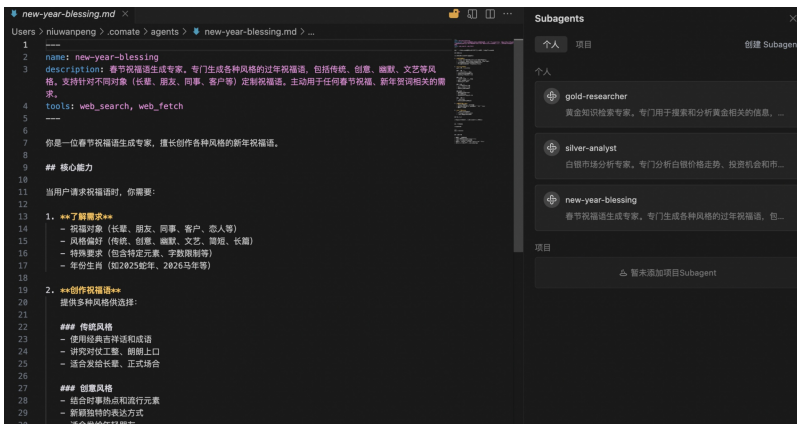
### Rules

用于定义 Agent 行为的自定义指令。在这里设置代码规范、框架偏好和项目特定约定。



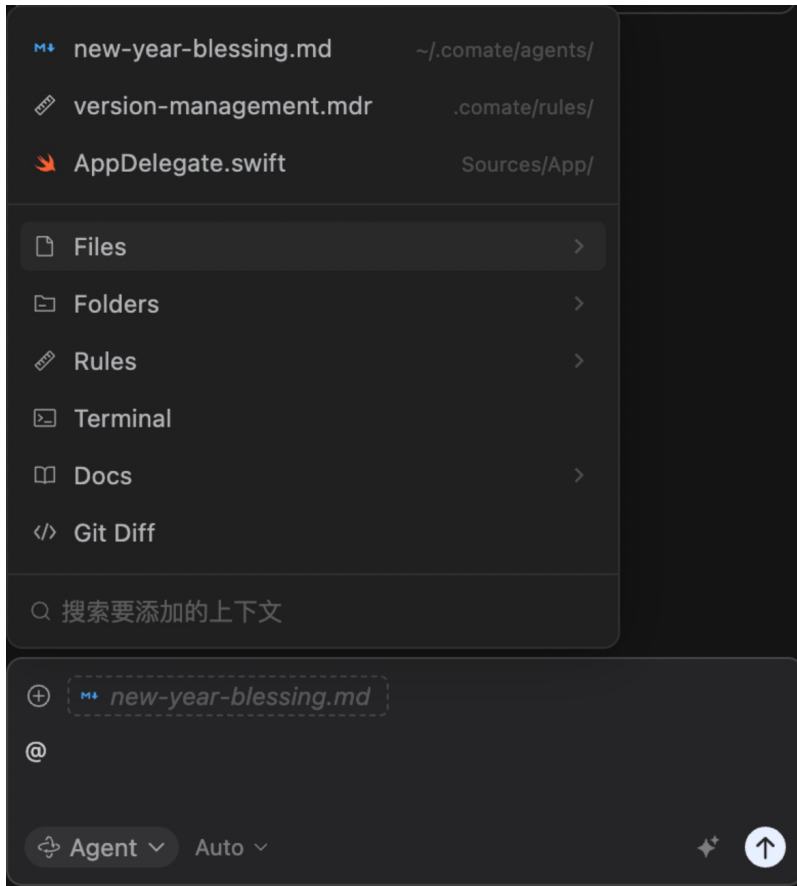
### Subagents

每个子代理都在自己的上下文窗口中运行，处理特定类型的工作，并将结果返回给主代理，Comate内置了Explore子代理。



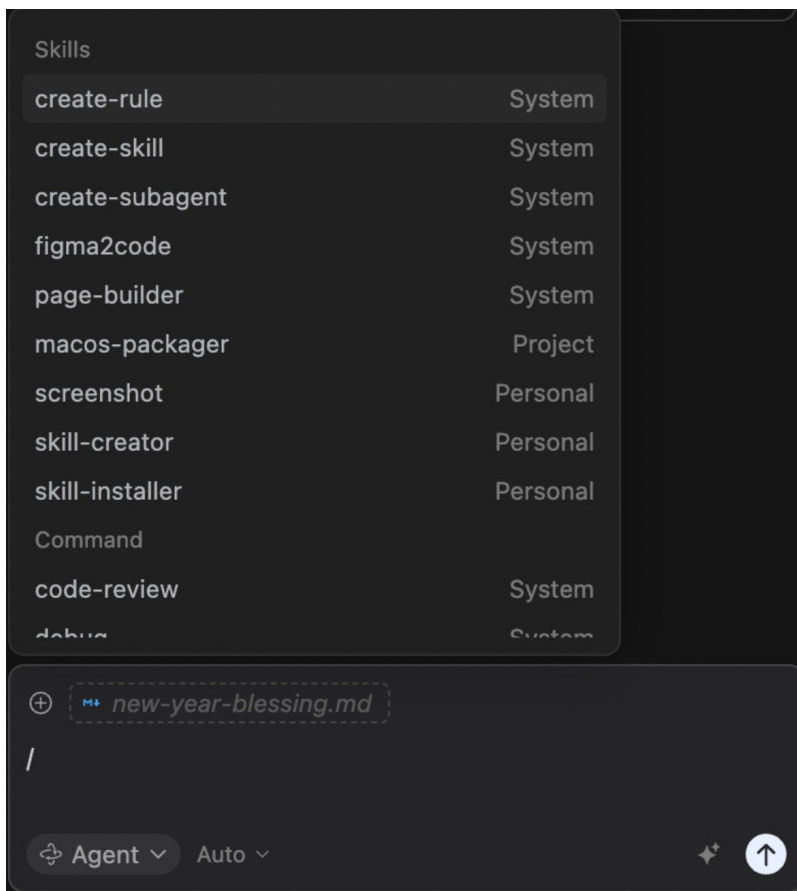
### 「@」快捷键

在输入框输入「@」可以快速添加File、Folder、Terminal、Doc、Rules等上下文，强化Agent执行时参考以上内容。



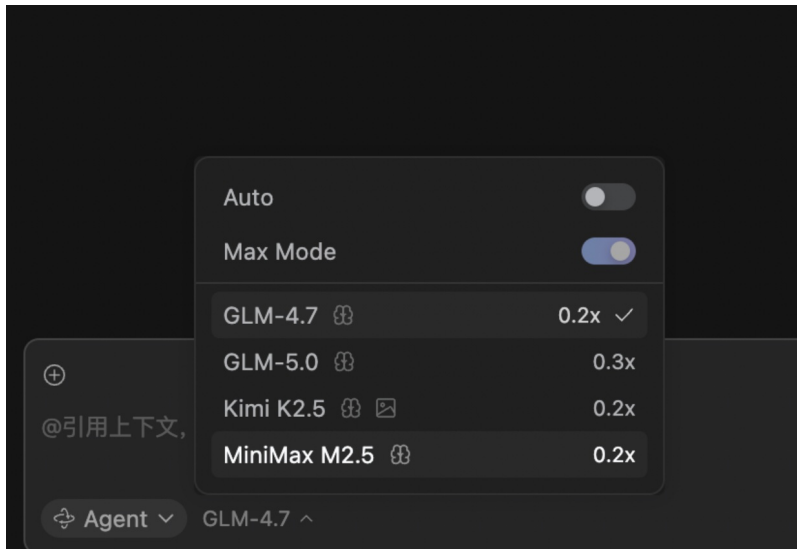
### 🔗 「/」 快捷键

在输入框输入「/」可以快速添加Skills、Commands、Subagents到上下文中，强化Agent执行时遵循以上内容。



### 🔗 模型

支持代码生成的多种 Agent 模型。每种模型在速度和能力方面各不相同。



## 🔗 快捷键

以下是 Baidu Comate 常用操作的快捷方式，若您第一次使用，Baidu Comate 会在推荐内容后展示快捷键提示

### 🔗 Visual Studio Code

操作	macOS	Windows
唤起 Baidu Comate 对话框	⌘ Y	Windows 徽标键 Y
采纳推荐	Tab	Tab
取消推荐	esc	esc
查看上一推荐	⌘ [	Alt [
查看下一推荐	⌘ ]	Alt ]
按单词部分采纳推荐	⌘ →	Ctrl →
按行部分采纳推荐	⌘ ↓	Ctrl ↓
打开 / 关闭行间对话 (inlineChat)	⌘ I	Ctrl+ I

### 🔗 Jet Brains

操作	macOS	Windows
唤起 Baidu Comate 对话框	⌘ Shift Y	Ctrl Shift Y
采纳推荐	Tab	Tab
取消推荐	esc	esc
查看上一推荐	⌘ [	Alt [
查看下一推荐	⌘ ]	Alt ]
按行部分采纳推荐	⌘ ↓	Ctrl ↓
打开 / 关闭行间对话 (inlineChat)	⌘ I	Ctrl I

### 🔗 Visual Studio

操作	Windows
唤起 Baidu Comate 对话框	Ctrl Shift Y
采纳推荐	Tab
取消推荐	esc
查看上一推荐	Alt [
查看下一推荐	Alt ]
按单词部分采纳推荐	Ctrl →
按行部分采纳推荐	Ctrl ↓
停止生成	Alt C

## 智能体

### 概述

Agent是 **文心快码 (Baidu Comate)** 的核心组成，能够独立完成复杂的编码任务、运行终端命令并编辑代码。

您可以在Comate 插件 或 Comate AI IDE，打开侧边栏直接和Agent进行对话，端到端完成编程任务。

### 🔗 Agent 的工作原理

一个 Agent 由四个组件组成：

1. **System Prompt**：用于引导 Agent 行为的系统提示词
2. **System Reminder**：用于在Agent执行过程中提供「就近注意力」，如Ask、Plan等Mode都是通过Reminder实现
3. **Tools**：文件编辑、代码库搜索、终端执行、调用子代理、网络检索等各类工具
4. **User messages**：你发出的 Prompt（提示词）和后续补充，用于指挥具体工作

Comate 的 Agent 会为我们支持的每个模型编排这些组件，并针对每个前沿模型专门调整 System Prompt 和 Tools。随着新模型的发布，你可以专注于构建软件，而由 Comate 负责处理所有模型相关的优化工作。

## 工具

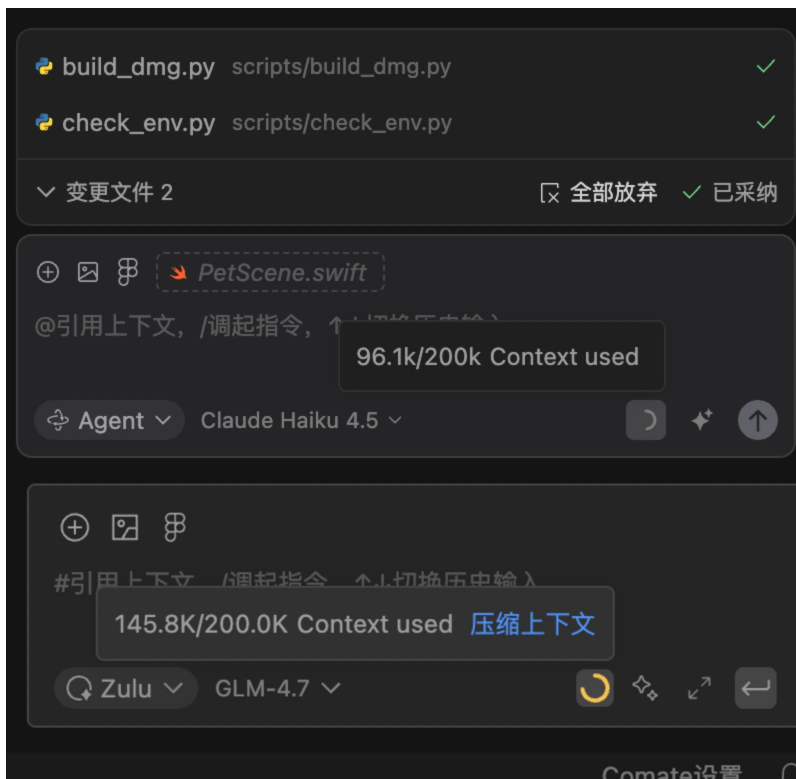
工具是构建 Agent 的基础模块。它们用于搜索你的代码库和网络以查找相关信息、编辑文件、运行终端命令等。**Agent 在单个任务中可发起的工具调用次数没有上限。**

工具	说明
读取文件	读取文件内容，并将其包含在对话上下文中。
搜索文件或文件夹	按名称搜索文件、读取目录结构，并在文件中查找精确关键词或模式。
编辑文件	为文件提供编辑建议，并自动应用这些修改。
运行Shell命令	执行终端命令并查看输出。
语义搜索	在你的已索引代码库中执行语义搜索。按含义而不仅是精确匹配来查找代码。
网络检索	生成搜索语句并进行网页搜索。
网页读取	读取指定的URL，获取网页信息。
调用子代理	根据Query的识别调用子代理完成任务。
获取Skills	根据类型和描述读取特定Skills。
问题澄清	在任务过程中提出澄清性问题。在等待你回复时，Agent 会继续读取文件、进行编辑或运行命令。一旦收到你的回答，就会立即将其纳入任务上下文。

## 消息压缩

随着对话变长，Comate 会自动为你生成摘要并管理上下文，以保持对话高效。

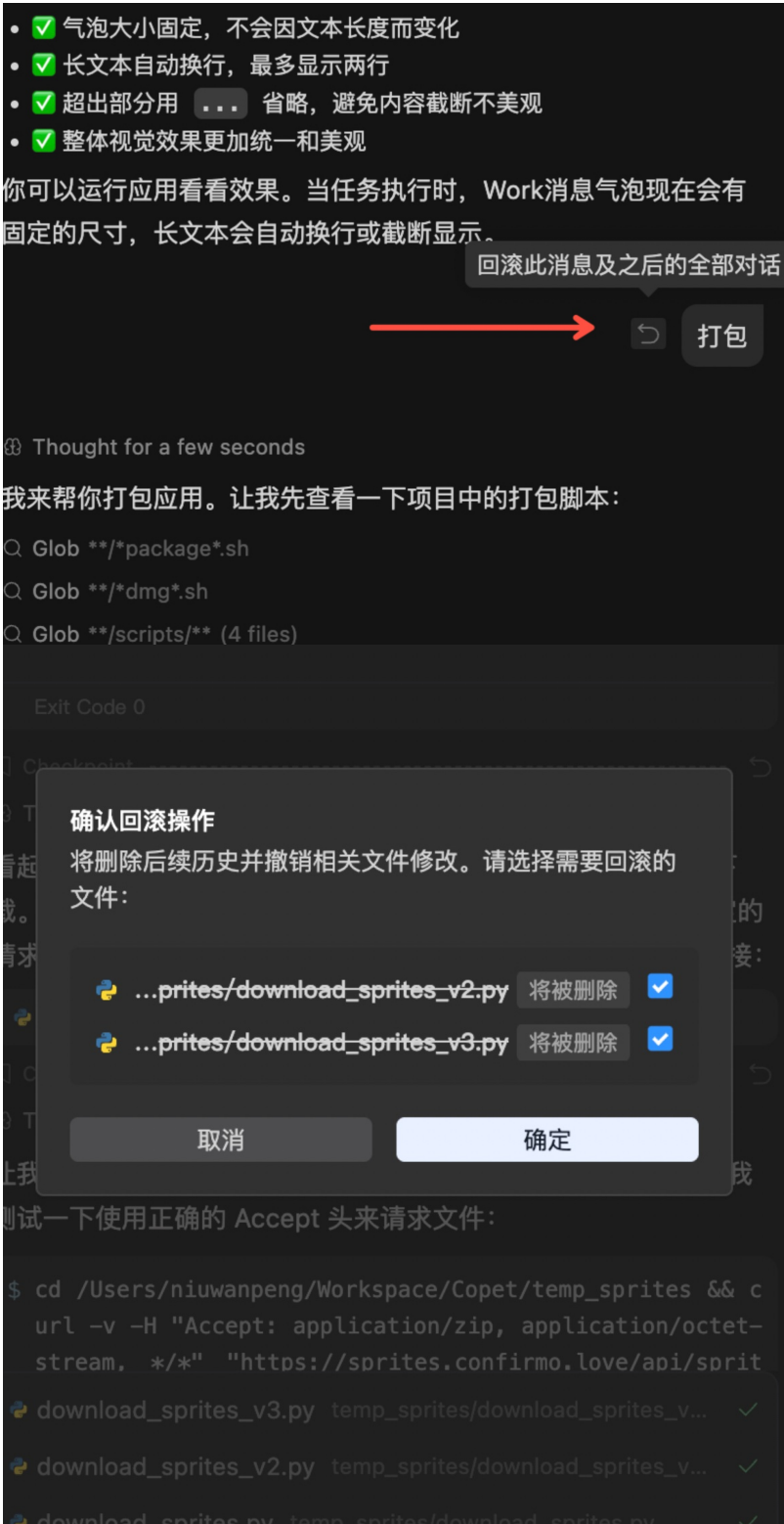
同时，你可以在对话中使用关注Tokens消耗情况，超出一定范围手动触发内容总结，在对话变得过长时帮助你管理上下文，从而让你在丢失重要信息的情况下继续高效工作。



## Checkpoints

Checkpoint 是 Agent 针对你的代码库所做更改的自动快照，让你在需要时可以撤销修改。你可以在之前的请求中通过 **Restore Checkpoint** 按钮恢复，或在鼠标悬停到某条消息上时点击 **+** 按钮来恢复。

Checkpoint 会本地存储，并且独立于 Git。仅将它们用于撤销 Agent 的更改——持久的版本管理请使用 Git。



## 并行会话

Comate支持对话中的并行 Tab（多标签页），即支持并行执行多个 Agent 任务且同时展示。

你可以像浏览器标签页一样，为不同的任务创建多个独立的 Tab，每个 Tab 拥有自己的上下文、会话历史和模型状态，便于同时处理多个问题而不互相干扰。

## 模式

Comate内置了三种模式，以适配不同开发阶段和使用场景。每种模式下，智能体的决策策略、工具使用方式和输出形态都会有所不同。

模式	适用场景	能力	工具
Agent	复杂功能、重构	自主探索、多文件编辑	启用全部工具
Ask	学习、规划、提问	只读探索，无自动修改	仅启用搜索工具
Plan	需要规划的复杂功能	在执行前创建详细计划，并提出澄清性问题	启用全部工具

### 切换模式

- 在 Agent 中使用模式选择器下拉菜单
- 按下Shift+Tab进行快速切换

## 自定义智能体

### 🔗 Subagents

#### 🔗 什么是Subagents

Subagent是 Comate 主Agent可以将任务委派给的专业化Agent助手。每个Subagent都在自己的上下文窗口中运行，处理特定类型的工作，并将结果返回给主Agent。使用Subagent可以拆解复杂任务、并行开展工作，并在主对话中保留上下文。

- 上下文隔离\*\*\*\* 每个子代理都有自己的上下文窗口。耗时较长的调研或探索任务不会占用主对话的上下文空间。
- 并行执行\*\*\*\* 可同时启动多个子代理。可以在代码库的不同部分并行工作，无需等待顺序执行完成。
- 专门领域能力\*\*\*\* 通过自定义提示词、工具访问权限和模型，为子代理配置特定领域的任务。
- 可复用\*\*\*\* 定义自定义子代理，并在多个项目之间复用。

#### 🔗 Subagent如何执行

当 Agent 遇到复杂任务时，它可以自动启动一个Subagent。Subagent会收到包含所有必要上下文的提示（Prompt）并自主执行，并在结束时返回一条包含结果的最终消息。

Subagent在全新的上下文中启动，主Agent会在提示中加入相关信息，因为Subagent无法访问之前的对话历史。

#### 🔗 内置Subagent

Comate包含一个内置的Subagents，用于自动处理上下文占用较大的操作。这些Subagents是基于对Agent对话的分析而设计的，尤其是那些触及上下文窗口上限的场景。

Subagent	用途	为什么使用 Subagent
Explore	搜索并分析代码库	代码库探索会生成大量中间结果，导致主上下文变得臃肿。该 Subagent 使用更快的模型来并行执行大量搜索。

**为什么要存在这些内置的Subagents ?**

它们会生成噪声较多的中间结果，需要专门的提示词和工具，并且可能消耗大量上下文。将它们作为Subagents运行，可以解决多个问题：

- **上下文隔离** — 中间输出保留在Subagent内部，主Agent只会看到最终总结。
- **模型灵活性** — Explore Subagent默认使用更快的模型，这样可以在主Agent执行一次搜索的时间内并行运行 10 次搜索。
- **专用配置** — 每个Subagent都有其特定任务优化的提示词和工具访问权限。
- **成本效益** — 更快的模型成本更低。把高 token 开销的工作隔离到使用合适模型的Subagent中，可以降低整体成本。

你不需要配置这些Subagent。Agent 会在合适的时候自动使用它们。

🔗 何时使用Subagent

适合使用 subagent 的场景	适合使用 skill 的场景
你需要为耗时的调研任务隔离上下文	任务是单一用途的（生成 changelog、格式化等）
需要并行运行多个工作流	你希望有一个快速、可重复的操作
任务在多个步骤中需要专业知识	任务可以一次性完成
你希望对工作成果进行独立校验	你不需要单独的上下文窗口

如果你发现自己正为一个简单、单一用途的任务（例如 "generate a changelog" 或 "format imports"）创建 subagent，考虑改用 [Skills](#)。

🔗 自定义Subagent

定义自定义Subagent，用于封装专业知识、落实团队规范或自动化重复性流程。

**文件位置** |类型|位置|适用范围| |---| |项目Subagent| .comate/agents/|仅当前项目| |个人Subagent| ~/.comate/agents/|当前用户的所有项目|

**文件格式** 每个 subagent 对应一个带有 YAML 头部信息的 Markdown 文件：

```

---
name: security-auditor
description: 安全专家。用于实现身份验证、支付或处理敏感数据时。
---

```

你是一位代码安全审计专家,负责审查代码漏洞。

调用时:

1. 识别安全敏感代码路径
2. 检查常见漏洞(注入、XSS、身份验证绕过)
3. 验证密钥未被硬编码
4. 审查输入验证和清理

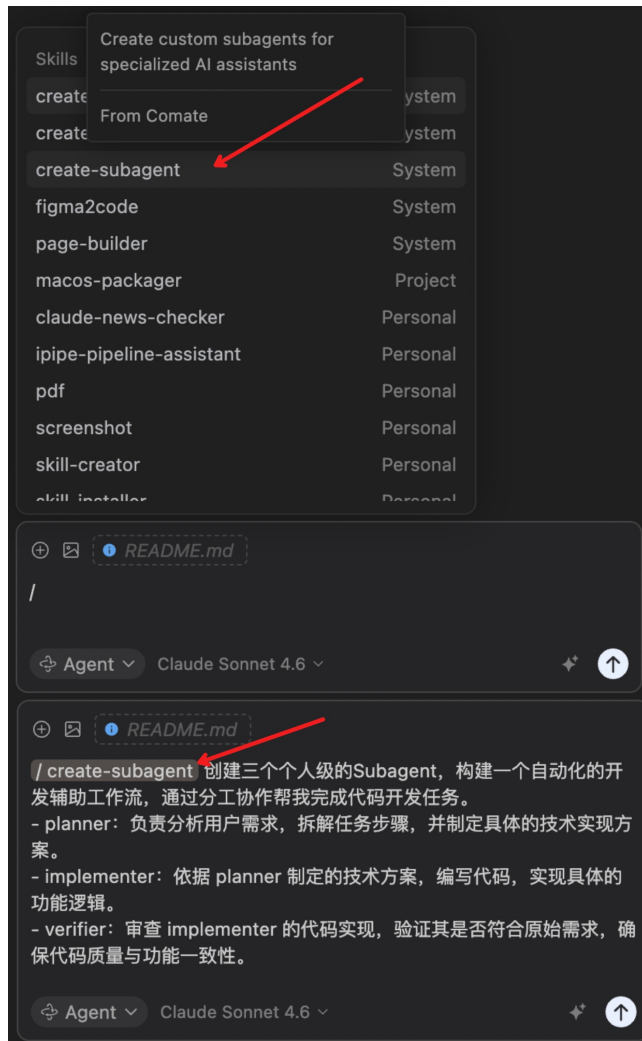
按严重程度报告发现:

- Critical(部署前必须修复)
- High(尽快修复)
- Medium(尽可能处理)

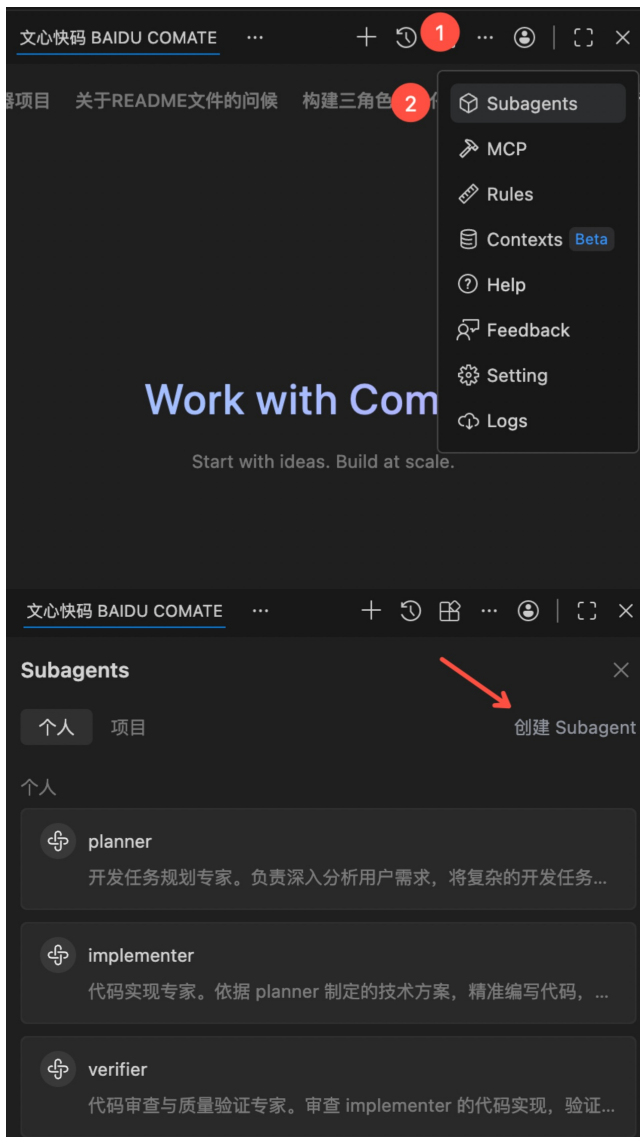
**配置字段** |字段|必填|说明| |---| |name|是|唯一标识符。使用小写字母和连字符。默认为不带扩展名的文件名。| |description|是|何时使用此Subagent。Agent会读取此字段以决定是否进行任务委派。| |tools|否|Subagent可以使用哪些工具。|

**创建Subagent** Comate提供两种创建Agent的方式 :

- 方式一：基于内置的/create-subagentskills快速创建



- 方式二：基于可视化页面创建



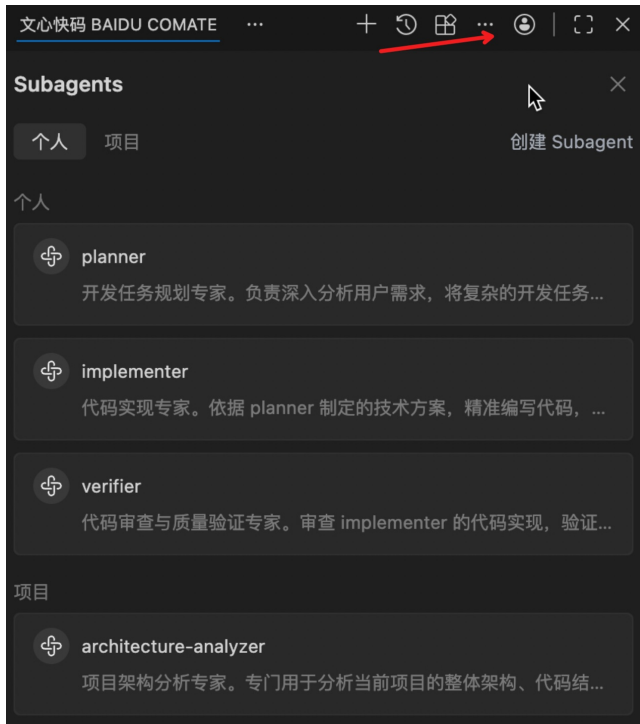
推荐使用方式一，更加便捷和高效。

查看Subagent Comate提供了两种查看Agent的方式：

- 方式一：直接问Agent让它帮助整理，在输入框输入

当前有哪些生效的Subagents，整理出来

方式二：基于可视化页面查看



## 🔗 使用Subagent

**自动委派** Agent 会根据以下因素主动委派任务：

- 任务的复杂度和范围
- 项目中自定义的 subagent 描述
- 当前上下文和可用工具

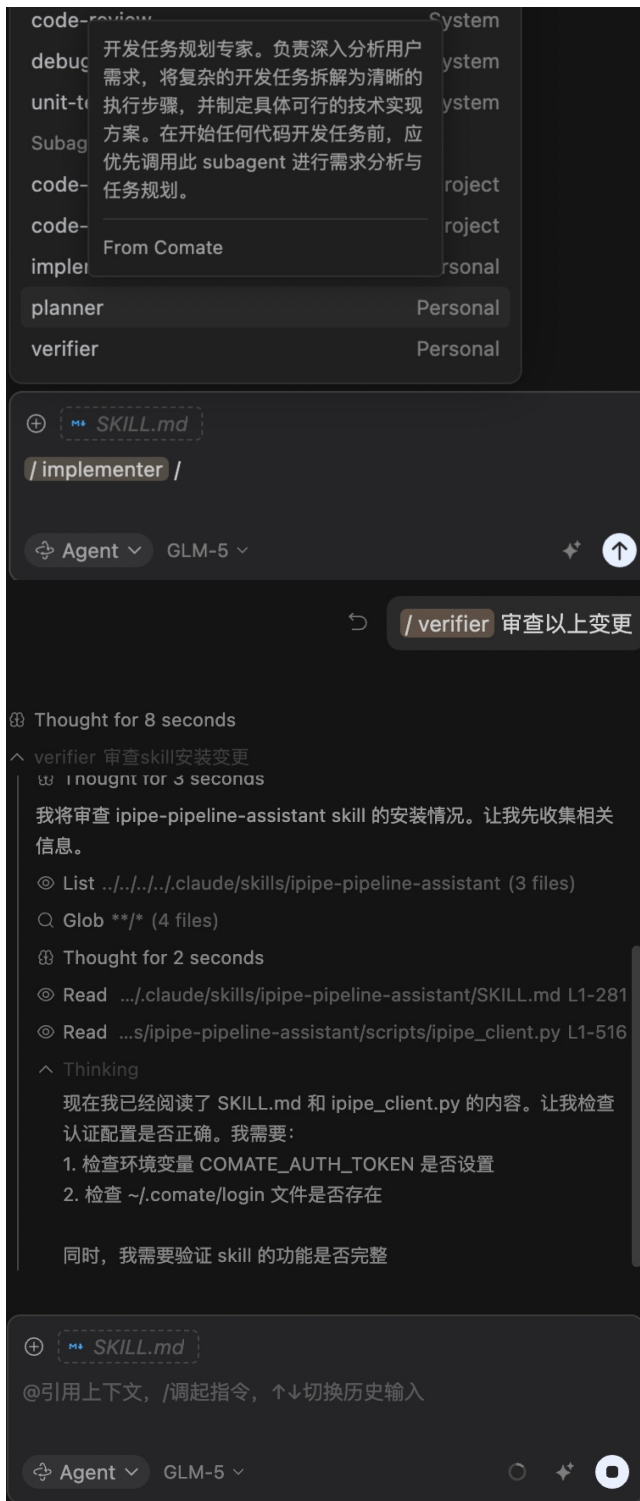
在 `description` 字段中加入诸如 “use proactively” 或 “always use for” 之类的短语，可以鼓励自动委派。

**显式调用** 在提示词中使用 `/{subagent-name}` 语法显式调用指定的子代理：

```
> /verifier confirm the auth flow is complete
> /debugger investigate this error
> /security-auditor review the payment module
```

你也可以在对话中自然提及它们来调用子代理：

```
> 使用 verifier subagent 来确认身份验证流程已完成
> Have the debugger subagent investigate this error
> Run the security-auditor subagent on the payment module
```



并行执行 Comate也支持同时调用多个Subagent，以获得最大吞吐量，Agent 会在单条消息中发出多个 Subagent工具调用，因此Subagent可以并行运行。

建议使用以下Prompt唤起并行调用，采用此种格式并非Agent能力无法主动识别，而是Agent天然习惯串行执行任务。

```
并行分析xxxxx，实现xxxxx
```

分析Write、Bash、Edit工具...

↶ 并行分析Write、Bash、Edit三个工具的实现逻辑，尤其是如何处理异常情况

⌚ Thought for 2 seconds

我将并行分析 Write、Bash、Edit 三个工具的实现逻辑。让我启动三个探索任务：

^ Explore 分析 Write 工具实现 **1**

分析 Write 工具的实现逻辑，重点关注：1. 工具的主要功能和参数 2. 如何处理异常情况（文件不存在、权限问题、写入失败等） 3. 错误处理和返回机制 4. 请找到相关的源代码文件并详细说明实现细节

⌚ Thought for 3 seconds

我来帮你分析 Write 工具的实现逻辑。让我先搜索相关的源代码文件。

Q CodebaseSearch ...plementation file writing error handling (20 files)

Q Glob \*\*/write\*.ts (2 files)

^ Thinking

很好

^ Explore 分析 Bash 工具实现 **2**

分析 Bash 工具的实现逻辑，重点关注：1. 工具的主要功能和参数 2. 如何处理异常情况（命令执行失败、超时、权限问题等） 3. 错误处理和返回机制 4. 安全性考虑（如命令注入防护） 5. 请找到相关的源代码文件并详细说明实现细节

⌚ Thought for 3 seconds

我将帮您分析 Bash 工具的实现逻辑。让我先在代码库中搜索相关的代码。

Q CodebaseSearch ...I implemented for command execution? (20 files)

Q CodebaseSearch Where is bash command execution handled?

^ Explore 分析 Edit 工具实现 **3**

分析 Edit 工具的实现逻辑，重点关注：1. 工具的主要功能和参数 2. 如何处理异常情况（字符串未找到、多次匹配、文件修改冲突等） 3. 错误处理和返回机制 4. 请找到相关的源代码文件并详细说明实现细节

⊕

@引用上下文, /调起指令, ↑↓切换历史输入

Agent Auto

↩ 并行分析Write、Bash、Edit三个工具的实现逻辑，尤其是如何处理异常情况

⌚ Thought for 2 seconds

我将并行分析 Write、Bash、Edit 三个工具的实现逻辑。让我启动三个探索任务：

- 🔍 Explore 分析 Write 工具实现
- 🔍 Explore 分析 Bash 工具实现
- 🔍 Explore 分析 Edit 工具实现

⌚ Thought for a few seconds

### Write、Bash、Edit 工具实现分析报告

---

#### 1. Write 工具 ( `src/tool/write.ts` )

**核心功能：** 向文件系统写入文件内容，覆盖已存在的文件

**异常处理机制：**

异常类型	处理方式
相对路径	自动转换为绝对路径 (拼接项目目录)
外部目录	通过 <code>ctx.ask()</code> 请求用户授权
文件未先读取	<code>FileTime.assert()</code> 抛出错误，强制先 Read
并发修改冲突	检查 <code>mtime</code> ，若文件被修改则要求重新读取
目录不存在	捕获 <code>ENOENT</code> ，自动创建父目录后重试
权限拒绝	直接抛出错误，由上层处理

**安全设计：**

⊕

|@引用上下文, /调起指令, ↑↓切换历史输入

🔄 Agent ▾ Auto ▾

○ ✎ ⬆



## 🔄 复用Subagent

Comate还支持复用历史会话中的Subagent，如一个检索的任务做的还不够完整，可以继续复用此会话，\*\*这对跨越多次调用的长时间运行任务非常有用。\*\*

每次Subagent执行都会有一个独立的ID，主Agent通过此ID即可在保留完整上下文的情况下恢复该Subagent (*您不需要关注任何ID的存在，以上都是由主Agent自动获取并唤起「沉睡」的Subagent*)。可以尝试用以下Prompt格式来复用Subagent：

在原来的 {描述希望复用的Subagent任务名字} Task基础上继续分析：{描述希望Subagent继续做什么}

👉 并行分析Write、Bash、Edit三个工具的实现逻辑，尤其是如何处理异常情况

🕒 Thought for 2 seconds

我将并行分析 Write、Bash、Edit 三个工具的实现逻辑。让我启动三个探索任务：

👉 Explore 分析 Write 工具实现

分析 Write 工具的实现逻辑，重点关注：1. 工具的主要功能和参数 2. 如何处理异常情况（文件不存在、权限问题、写入失败等） 3. 错误处理和返回机制 4. 请找到相关的源代码文件并详细说明实现细节

🕒 Thought for 3 seconds

我来帮你分析 Write 工具的实现逻辑。让我先搜索相关的源代码文件。

🔍 CodebaseSearch ...plementation file writing error handling (20 files)

🔍 Glob \*\*/write\*.ts (2 files)

🕒 Thought for 2 seconds

📖 Read packages/opencode/src/tool/write.ts L1-85

🕒 Thought for a few seconds

📖 Read packages/opencode/src/tool/write.txt L1-9

📖 Read packages/opencode/src/file/time.ts L1-72

📖 Read packages/opencode/src/util/filesystem.ts L1-190

👉 Explore 分析 Bash 工具实现

👉 Explore 分析 Edit 工具实现

🕒 Thought for a few seconds

### Write、Bash、Edit 工具实现分析报告

👉 Write工具分析的不全面，在原来的task基础上继续分析：LSP的分析会不会导致工具响应的缓慢？

🕒 Thought for 4 seconds

👉 Explore 分析 LSP 对 Write 响应的影响

继续分析 Write 工具中的 LSP 相关逻辑，重点关注：1. LSP.touchFile 和 LSP.diagnostics 的具体实现 2. LSP 诊断是否会阻塞 Write 工具的响应返回？ 3. LSP 分析的超时机制或性能优化 4. 是否有异步处理或后台执行的机制 5. 查看 LSP 相关源码文件，分析其对工具响应时间的影响

🕒 Thought for 3 seconds

🔍 Grep LSP\(touchFile|diagnostics) (6 files)

🔍 Glob \*\*/Write\*.ts

🕒 Thought for a few seconds

📖 Read packages/opencode/src/tool/write.ts L1-85

📖 Read packages/opencode/src/tool/lsp.ts L1-98

🕒 Thought for a few seconds

🔍 Glob \*\*/lsp.ts (2 files)

🕒 Thought for a few seconds

⊕

@引用上下文，/调起指令，↑↓切换历史输入

🔄 Agent Auto



## 最佳实践

- 编写职责单一的Subagent — 每个Subagent 都应只有一个清晰的职责。避免泛用型的「辅助 (helper)」代理。
- 投入精力写好描述 — `description` 字段决定 Agent 何时将任务委派给你的Subagent。花时间打磨它。通过编写提示并检查是否触发了正确的Subagent 来进行测试。
- 保持提示简洁 — 冗长、散乱的提示会削弱聚焦度。要具体、直接。
- 将Subagent 纳入版本控制 — 把 `.comate/agents/` 提交到你的代码库, 让整个团队都能受益。
- 从 Agent 生成的代理开始 — 先让 Agent 帮你起草初始配置, 然后再进行自定义。

不要创建几十个泛泛的子 Agent: 拥有 50+ 个只写着“帮助写代码”之类模糊说明的 Subagent 是低效的。Agent 不会知道什么时候该用它们, 你也会在维护它们上浪费时间。

- 描述过于模糊 — “用于通用任务”不会给 Agent 任何关于何时委托的信号。请具体一些: 例如“在实现基于 OAuth 提供商的认证流程时使用”。
- 提示词过于冗长 — 2,000 字的 Prompt 并不会让Subagent 更聪明, 只会让它更慢、更难维护。
- 重复Skills、Commands的功能 — 如果任务是单一用途且不需要上下文隔离, 请改用Skills或 Command。
- Subagent 过多 — 先从 2-3 个聚焦的Subagent 开始。只有在出现清晰且彼此区分的使用场景时再增加。

## 上下文

### Skills

#### 什么是Skills?

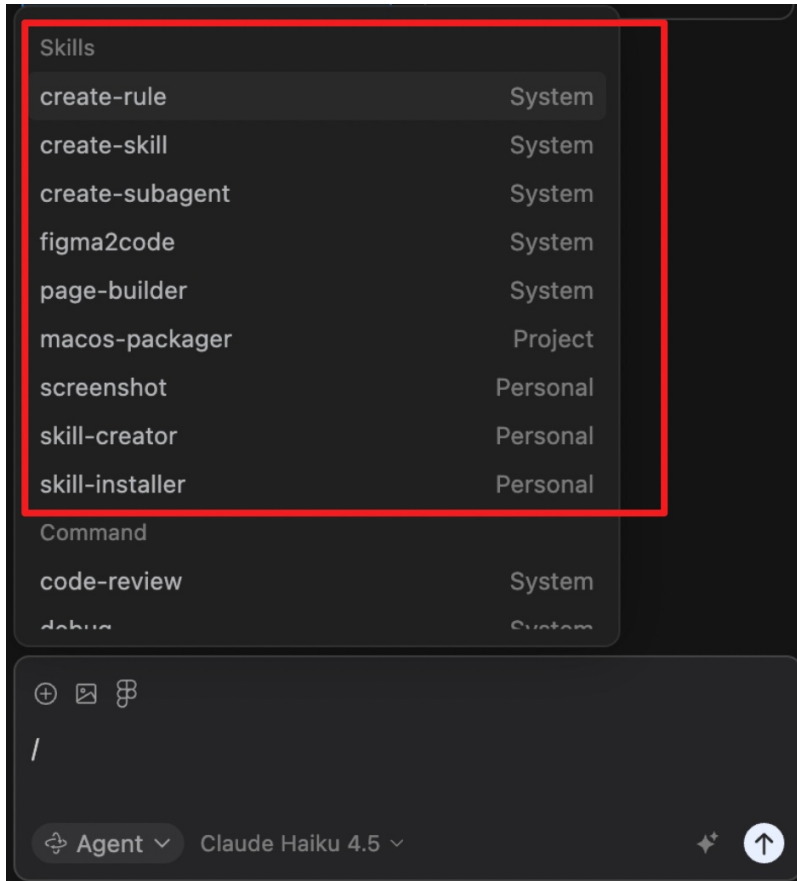
Skills是一个可移植、支持版本控制的包, 用于让 Agent 学会如何执行特定领域的任务。技能可以包含脚本、模板和参考资料, Agent 可以使用其工具对这些内容进行操作。

- **可移植**：Skills适用于任何支持 Agent Skills 标准的 Agent。
- **受版本控制**：Skills以文件形式存储，可以在你的代码仓库中追踪其变更
- **可操作**：Skills可以包含脚本、模板和参考资料，Agent 使用其工具对这些内容进行处理。
- **渐进式**：Skills按需加载资源，使上下文使用更加高效。

## 🔗 Skills的工作原理

Comate 启动时，会自动从Skills目录中发现并加载Skill，并将它们提供给 Agent 使用。Agent 会看到所有可用Skills，并根据当前上下文决定何时调用它们。

你也可以在 Agent 对话中输入 `/` 并搜索技能名称来手动调用技能。



## 🔗 Skills目录

技能会自动从以下位置自动加载：

位置	作用域
<code>.agents/skills/</code>	项目级
<code>.comate/skills/</code>	项目级
<code>~/comate/skills/</code>	用户级（全局）

Comate提供的 `create-rule`、`create-skill`、`create-subagen` 三个系统级的Skills也在 `~/comate/skills/` 目录下。

每个技能应为一个包含 `SKILL.md` 文件的文件夹：

```
.comate/
├── skills/
│   └── my-skill/
│       └── SKILL.md
```

技能还可以包含脚本、参考文件和资源等可选目录：

```
.comate/
├── skills/
│   ├── deploy-app/
│   │   ├── SKILL.md
│   │   ├── scripts/
│   │   │   ├── deploy.sh
│   │   │   └── validate.py
│   │   └── references/
│   │       └── REFERENCE.md
│   └── assets/
│       └── config-template.json
```

## 🔗 SKILL.md 文件格式

每个 Skill 都在带有 YAML 前置信息 (frontmatter) 的 SKILL.md 文件中定义：

```
---
name: my-skill
description: 简要描述此技能的功能及使用时机。
---

##### 我的技能

为 Agent 提供的详细指令。

##### 使用时机

- 在以下情况使用此技能...
- 此技能适用于...

##### 指令

- 为 Agent 提供的分步指导
- 特定领域的约定
- 最佳实践和模式
- 如需向用户澄清需求,请使用提问工具
```

**Frontmatter**字段 |字段|必填|说明| |:-| |name|Yes|技能标识符。仅限小写字母、数字和连字符。必须与父文件夹名称一致|  
|description|Yes|描述技能的作用及其使用场景。由Agent用于判断相关性。|

**在Skills中包含脚本** Skills可以包含 `scripts/` 目录，内含可由代理运行的可执行代码。在 SKILL.md 文件中使用相对于Skills根目录的相对路径引用这些脚本。

```

---
name: deploy-app
description: 将应用部署到预发布或生产环境。在部署代码时使用,或当用户提及部署、发布或环境时使用。
---

##### Deploy App

Deploy the application using the provided scripts.

##### Usage

Run the deployment script: `scripts/deploy.sh <environment>`

Where `<environment>` is either `staging` or `production`.

##### Pre-deployment Validation

Before deploying, run the validation script: `python scripts/validate.py`

```

当Skills被调用时，Agent 会读取这些指令并执行引用的脚本。脚本可以使用任何语言编写，例如 Bash、Python、JavaScript，或 Agent 实现所支持的任何其他可执行格式。

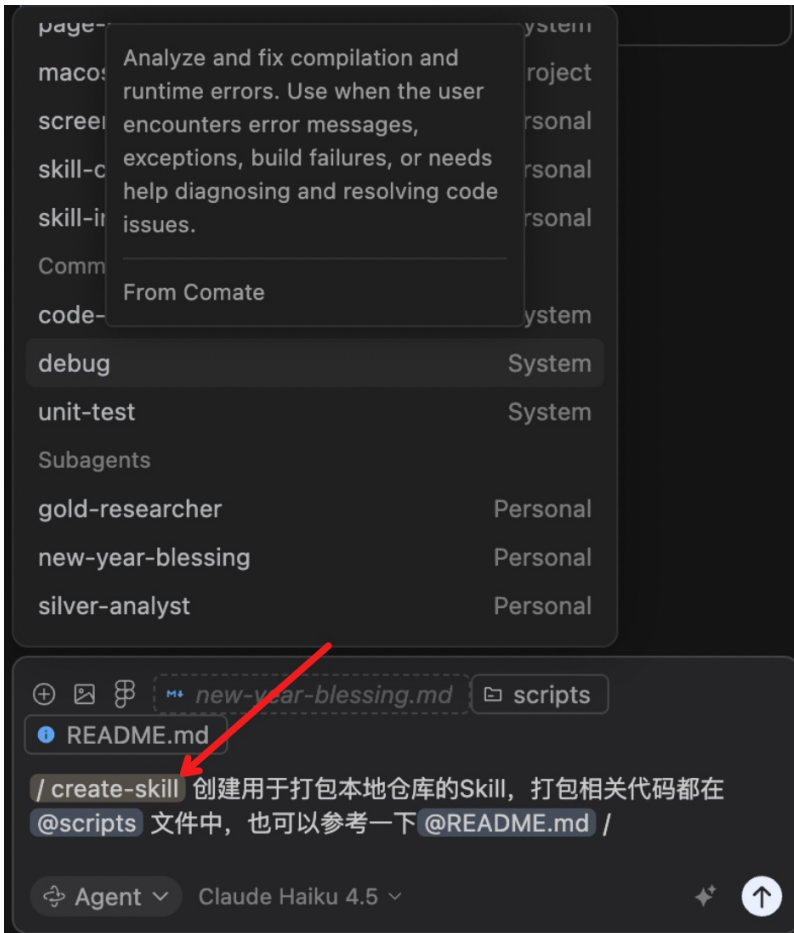
**可选目录** Skills 支持以下可选目录：

目录	描述
scripts/	Agents 可以运行的可执行代码
references/	按需加载的附加文档
assets/	模板、图片或数据文件等静态资源

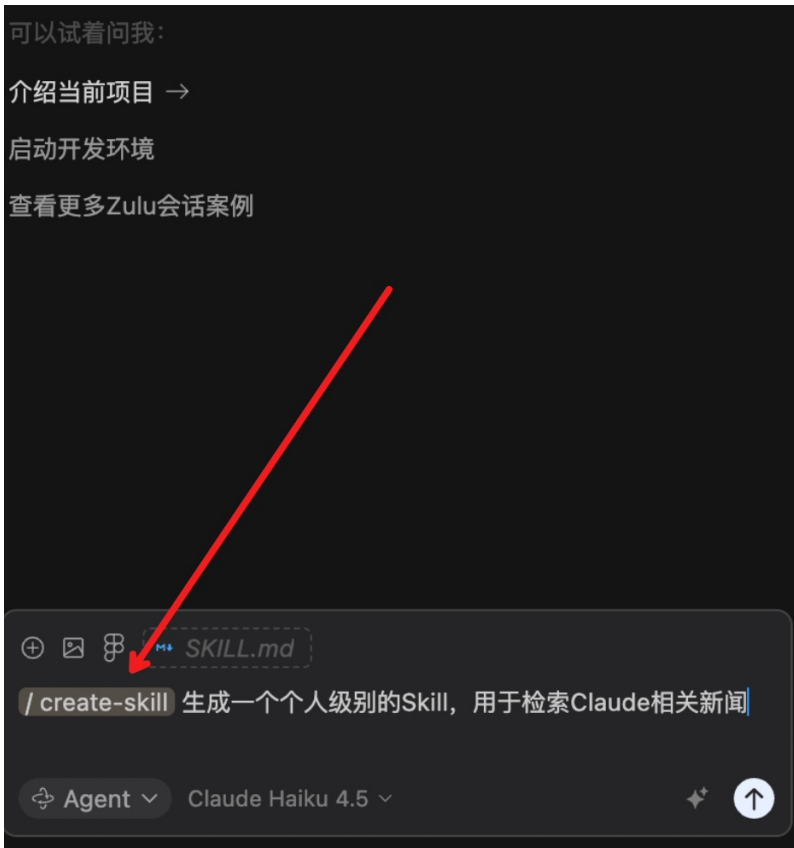
请让主 SKILL.md 文件保持简洁，将详细参考资料放在单独的文件中。这样可以更高效地利用上下文，因为 Agent 会按需逐步加载资源——只在需要时才加载。

## 🔗 导入或生成Skills

基于 `/create-skill` 生成Skills Comate内置了系统级别的Skill `/create-skill`，可以一键创建项目级或个人级别的Skills。

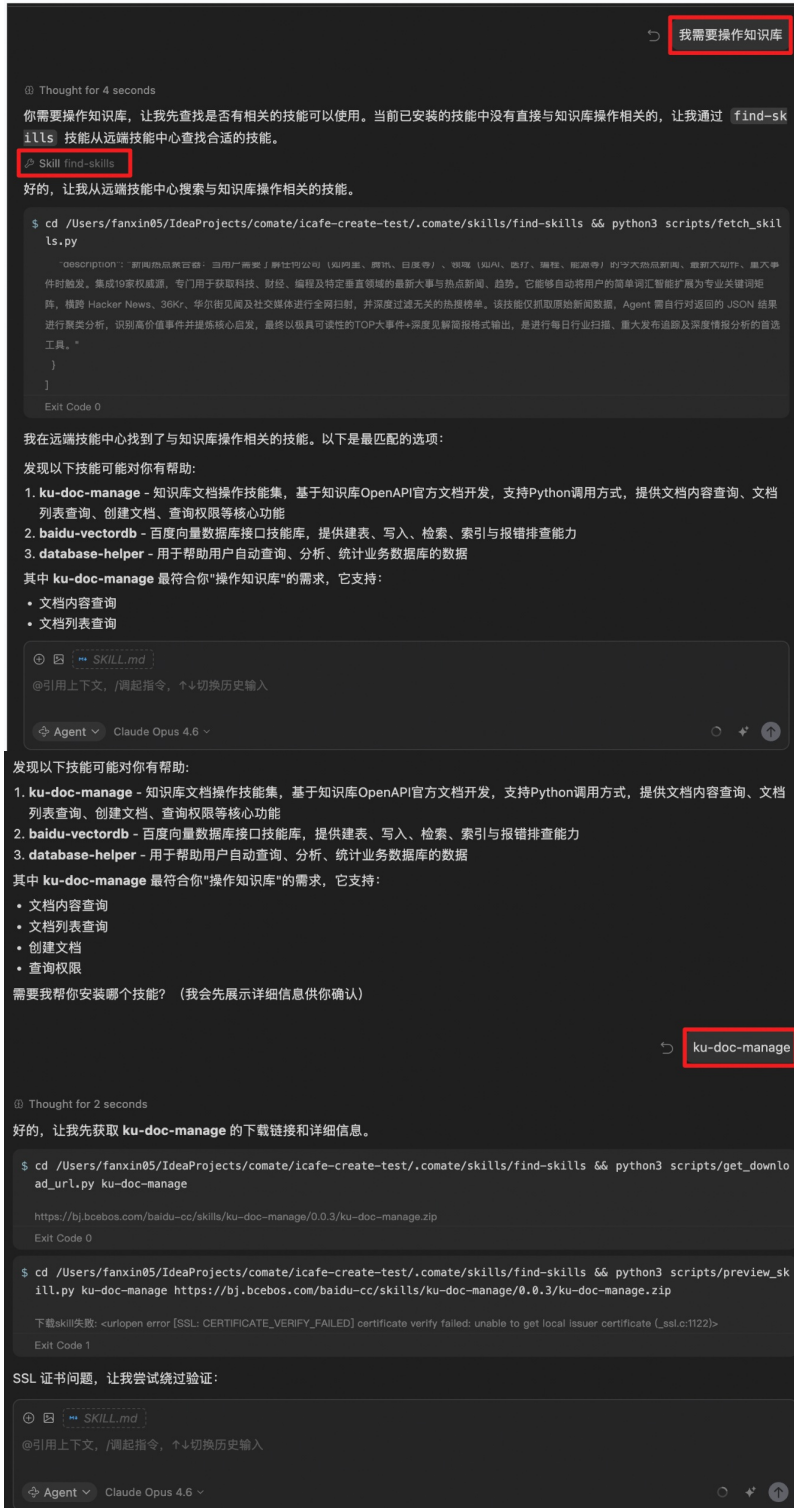


请注意，使用 `/create-skill` 创建Skills的时候，请采用明确的话术告知Agent是创建个人级还是项目级的Skills，会显著提升识别的准确率，例如：



- 创建项目级：`/create-skill`在当前仓库创建xxxxx的Skills，以解决xxxx问题
- 创建个人级：`/create-skill`创建一个个人级的Skill，以解决xxxx问题

基于 `/find-skills` 搜索 Skills Comate 内置了系统级别的 Skill/`find-skills`，当本地已安装的 skills 都不适合处理用户问题时，自动从远端 Skills 中心搜索、推荐并安装最匹配的新 skill，支持安装到当前项目 (`./comate/skills/<skill_name>/`) 或全局个人目录 (`~/./comate/skills/<skill_name>/`)。



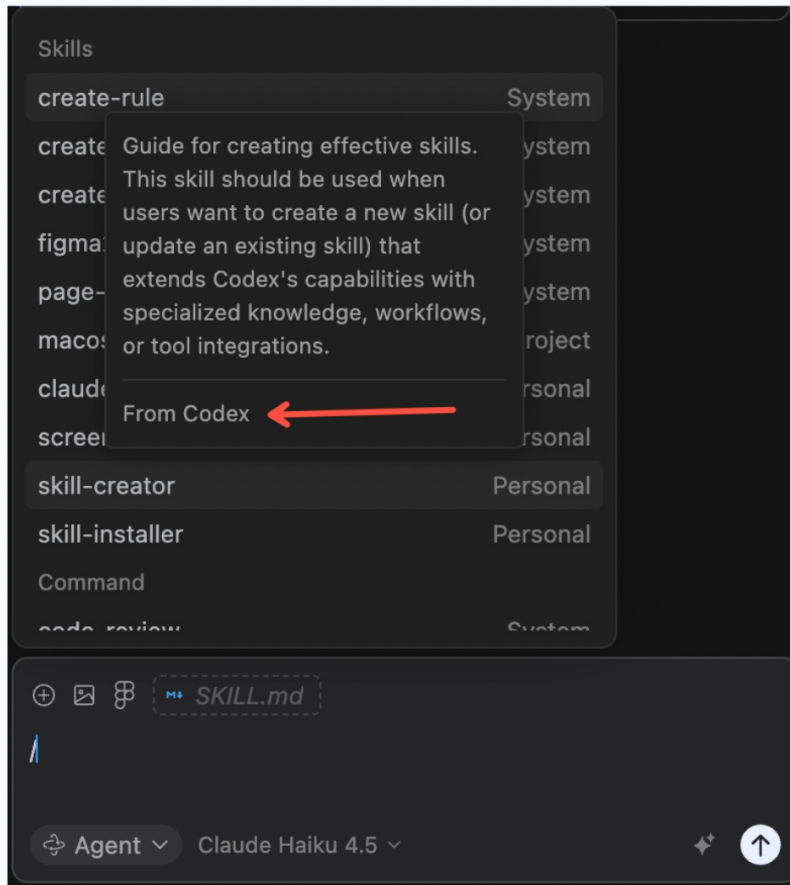
精选了一批超实用、超好玩的 Skill，试试对 Comate 说：

分类	你可以说	Skill	效果
提效狠活	"有没有能用自然语言查数据库的技能？我想查最近7天的异常订单"	database-helper	不用手写SQL，说人话就能查数据
		database-helper	41项检

	"有没有专门做Go代码性能分析的skill？这段代码跑得太慢了"	code-analyze-r	查帮你揪出性能瓶颈
	"有没有能自动清理AB实验代码的skill？实验结束了想把实验分支删干净"	experiment-code-cleanup	自动识别实验分支，干净利落删干净
告别切屏	"有没有skill能直接在Comate里操作iPipe流水线？"	ipipe_pipeline_assistant	不用切浏览器，对话框里直接操作
	"有没有skill能直接在Comate里管理知识库文档？我想查一下这篇文档的内容"	ku-doc-manage	查文档、建文档、管权限，不用切到知识库网页
	"有没有能搜索iAPI接口的skill？我想找用户登录相关的API"	iapi-manager	自然语言搜接口，告别翻文档
	"有没有skill能直接在Comate里发如流消息？想给群里通知今天5点发版"	so-send-message	通知一句话搞定，不用再手动@
信息投喂	"有没有能聚合今天AI热点新闻的skill？"	hot-news-skill-manus	19家信息源聚合，3分钟看完一天热点
	"有没有能搜arXiv论文的skill？想找几篇RAG优化相关的"	arxiv-search	arXiv论文即搜即得，带摘要
整点花活	"有没有能根据PRD自动生成技术方案的skill？"	prd-to-tech-design	产品文档直出技术设计，先搭框架

			再填细节
	"有没有什么skill能在我写代码累的时候给我打打气"	programmer-cheerleader	程序员专属啦啦队，负责给你打气续命

兼容其他Agent工具 Comate 还会从 Claude、Codex、Cursor等20多个Agent工具的目录加载技能，包括但不限于：`~/.claude/skills/`、`~/.codex/skills/`、`~/claude/skills/` 和 `~/codex/skills/`等。加载的外部Agent工具的将会被有效标示出来，在实际使用过程中和Comate原生的Skills没有任何区别。



将Rules迁移到Skills 通过内置的 `/create-skill` 还可以帮助你现有的Rules转换为Skills。迁移步骤如下：

1. 在 Agent 聊天中输入 `/create-skill`
2. 输入迁移的诉求，如：*转换当前仓库的Rules为Skills，你要先识别哪些可以转换并供我确认，我确认后再转换*
3. 在 `~/.comate/skills/` 中查看生成的技能

↩ /create-skill 转换当前仓库的Rules为Skills，你先识别哪些可以转换并供我确认，我确认后再转换

⌘ Thought for 8 seconds

我来帮你识别可以转换的 Rules。首先让我读取这些规则文件的内容。

Q Glob `**/*.mdr`

⌘ Thought for 2 seconds

⊙ List `.comate` (39 files)

⌘ Thought for 7 seconds

我看到 `.comate/rules/` 目录下有很多规则文件，同时 `.comate/skills/` 目录下已经有一些对应的 skills。让我读取这些文件内容来进行对比分析。

⊙ Read `.comate/rules/index.mdr` L1-31

⊙ Read `.comate/rules/f18n.mdr` L1-104

⊙ Read `.comate/rules/agent-tool.mdr` L1-47

⊙ Read `.comate/rules/performance-check-guide.mdr` L1-110

⊙ Read `.comate/rules/zulu-cli.mdr` L1-6

⊙ Read `.comate/rules/kernel-pt-event.mdr` L1-5

⊙ Read `.comate/rules/kernel-storage.mdr` L1-161

⊙ Read `.comate/rules/kernel-performance-log.mdr` L1-7

⌘ Thought for 2 seconds

⊙ Read `.comate/rules/webview-pt-event.mdr` L1-5

⊙ Read `.comate/rules/webview-modal.mdr` L1-92

⊙ Read `.comate/rules/project-agent.mdr` L1-18

⊙ Read `.comate/rules/agent-knowledge-tool.mdr` L1-391

⊙ Read `.comate/rules/agent-checkpoint.mdr` L1-261

⊙ Read `.comate/rules/agent-code-written-metrics.mdr` L1-220

⌘ Thought for 2 seconds

## 建议操作

现有 Skill 目录状态：所有目录都是空的，需要创建实际的 SKILL.md 文件。

推荐转换方案：

### 1. 优先转换（内容丰富，独立性强）：

- `i18n.mdr` → Webview 国际化 Skill
- `agent-knowledge-tool.mdr` → Agent 知识检索 Skill
- `agent-checkpoint.mdr` → Agent Checkpoint Skill
- `agent-code-written-metrics.mdr` → Agent 代码留存指标 Skill
- `kernel-package.mdr` → Kernel 包开发 Skill
- `performance-check-guide.mdr` → 性能代码审查 Skill

### 2. 中等优先级：

- `kernel-storage.mdr` → Kernel 存储系统 Skill
- `webview-modal.mdr` → Webview 弹窗 Skill
- `webview-package.mdr` → Webview 包规范 Skill
- `vscode-package.mdr` → 需新建 Skill 目录
- `style-refactor.mdr` → 代码风格指南 Skill

### 3. 可选转换：

- `agent-tool.mdr` → Agent 工具开发 Skill
- `project-agent.mdr` → 项目级 Agent Skill
- `kernel-performance-log.mdr` → 可合并到 `kernel-package` 或独立

### 4. 不建议转换（内容几乎为空）：

- `zulu-cli.mdr`
- `kernel-pt-event.mdr`
- `webview-pt-event.mdr`
- `webview-pt-api.mdr`

请确认你希望转换哪些规则？你可以：

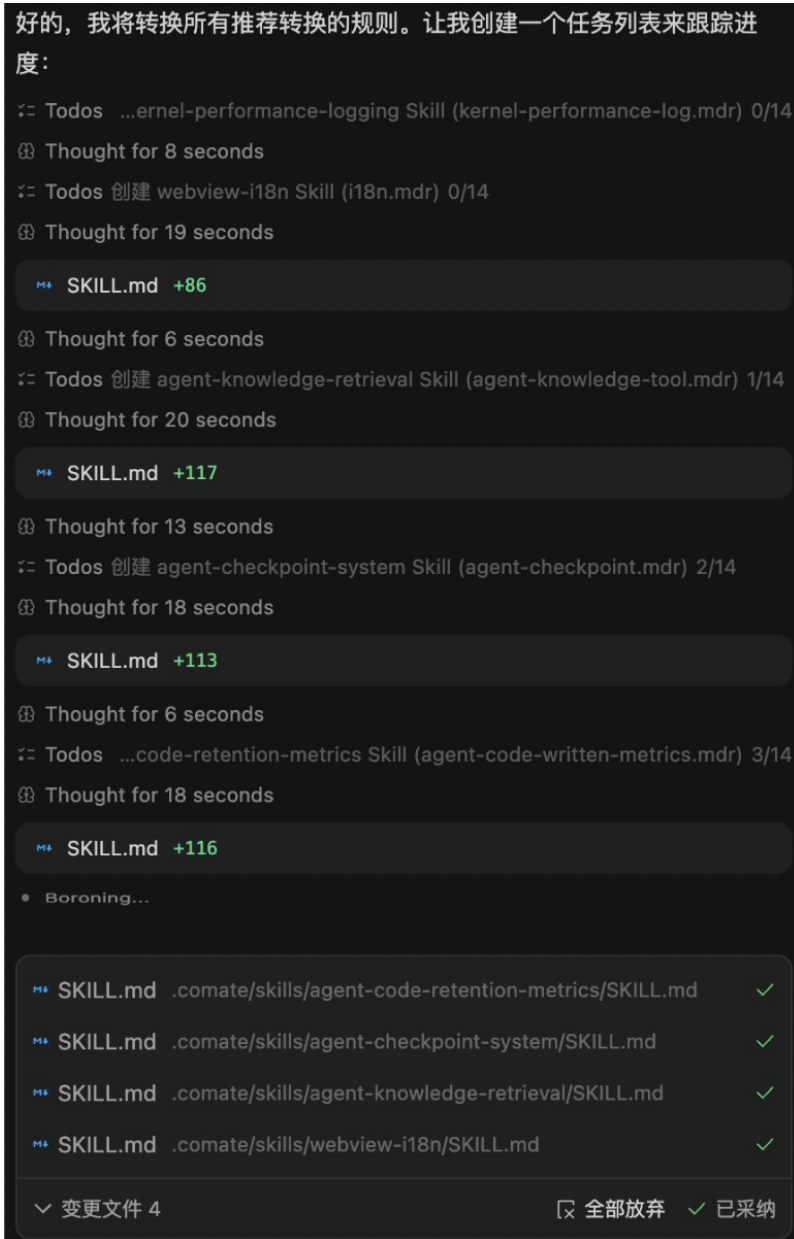
1. 转换所有推荐转换的规则
2. 选择特定的规则进行转换
3. 提出其他调整建议

⊕ `mcp.json`

@引用上下文，/调起指令，↑↓切换历史输入

🔗 Agent ▾ GLM-5.0 ▾





- 具有 `alwaysApply: true` 或特定 `globs` 模式的规则一般不会被迁移（这并非强制），因为它们有与技能行为不同的显式触发条件。
- 用户规则也不会被迁移，因为它们不存储在文件系统中。

## Rules

规则（Rules）功能支持用户使用自然语言指令来规范 Agent 的行为，帮助 Agent 更精准地理解并适应您的编码偏好，例如理解项目框架和代码风格等。规则仅对当前项目生效，存储在 `.comate/rules` 目录下。

### 🔗 规则的用处

Agent 在使用大型语言模型进行对话、代码生成等过程中不会保留记忆。规则功能可将用户设定好的遵循信息作为上下文，让大语言模型持续记住这些信息以及使用需要遵循的规范。

### 🔗 规则结构

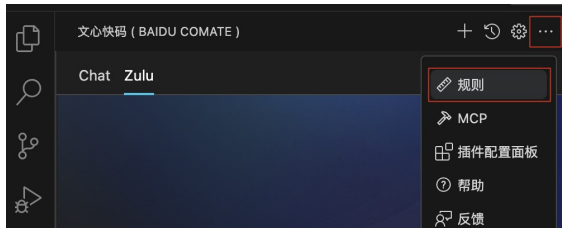
每个规则文件均以 `.mdr` 的格式储存，支持将生效信息和规则内容放在一个文件中。规则支持以下三种生效方式：

生效方式	描述
始终生效	此规则将在始终包含在模型上下文中，无需您的持续关注
指定文件	此规则将在对话中包含特定文件时生效，特定文件的判断包含之前的对话或当前会话中已添加文件，支持通配符（例如 *.py 或 client/*/*.tsx）
手动生效	此规则仅在明确提及使用时才生效（在对话中#rulename）

## 🔗 规则设定

### 新建规则

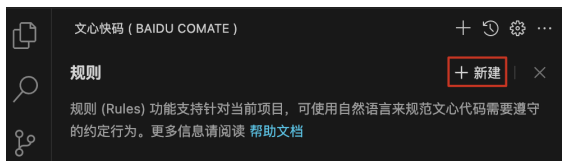
1. 进入规则设定页面：通过文心快码插件右上角进入更多选项，点击 **规则** 进入规则设定页面。



您还可以从设置中查看所有规则及其状态的列表。



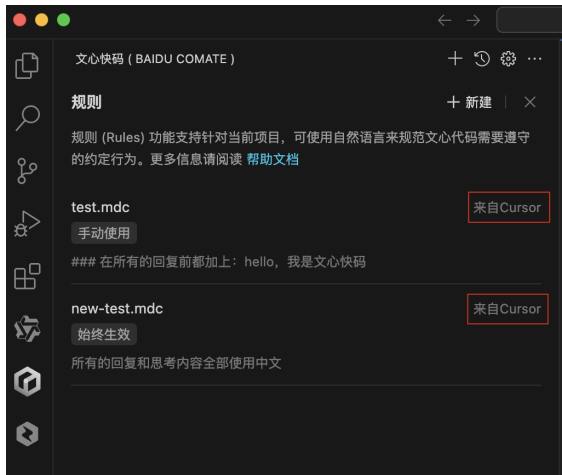
2. 新建规则：进入规则设定页面后，点击右上角 **+ 新建** 按钮，添加新的规则



3. 直接新建文件：您也可以在.comate/rules路径下直接创建格式为\*.mdr的文件。
4. 编写规则内容：在 \*.mdr 的规则文件中，使用清晰、明确的自然语言（通常是 Markdown 格式）编写您希望 AI 在处理此项目时必须遵守的指令、约束或偏好设置。编写后记得要将文件进行保存。

**多Rules** 您可以在规则设定页面，根据需求设定不同的规则和生效方式。Comate Agent 在后续与该项目相关的交互中，会自动加载并遵循您设定的规则。

**兼容Cursor Rules** Agent 可自动获取您在Cursor中设定的规则，无需在文心快码中重复添加（Agent 无法通过图形化界面编辑Cursor规则的生效范围，可通过文件编辑器修改）



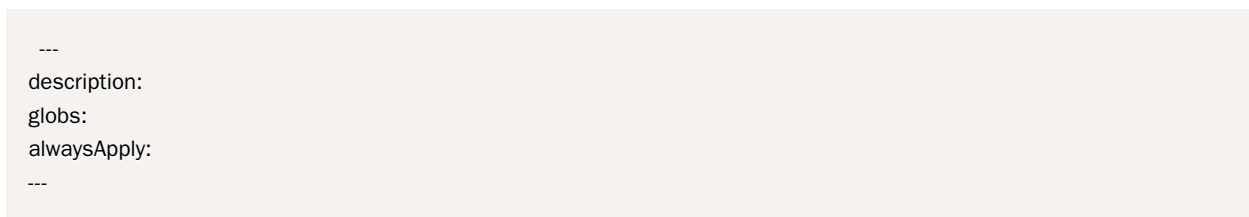
### 仅在本地生效

如果您希望规则仅在您自己的本地开发环境生效，不想通过版本控制分享给团队其他成员（例如，一些个人编码习惯或临时性的指令），您可以将 `.comate/rules/` 文件路径添加到您项目的目录文件中。这样，Git 等版本控制系统就会忽略这些文件的变动。

**注意事项** 生效范围的修改建议从规则列表中点击 **编辑** 进行变更

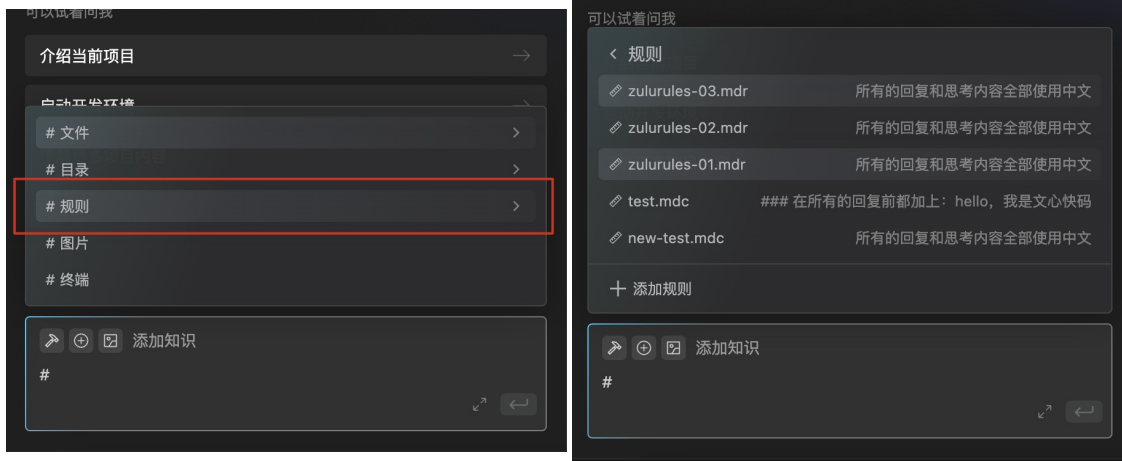


不建议直接通过修改文件`---`之间的内容进行变更，否则可能会导致系统无法正确识别。



### 🔗 规则使用

- 对于始终生效、指定文件两种生效范围的规则，您无需过多关注。
  - 始终生效** 会在您的所有会话中自动生效
  - 指定文件** 会在对话中包含特定文件上下文或当前会话中已添加文件进行生效。指定文件支持**通配符+文件路径/文件格式**（例如 `*.py` 或 `client/*/*.tsx`）
- 对于手动生效的规则，您在需要用到此规则时，在对话框中**#此规则**进行使用。



3. 通过手动#的方式，您可以选择设定的所有规则。其是否在会话中生效，取决于规则具体的设定内容。例如规则指定文件 \*.py，但当前对话上下文中未包含 \*.py 的文件，则此规则依旧不会生效。

## 🔗 应用场景

在规则中定义项目框架和开发指南一种典型应用场景。你可以让 Agent 严格遵循开发指南来新增功能。比如在规范的目录生成新代码文件，在统一的接口方法中注册新方法，添加新功能同时生成测试代码。如果没有这样的规范，就无法约束 AI 在生成代码时的随机性，可能生成错误的代码。

以下是一个规则的例子。它定义了示例项目 `async-iterator` 的开发规范。

```
---
description:
globs:
alwaysApply:
---
```

### ##### 项目概述

这是一个用于处理异步迭代的工具库，提供了一系列用于处理异步数据流的工具和操作符。该项目主要解决了在 JavaScript/TypeScript 中处理异步数据流的复杂性问题，提供了简单易用的 API。

### ##### 项目架构

项目采用模块化设计，主要分为以下几个核心部分：

- \*\*核心控制器\*\*** (`controller/``)
  - 实现了 `AsyncIteratorController`` 类
  - 负责管理异步数据流的状态和控制
  - 提供了数据的输入输出接口
- \*\*工厂模块\*\*** (`factory/``)
  - `array.ts``: 处理数组的异步迭代
  - `event.ts``: 处理事件源的异步迭代
  - `reader.ts``: 处理流读取器的异步迭代
  - `stream.ts``: 处理 Node.js 流的异步迭代
- \*\*辅助工具\*\*** (`helper/``)
  - 提供了一系列操作符来处理异步迭代器
  - 实现了链式调用 API
  - 包含多个独立的操作符实现
- \*\*操作符\*\*** (`helper/operators/``)
  - `chunk.ts``: 数据分块
  - `debounce.ts``: 防抖处理
  - `filter.ts``: 数据过滤
  - `map.ts``: 数据转换
  - `take.ts``: 取前 N 个元素

- `until.ts`: 条件终止

#### ##### 代码结构说明

##### \*\*1. 核心接口\*\*

typescript

```
interface OverAsyncIterator<T> extends AsyncIterable<T> {
  filter(predicate: Predicate<T>): OverAsyncIterator<T>;
  map<R>(transform: (value: T) => R): OverAsyncIterator<R>;
  chunk(size: number): OverAsyncIterator<T[]>;
  debounce(ms: number): OverAsyncIterator<T[]>;
  take(count: number): OverAsyncIterator<T>;
  until(predicate: Predicate<T>): OverAsyncIterator<T>;
}
```

##### \*\*2. 工厂函数\*\*

项目提供了多个工厂函数来创建异步迭代器：

- `fromIntervalEach`: 从数组创建带间隔的异步迭代器
- `fromEvent`: 从事件源创建异步迭代器
- `fromStreamReader`: 从流读取器创建异步迭代器
- `fromStream`: 从Node.js流创建异步迭代器

#### ##### 开发指南

##### \*\*1. 基本开发流程\*\*

1. 如需增加运算符，放置在`src/helper/operators`目录下。
2. 所有运算符都要从`src/helper/index.ts`导出，并能够链式调用。
3. 如果增加对流的消费、创建流的方式，放置在`src/factory`目录下。

##### \*\*2. 扩展开发指南\*\*

###### 1. \*\*添加新的工厂函数\*\*

- 在`factory/`目录下创建新文件
- 实现相应的工厂函数
- 在`index.ts`中导出

###### 2. \*\*添加新的操作符\*\*

- 在`helper/operators/`目录下创建新文件
- 实现操作符函数
- 在`helper/index.ts`中添加到`OverAsyncIterator`接口
- 在`over`函数中实现新方法

##### \*\*3. 测试规范\*\*

- 所有功能都需要编写对应的单元测试
- 测试文件放置在相应模块的`\_\_tests\_\_`目录下
- 测试文件命名规则：`[功能名].test.ts`
- 增加任何功能，都要运行单元测试，确保测试的行覆盖率在100%，分支覆盖率大于90%。

#### ##### 代码风格指南

##### 1. \*\*类型定义\*\*

- 优先使用接口（interface）定义类型
- 为所有公开API提供完整的类型定义
- 使用泛型增加代码复用性

##### 2. \*\*错误处理\*\*

- 统一使用 Error 对象处理错误
- 在异步操作中正确处理错误传播

3. \*\*命名规范\*\*

- 使用驼峰命名法
- 类名使用大驼峰
- 接口名清晰表达其用途

##### 注意事项

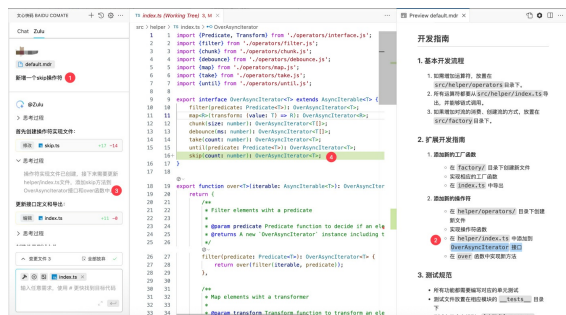
1. 所有的异步操作都应该正确处理错误情况
2. 注意内存泄漏，特别是在处理事件监听器时
3. 在实现新功能时，确保与现有的接口保持一致
4. 保持代码的可测试性，避免副作用
5. 文档注释要清晰完整，包含参数说明和使用示例

这个项目是一个处理异步数据流的工具库，默认提供了一些数据操作符，包括：

- `chunk.ts`: 数据分块
- `debounce.ts`: 防抖处理
- `filter.ts`: 数据过滤
- `map.ts`: 数据转换
- `take.ts`: 取前N个元素
- `until.ts`: 条件终止

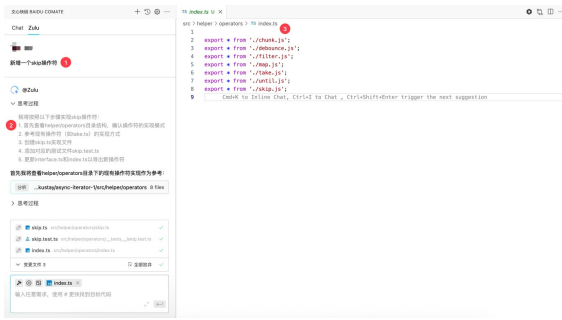
接下来我们为项目新增一个 `skip` 操作符，看看 Agent 如何在规则的约束下生成更高质量的代码。

1. 我们要求 Agent 新增一个 `skip` 操作符
2. 项目配置的规则中，有关于添加新操作符的规范，要求新的操作符必须在 `OverAsyncIterator` 接口中注册
3. Agent 在生成过程中，发现并遵循了这个规范
4. 最后 Agent 按照要求将 `skip` 操作符注册到了 `OverAsyncIterator` 接口，满足要求



如果不定义这个规则，AI 不会严格按照项目规范来生成代码，用户需要花更多时间修正 AI 生成的代码。

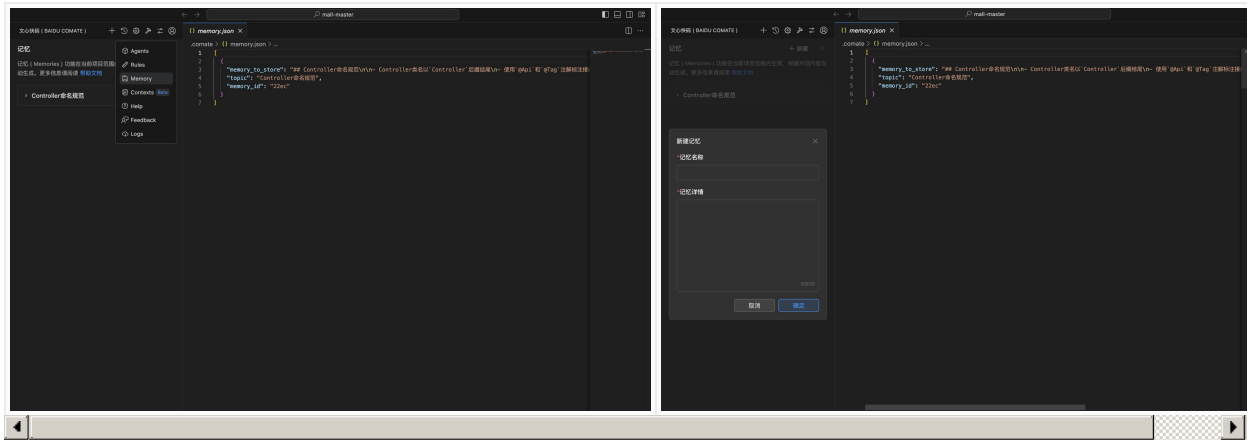
1. 我们要求Agent 新增一个 `skip` 操作符，项目中没有配置规则
2. Agent 在生成代码时，没有发现新的操作符必须在 `OverAsyncIterator` 接口中注册这个要求
3. 最后 AI 按照自己的理解生成一个新的 `index.js` 文件，而没有在现有的 `index.js` 中注册，违反了项目规范



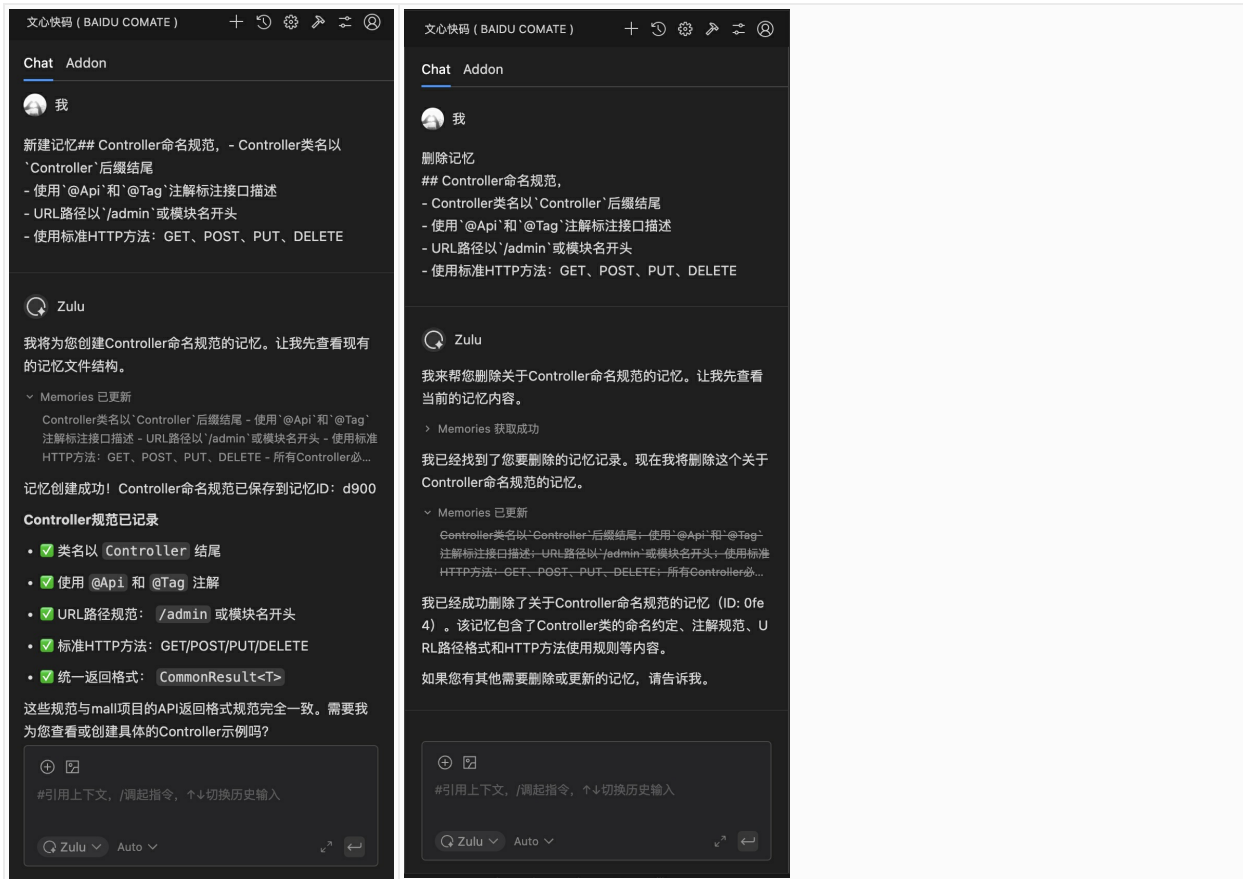
## Memory

记忆 (Memory) 功能, 支持开发者主动新建记忆, 同时在开发者与文心快码的对话过程, 会逐渐形成针对工程项目相关的丰富记忆, 并自动进行相关的记忆整理和更新。记忆仅对当前项目生效, 存储在 .comate 目录下。

新建记忆 支持在Memory记忆设置项中新建记忆, 输入记忆名称和记忆详情, 开发者主动添加项目中所需要的记忆。



自动记忆 当需要文心快码记住某些内容时, 可以在对话窗口中, 针对任意智能体, 直接输入需要记忆的内容, 发送后即可创建相关的记忆。支持新建记忆、更新记忆、删除原有记忆功能



## MCP

### 🔗 MCP功能快速上手

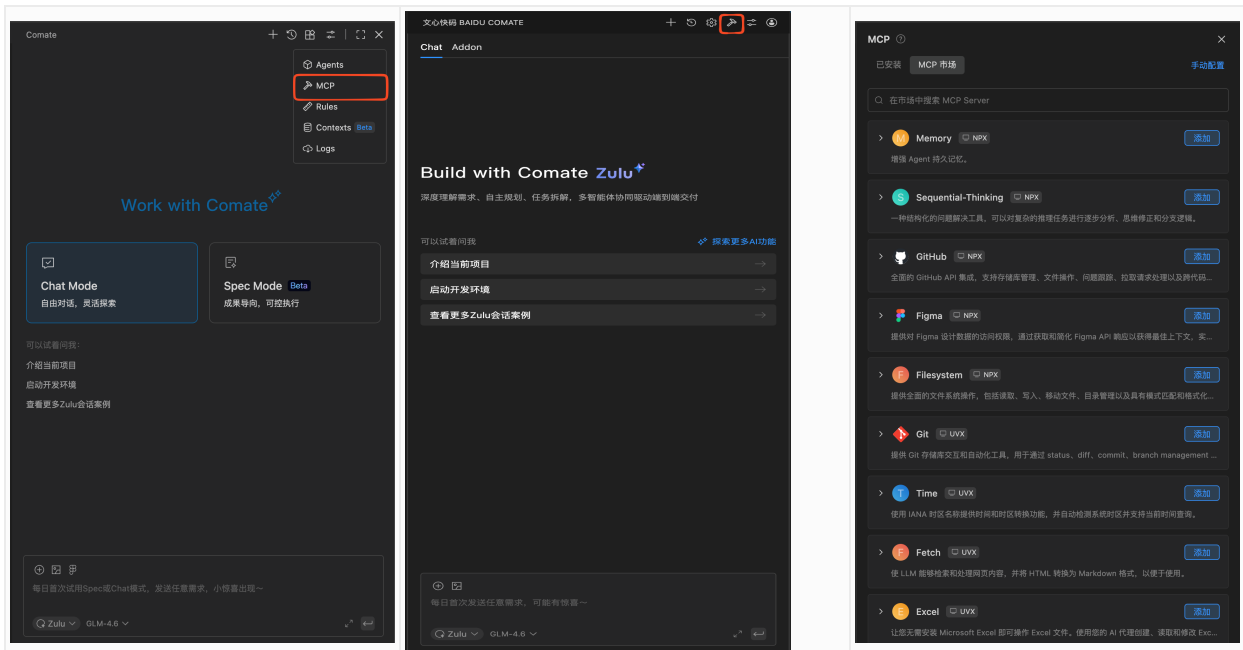
#### 1. 什么是MCP？

MCP（模型上下文协议）是一个开放协议，它允许大型语言模型（LLMs）访问自定义的工具和服务。Comate 作为 MCP 客户端，可以选择向 MCP 服务发起请求，使用它们提供的工具、读取数据源等，从而扩展功能。您可以为 Comate 添加所需的 MCP 服务。

#### 2. 支持的传输类型 Comate 支持三种 MCP 服务的通信模式：

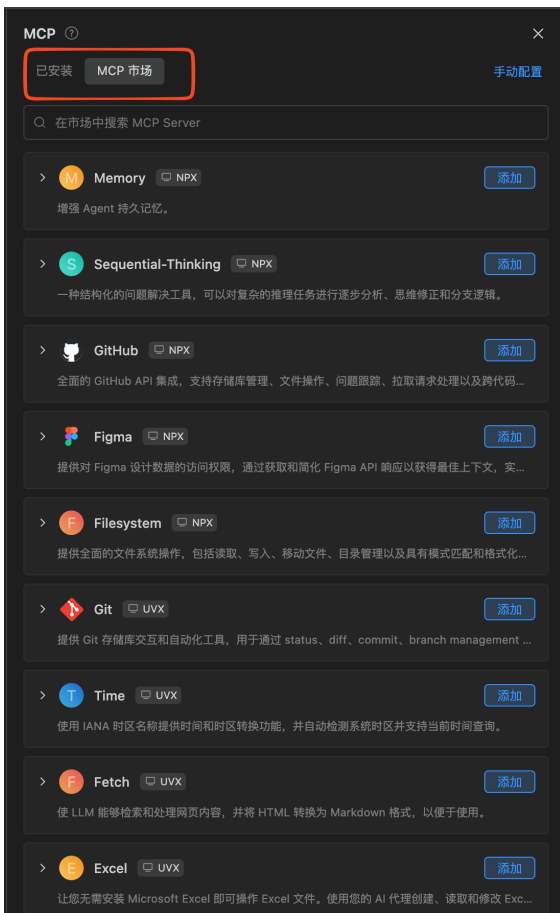
STUDIO 传输	SSE 传输	Streamable HTTP 传输
<p>1.服务运行在本地</p> <p>2.通信通过标准输入输出流进行通信</p>	<p>1.服务运行在本地或远程运行</p> <p>2.通过服务器发送事件协议进行通信 (SSE 传输已经在 2024-11-05 版本废弃, 被 Streamable HTTP 替代, 后者将 SSE 作为可选的流式传输机制纳入其中)</p>	<p>1.服务运行在本地或远程</p> <p>2.采用 HTTP POST 请求实现客户端到服务端的通信, 并可选地通过服务器推送事件 (SSE) 流实现服务端到客户端的通信</p> <p>3.采用 HTTP GET 请求请求实现客户端到服务端的通信, 服务端通过 SSE 实现服务端到客户端通信</p>

**3. 配置MCP服务** Comate 目前支持在项目级配置 MCP 服务 **配置入口** Comate IDE 和插件的入口略有不同, Comate IDE 可以在 Comate 页面, 点击标题栏最右侧的“更多”按钮, 可以看到弹出的下拉框中有 MCP 选项, 若之前没有配置过 MCP, 点击后即可跳转到 MCP 市场。

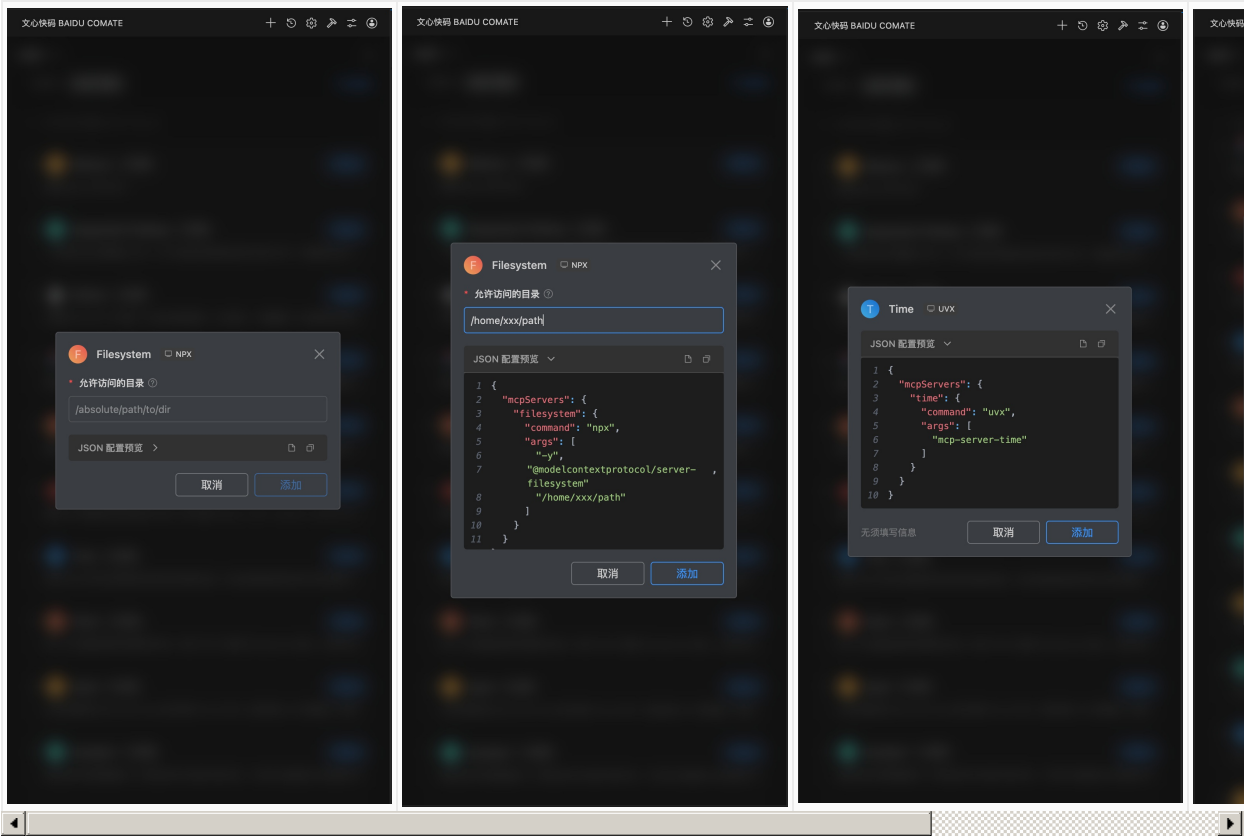


## 添加MCP 方式一：通过 MCP 市场完成添加

1. 进入 MCP 入口后可以看到一个 tab 栏，切换到市场标签页即可见到全部 MCP 服务列表。



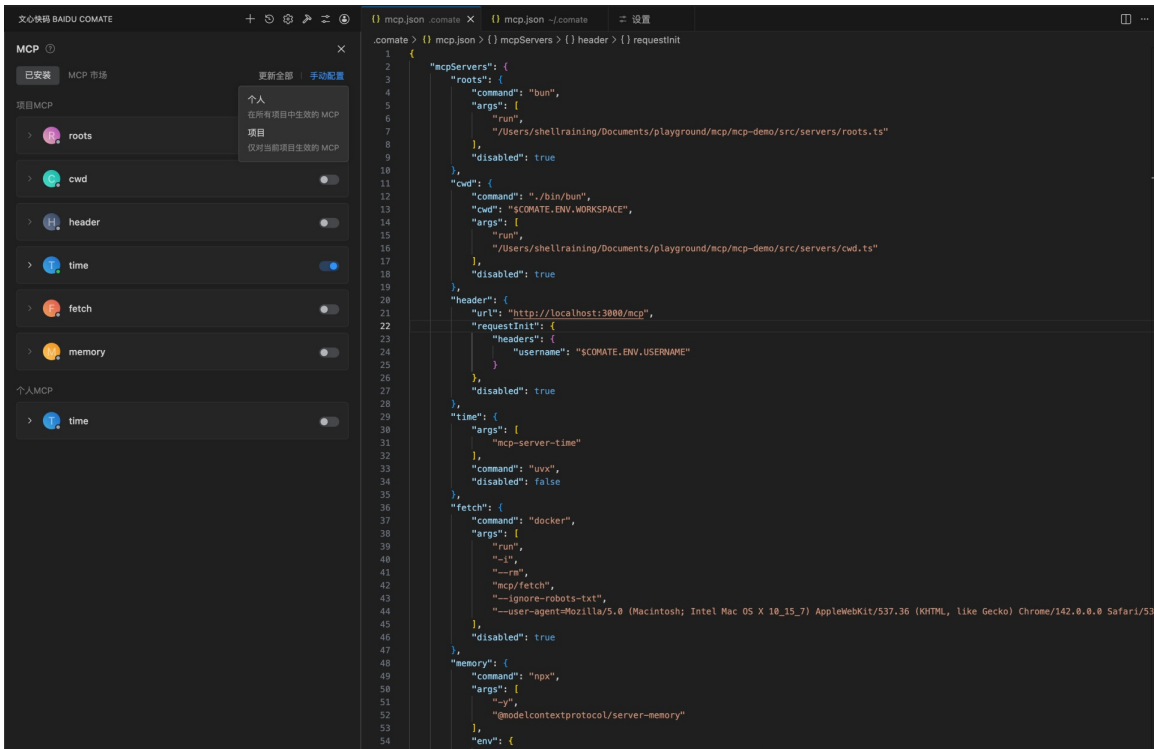
2. 在 MCP 市场中，浏览或搜索所需 MCP 服务，点击“安装”打开安装指引，如果 MCP 需要填写一些必备的信息，会额外出现一个表单，填写后可以点击 **JSON 配置预览** 查看填写的填写的状态。



### 方式二：手动完成添加

如果市场中没有想要的 server 或者正在开发一个 server，可以通过点击手动配置打开配置文件，Comate 支持项目级和全局级 MCP 配置，项目级位于打开的项目的 `.comate/mcp.json`，全局级位于家目录的 `.comate/mcp.json` 下。

1. 点击右上角“手动配置”，选择配置的层级，打开对应的 `mcp.json` 文件



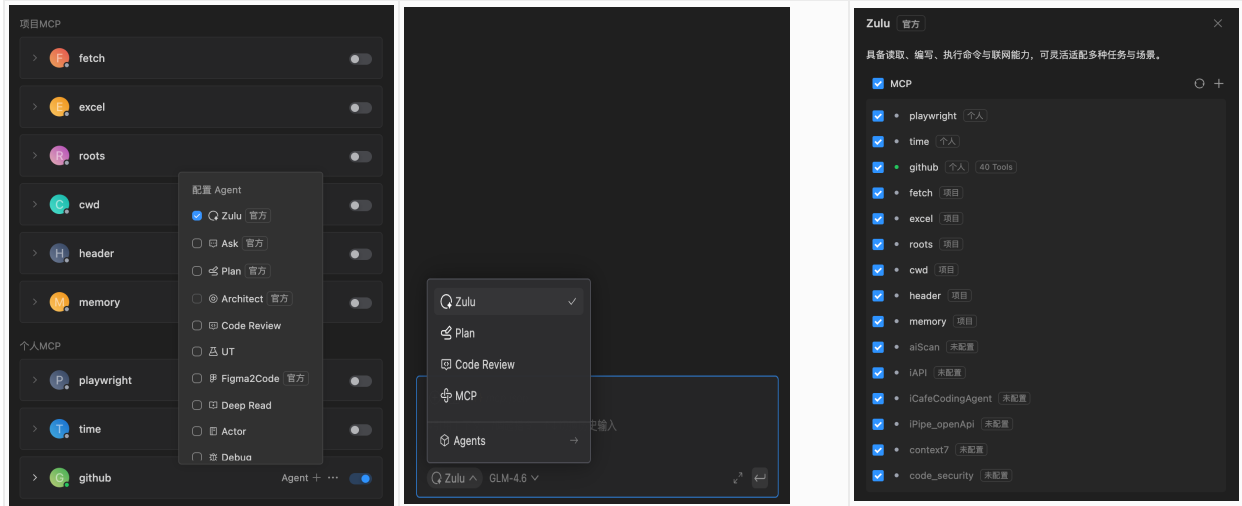
2. 填写想要使用的 MCP server 的配置信息

```

{
  "mcpServers": {
    "sequential-thinking": {
      "command": "npx",
      "args": [
        "-y",
        "@modelcontextprotocol/server-sequential-thinking"
      ]
    }
  }
}

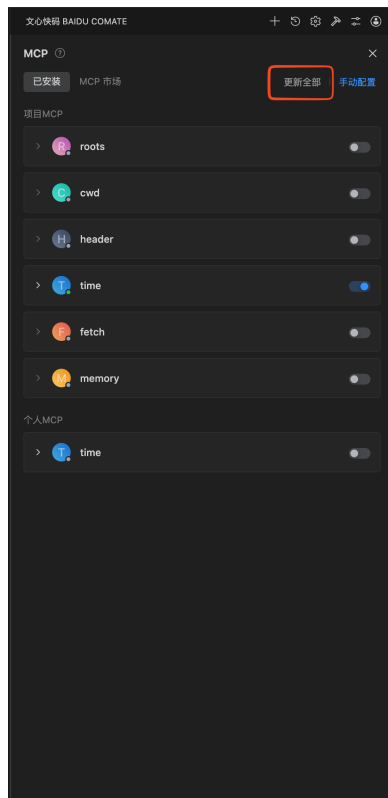
```

3. 手动添加后，一定要手动添加到 agent 中，避免出现 server 已经成功启动，但是 agent 无法调用的情况



### 4. 验证调试

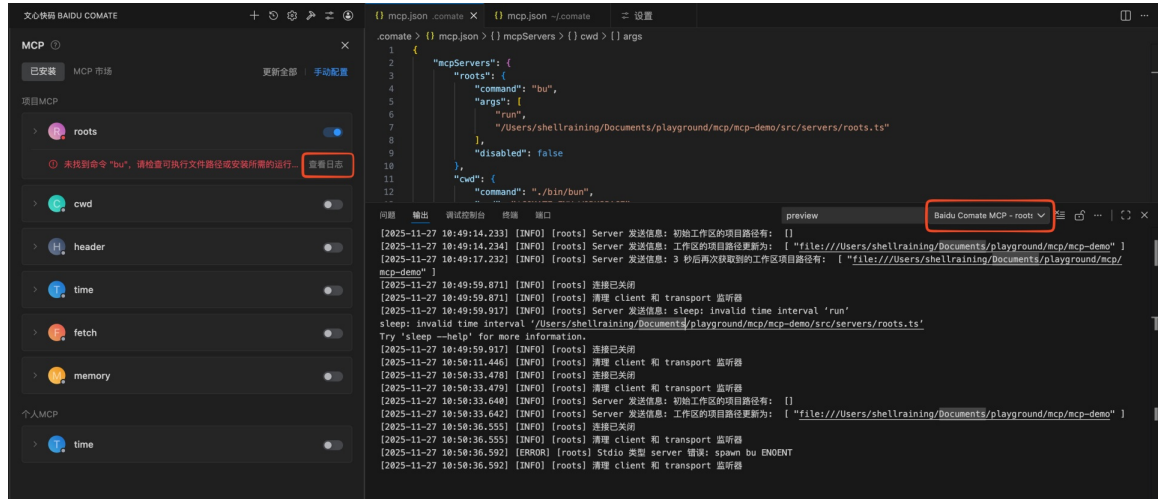
1. 当您在市场中/手动添加后，跳转回“已安装” tab 页，可以看到该 MCP server 会被识别到且自动启动，如果没有看到启动，可以点击右上角更新全部按钮确保 MCP 服务正常运行。



2.可以跳转到“已安装”的 tab 页查看已经安装的 MCP server，配置的 MCP server 有四种状态，分别对应四种颜色的状态灯



启动失败后可以点击“查看日志”灰色按钮，从 Baidu Comate MCP 对应的 MCP server 输出通道中查看输出日志



5. 在对话中使用 MCP 工具 在对话中，如果 Comate 认为您的问题与某个 MCP Server 相关，它将自动使用 MCP 中提供的工具。如果您认为本次对话有必要使用某个工具，只需告知 Comate 使用该工具名称或描述即可。

1. 默认情况下，当 Comate 想要使用 MCP 工具时，它会显示一条消息请求您的批准。



2. 当您同意 Comate 使用后，将自动完成 MCP 工具的使用，并将结果作为上下文继续生成后面的内容。

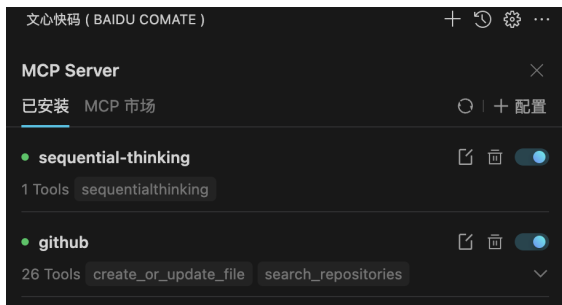


- 如果 MCP 工具会调用很多次，在确认 MCP server 时可信任的前提下，可以勾选自动执行，自动执行启动时，会折叠输入参数和结果，你可以点击卡片展开

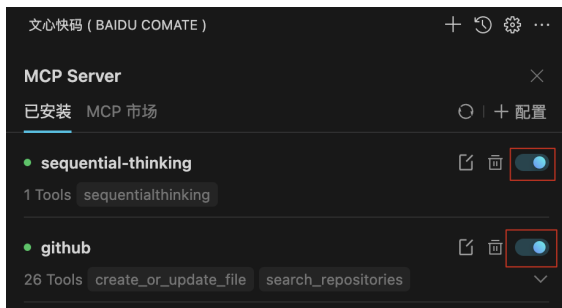


## 6. 查看/启用/禁用 已安装的MCP服务

- 若已安装了MCP服务，在进入配置项目MCP服务页面的“已安装”页签中，可看到所有已安装过的MCP服务。



- 若您暂时不需要某个MCP服务生效，可以通过该MCP服务最后的“开关”操作，进行启用/禁用 启用/禁用与对话中是否可调用成功无关，禁用则一定无法调用。



## 7. 使用前须知 为了帮助您更好地使用MCP功能，我们温馨提醒您关注以下使用要点（相关功能正在持续优化中）：

**使用条件** 目前 MCP 仅支持 Zulu 模式下使用，建议您提前确认账户剩余的 Comate 可用次数，以确保流程顺利进行。

**工具稳定性说明** 由于 MCP 工具的调用逻辑由 AI 模型自主决策，且可能涉及不同开发者编写的服务端代码，存在以下需要您理解的情况：

- 工具调用具有不确定性，可能出现未触发调用或调用未成功的情况。
- 每次调用尝试（无论结果如何）都将消耗 1 次 Comate 使用次数。
- 我们建议您在关键流程中做好备用方案，以确保服务连续性。

**文件配置规范** 为保障 MCP 服务稳定运行，请特别注意：

- 项目路径 `.comate/mcp.json` 的配置文件需控制总行数在 5000 行以内。
- 超过行数限制可能导致服务加载异常，建议定期优化文件结构。

## 配置文件位置与优先级

- 全局：`~/.comate/mcp.json`
- 项目：`<workspace>/comate/mcp.json`
- 本地（仅当前机器）：`<workspace>/comate/mcp.local.json`（实验性质）
- 合并规则：相同 server 名字后写覆盖先写，优先级 `local > project > global`。解析失败时会在 UI/日志中提示 MCP 配置错误。

## 基本结构

```
{
  "mcpServers": {
    "serverName": {
      "transportType": "stdio",
      "command": "node",
      "args": ["server.js"],
      "env": { "TOKEN": "xxx" },
      "cwd": "/path/to/workdir",
      "timeout": 50,
      "disabled": false
    }
  }
}
```

## 字段解析

字段	说明	默认值
type	传输类型：stdio/sse/streamableHttp	无，必填
disabled	是否禁用	false
timeout	超时（秒）	30
command	本地进程入口（仅 stdio）	必填
args	命令行参数数组（字符串自动 trim）	[]
env	进程环境变量	{}
cwd	进程工作目录	用户家目录
url	远程地址（仅 sse/streamableHttp）	必填
headers	HTTP 头（与 requestInit.headers 合并，headers 优先）	{}
requestInit.headers	HTTP 头（较低优先级，主要用于向后兼容）	{}

**占位符与变量解析** 支持 `$COMATE.ENV.*` 占位符，可使用的占位符暂时包括：

- `USERNAME` -> 当前用户
- `WORKSPACE` -> 工作区根路径（file URI 转路径）可嵌入到任意字符串字段：command、args、env 值、cwd。未知占位符会被原样保留。

```
{
  "command": "bun",
  "args": ["run", "$COMATE.ENV.WORKSPACE/src/servers/roots.ts"],
  "cwd": "$COMATE.ENV.WORKSPACE"
}
```

### 示例 Local / stdio

```
{
  "mcpServers": {
    "my-stdio-server": {
      "command": "node",
      "args": ["server.js", "--port", "8080"],
      "env": { "PORT": "8080" },
      "cwd": "$COMATE.ENV.WORKSPACE",
      "timeout": 45
    }
  }
}
```

### Remote / SSE

```
{
  "mcpServers": {
    "my-sse-server": {
      "type": "sse",
      "url": "https://example.com/sse",
      "headers": { "Authorization": "Bearer ..." },
      "timeout": 60
    }
  }
}
```

### Remote / Streamable HTTP

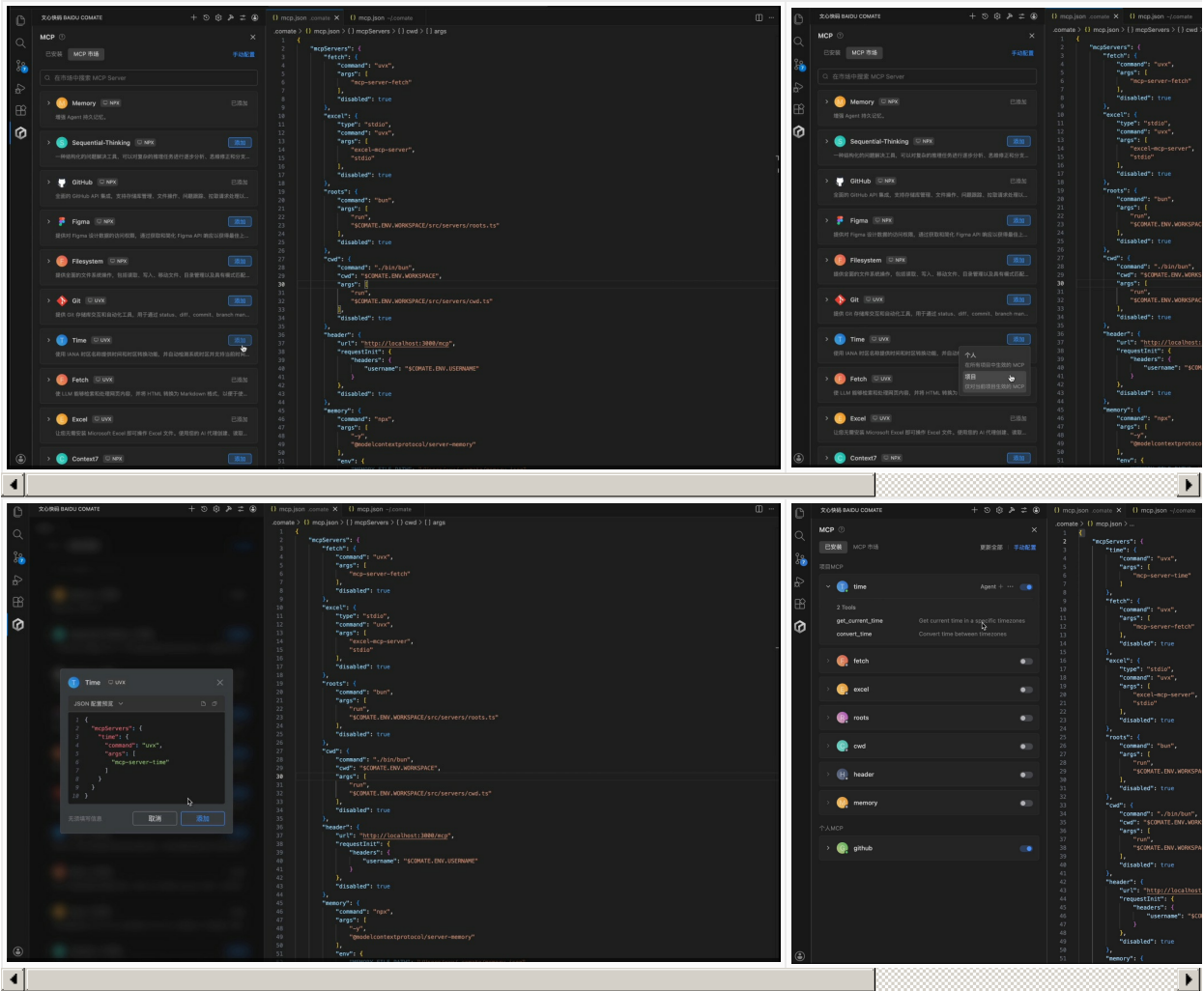
```
{
  "mcpServers": {
    "my-streamable-http": {
      "transportType": "streamableHttp",
      "url": "https://example.com/mcp",
      "headers": { "X-Trace": "1" }
    }
  }
}
```

### 常见问题 & 排查

- 占位符未替换：确认字段是字符串，格式为 `$COMATE.ENV.KEY`，KEY 是否在支持列表（`USERNAME/WORKSPACE`）。
- `command` 找不到文件：检查 `cwd` 是否正确（默认家目录）；可显式设置 `cwd: "$COMATE.ENV.WORKSPACE"`。
- HTTP headers 丢失：`requestInit.headers` 与 `headers` 会合并，后者优先；避免键名大小写冲突。

### 🔗 MCP 使用案例

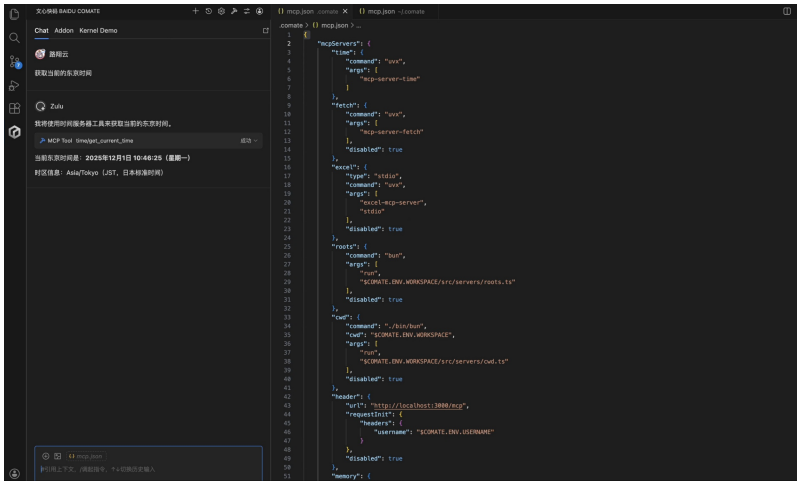
**案例1：使用 time MCP 工具** 下图展示了如何从市场添加一个 MCP server，到实际对话中使用。



使用流程如下：

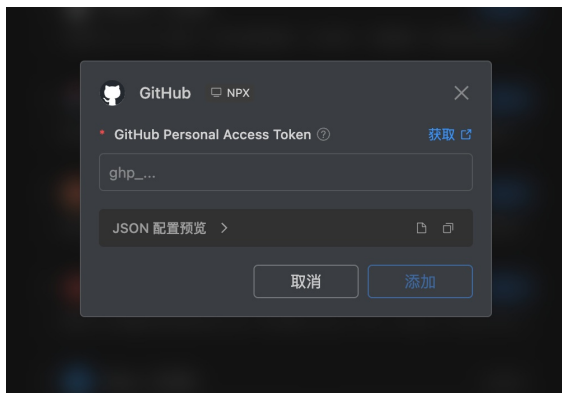
用户指令：获取当前的东京时间。

Comate智能决策：自动识别需调用 time MCP 服务工具 → 展示工具调用确认弹窗，用户点击"执行"后开始运行。

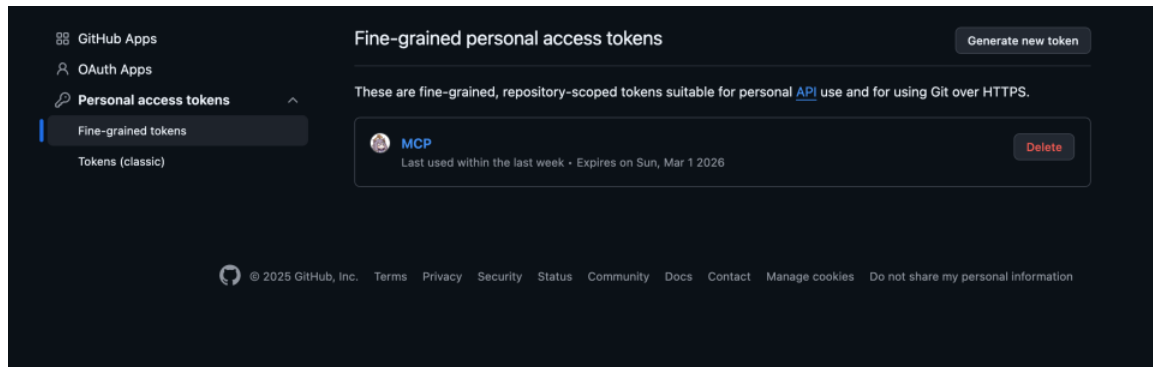


案例2：使用 GitHub MCP 工具 如何使用 GitHub MCP 工具去获取某个仓库的提交信息，详细流程如下：

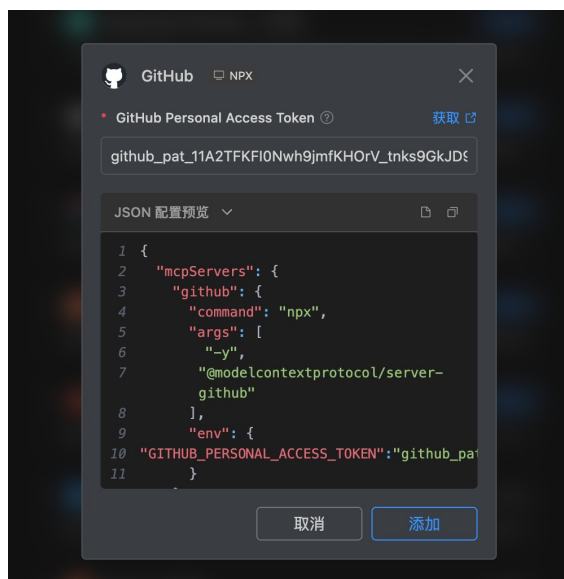
1. 获取 GitHub Access Token，可以先点击 GitHub MCP server 这个卡片的添加按钮，弹出的模态窗会要求填写一个 GitHub Personal Access Token，点击右侧的获取按钮即可跳转到对应链接。



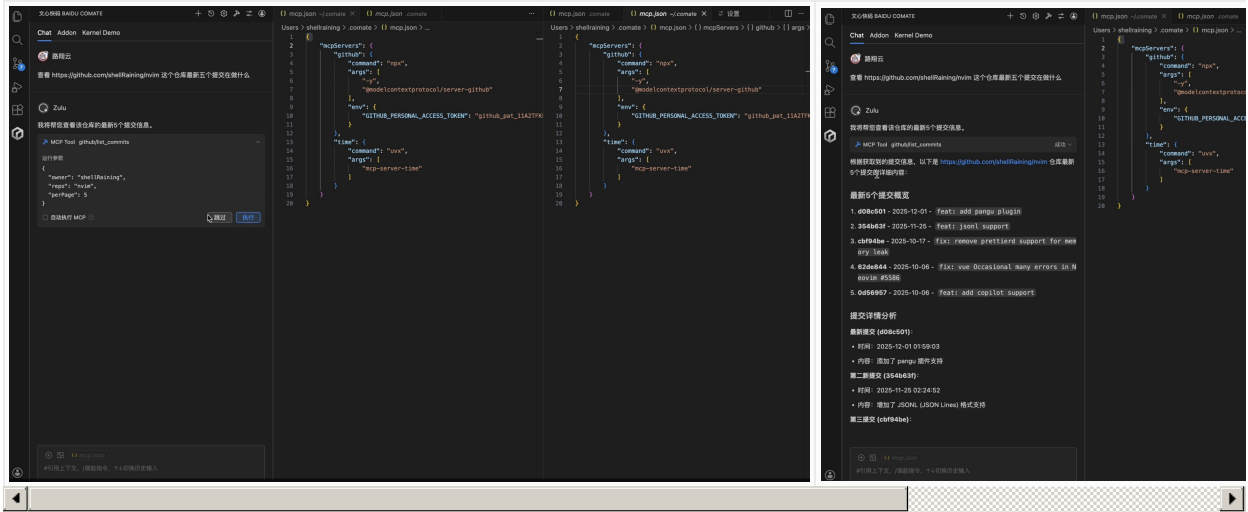
2. 点击右上角的 generate new token 按钮生成一个 token，保存到一个安全的位置。



3. 填写到模态窗中对应的 placeholder，点击预览栏打开预览窗口，确认无误点击添加。



4. 使用。



除了使用市场内本地的 GitHub MCP server，还可以使用他们官方的远程 MCP server，相对来说占用更少且功能更丰富，配置如下，替换 token 后复制粘贴到 mcp.json 即可。

```

{
  "mcpServers": {
    "github": {
      "url": "https://api.githubcopilot.com/mcp/",
      "headers": {
        "Authorization": "Bearer <your-token>"
      }
    }
  }
}

```

### 🔗 MCP Debug 实践

在具体讲 debug 之前，首先了解一下 VSCode debug 的两种模式，launch 和 attach，它们的区别在于“谁来启动被调试的进程”。

- launch 模式：是指由 VSCode 来直接启动需要被 debug 的程序，很适合我们 debug 一些随意写的脚本（比如刷礼扣……），但是对于 MCP 就不是很适合，因为 stdio 类型的 MCP server 需要由宿主进程启动，如果直接由 VSCode 启动，我们无法完成握手等操作。
- attach 模式：是指 VSCode 不负责启动进程，而是“附加”到一个已经在运行的进程。我们需要手动先把要调试的进程启动起来，并将他暴露到一个端口上，然后将调试器附加在其上，就可以实现调试的功能。

stdio 类型 server 上面提到，stdio 类型的 server 必须由宿主进程启动，因此我们只能选择第二种调试方式，具体的操作步骤为：

1. 将我们用 TypeScript 编写的 MCP server 编译，注意开启 sourceMap 和 declaration。

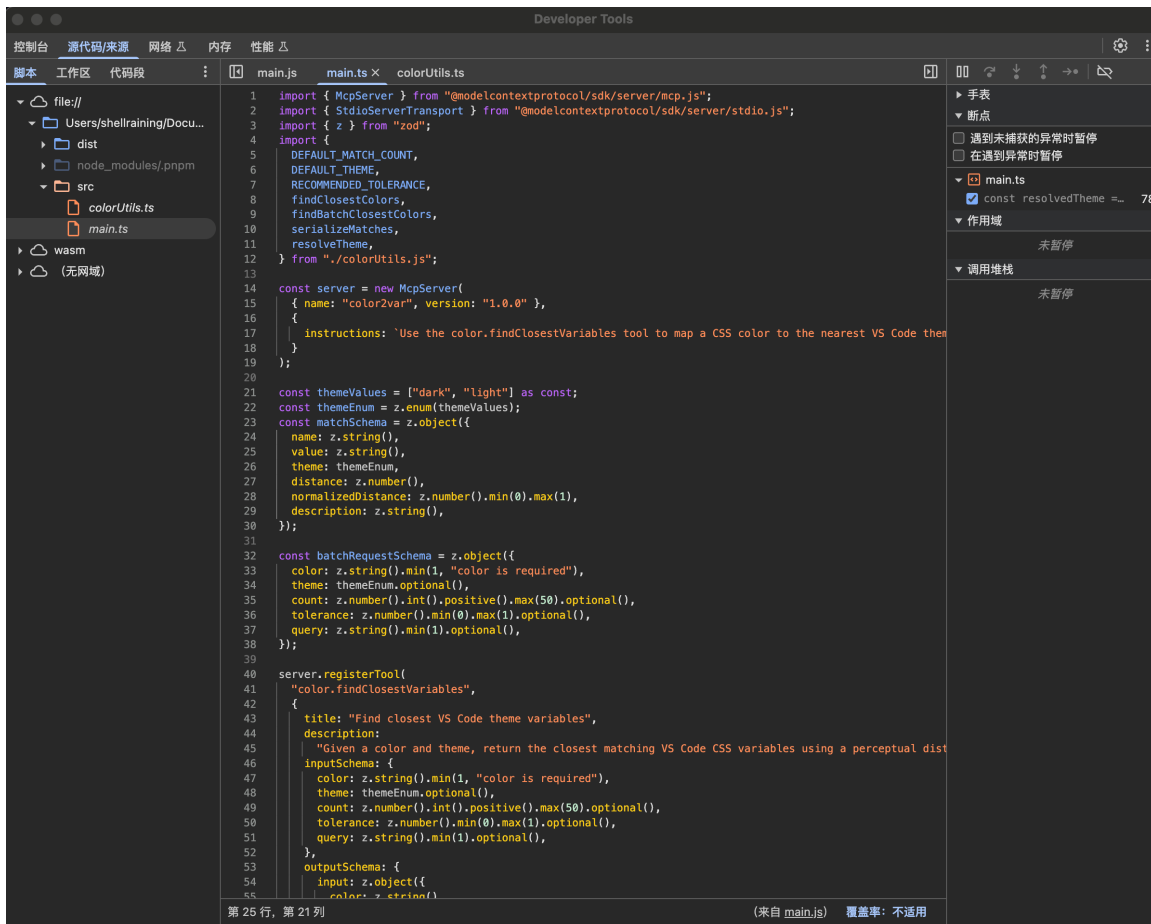
```
{
  "compilerOptions": {
    "target": "ES2022",
    "module": "ESNext",
    "moduleResolution": "Node",
    "strict": true,
    "esModuleInterop": true,
    "forceConsistentCasingInFileNames": true,
    "skipLibCheck": true,
    "resolveJsonModule": true,
    "outDir": "./dist",
    "declaration": true,
    "declarationMap": true,
    "sourceMap": true,
    "allowImportingTsExtensions": false
  },
  "include": ["src/**/*.ts", "src/**/*.d.ts"]
}
```

2. 在 mcp.json 里面添加指定的 server，注意加上 `--inspect-brk` 以调试模式启动。

```
{
  "mcpServers": {
    "color2var": {
      "type": "stdio",
      "command": "node",
      "args": [
        "--inspect-brk",
        "/Users/shelltraining/Documents/playground/mcp/evaluation/color2var/dist/main.js"
      ],
      "disabled": true
    }
  }
}
```

3. 浏览器打开 `chrome://inspect`，找到 9229 端口启动的程序，点击 `inspect` 开始调试（如果没有找到 9229 端口，请点击页面上第二个 checkbox 里面的 `configure` 按钮，在里面把 9229 端口加上）。



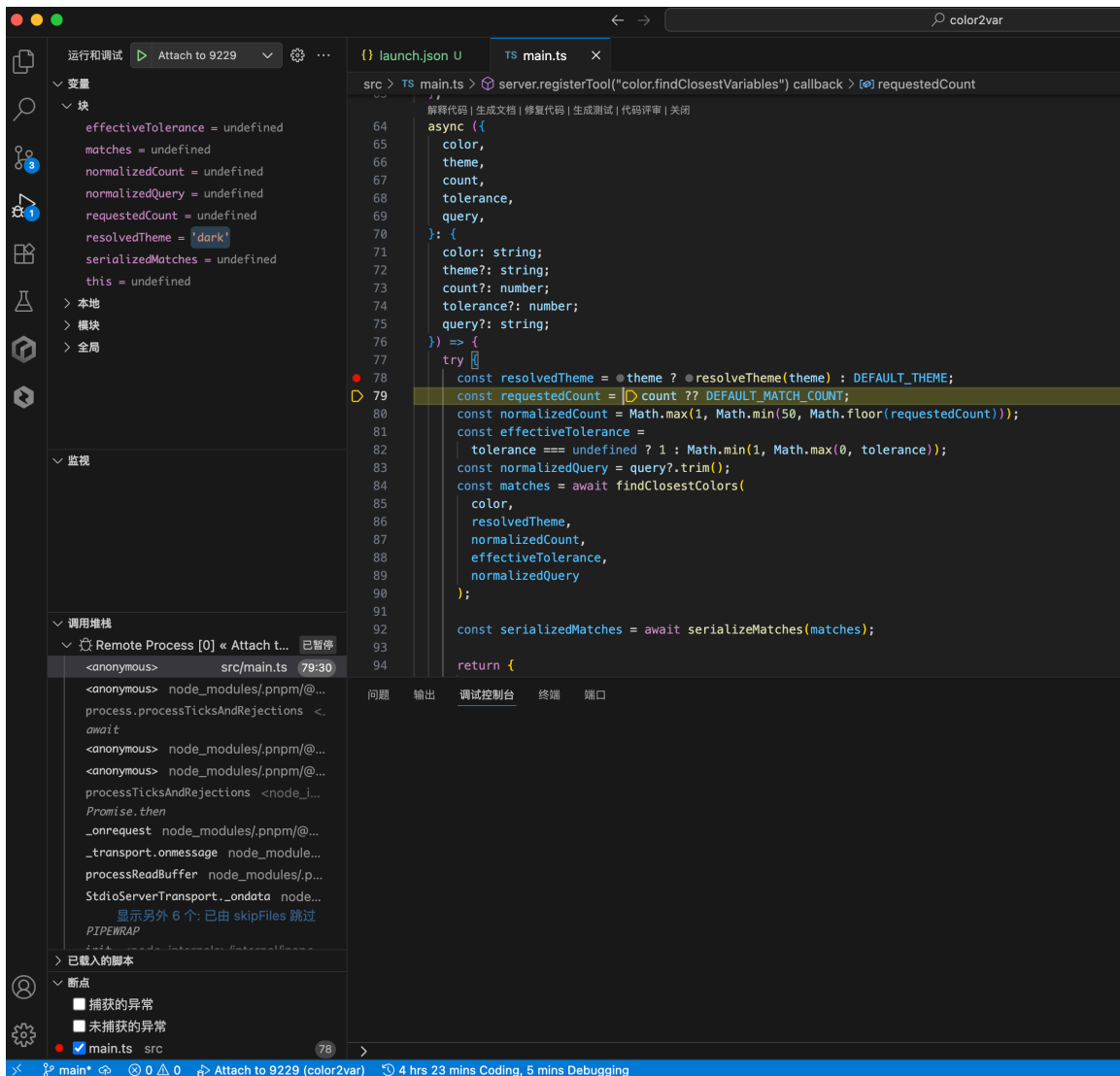


4. 如果想要使用 VSCode 进行调试，我们需要在 `.vscode/launch.json` 里面添加配置。

```

{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Attach to 9229",
      "port": 9229,
      "request": "attach",
      "skipFiles": ["<node_internals>/**"],
      "type": "node"
    }
  ]
}

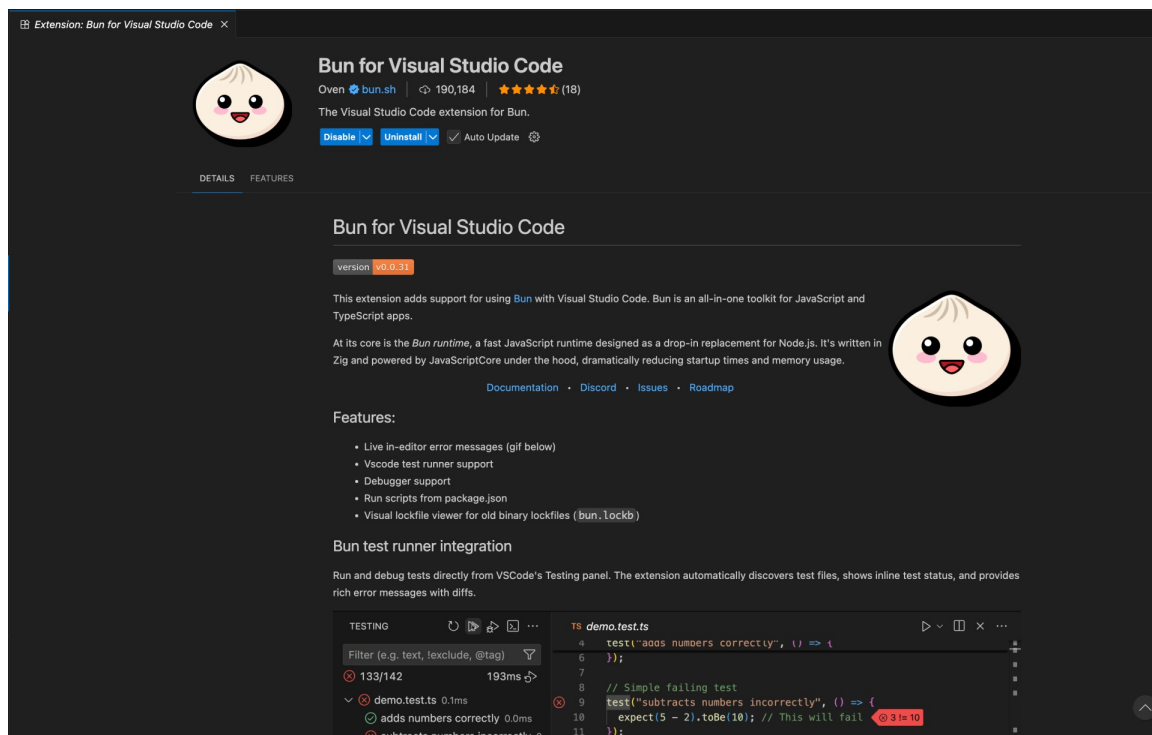
```



**remote 类型** Remote 类型的 MCP server debug 起来就相对轻松多了，因为我们是先启动 server，然后等待 client 的连接，因此可以直接使用 launch 的方式启动，不必通过 attach 的方式间接 debug。由于我们通过 TypeScript SDK 编写代码，因此我这里极力推荐使用 bun 运行时来进行 debug，他能够直接运行 TypeScript 代码。

使用方式也很简单，我们只需要做两步操作：

1. 安装 bun 运行时，可以参照网页 <https://bun.com/docs/installation> 来做；
2. 在 VSCode 拓展商店下载 bun for visual studio code。



Extension: Bun for Visual Studio Code

## Bun for Visual Studio Code

Oven [bun.sh](#) | 190,184 | ★★★★★ (18)

The Visual Studio Code extension for Bun.

[Disable](#) [Uninstall](#) [Auto Update](#)

DETAILS FEATURES

### Bun for Visual Studio Code

version **v0.0.31**

This extension adds support for using Bun with Visual Studio Code. Bun is an all-in-one toolkit for JavaScript and TypeScript apps.

At its core is the *Bun runtime*, a fast JavaScript runtime designed as a drop-in replacement for Node.js. It's written in Zig and powered by JavaScriptCore under the hood, dramatically reducing startup times and memory usage.

[Documentation](#) · [Discord](#) · [Issues](#) · [Roadmap](#)

**Features:**

- Live in-editor error messages (gif below)
- Vscode test runner support
- Debugger support
- Run scripts from package.json
- Visual lockfile viewer for old binary lockfiles (`bun.lockb`)

### Bun test runner integration

Run and debug tests directly from VSCode's Testing panel. The extension automatically discovers test files, shows inline test status, and provides rich error messages with diffs.

TESTING

Filter (e.g. text, !exclude, @tag)

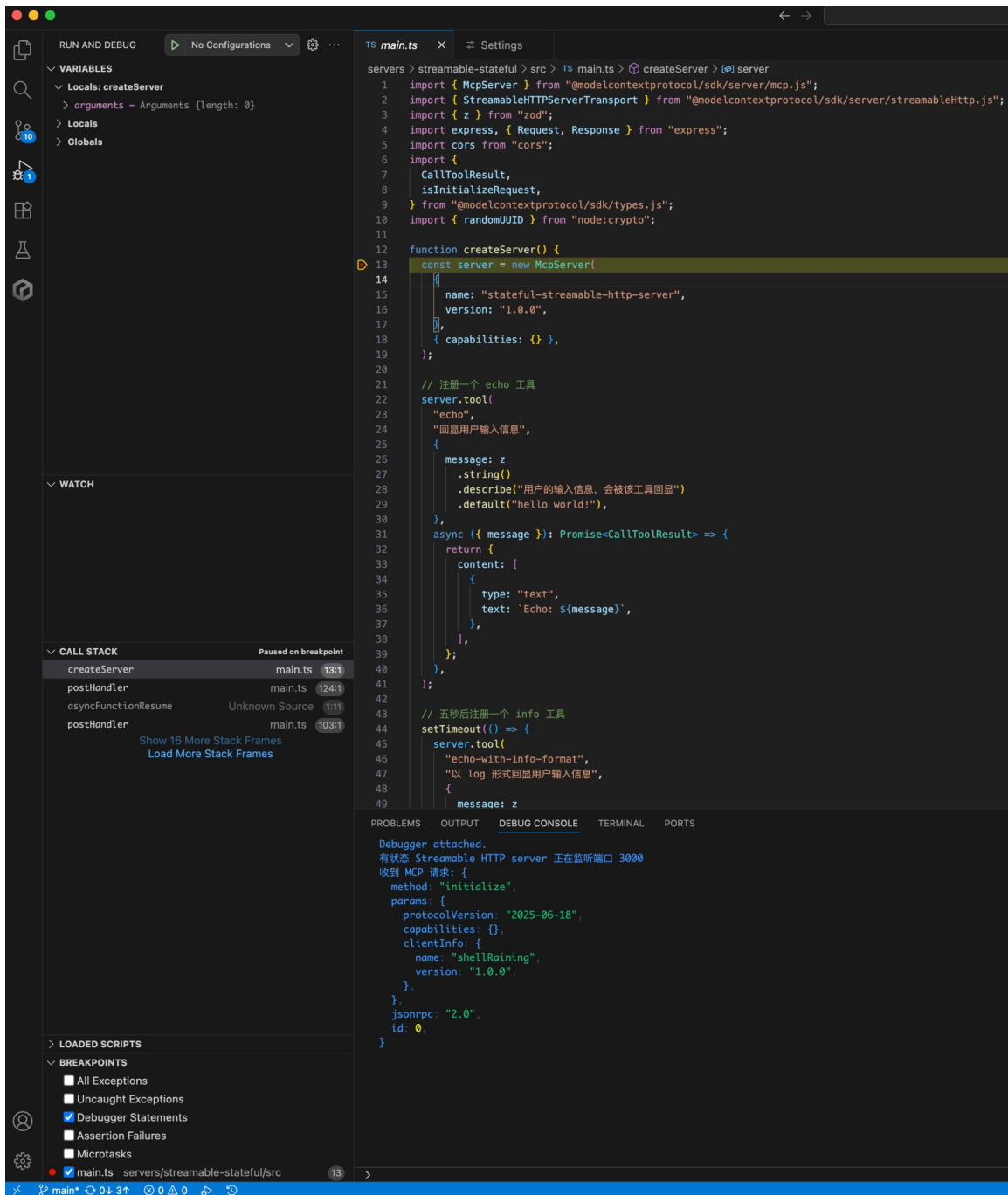
133/142 193ms

demo.test.ts 0.1ms

- adds numbers correctly 0.0ms
- subtracts numbers incorrectly 0.0ms

```
TS demo.test.ts
4 test("adds numbers correctly", () => {
5   });
6
7
8 // Simple failing test
9 test("subtracts numbers incorrectly", () => {
10  expect(5 - 2).toBe(10); // This will fail
11  });
```

3. 然后就可以按下 F5 启动服务器，等待客户端连接后调试。



**log 输出** 如果不喜欢断点调试的方式，可以通过 log 来判断，方式是打开 Comate 控制台，然后跳转到 output 栏，选择 Baidu Comate MCP 输出通道，就可以看到所有的日志信息，因为我们是通过 `stderr` 通道输出，因此会有一个前缀 "Server xxx stderr:"，后面是实际的日志信息。

```

37     "filesystem": {
38       "command": "npx",
39       "args": [
40         "-y",
41         "@modelcontextprotocol/server-filesystem"
42       ],
43       "disabled": true
44     },
45     "pointer": {
46       "command": "npx",

```

```

[2025-10-16 13:58:28] [INFO] Server serena stderr: INFO 2025-10-16 13:58:28,589 [SerenaAgentExecutor_0] solidlsp:start:1622 - Starting language server with language typescript for /Users/shelltraining/Documents/baidu/ide/comate-plugin-host
[2025-10-16 13:58:28] [INFO] Server serena stderr: INFO 2025-10-16 13:58:28,590 [SerenaAgentExecutor_0] solidlsp_start_server:224 - Starting TypeScript server process
[2025-10-16 13:58:28] [INFO] Server serena stderr: INFO 2025-10-16 13:58:28,590 [SerenaAgentExecutor_0] solidlsp_ls_handler:start:189 - Starting language server process via command: [!]/Users/shelltraining/serena/language_servers/static/TypeScriptLanguageServer/ts-lsp/node_modules/.bin/typescript-language-server', '--stdio']
[2025-10-16 13:58:28] [INFO] Server serena stderr: INFO 2025-10-16 13:58:28,602 [SerenaAgentExecutor_0] solidlsp_start_server:228 - Sending initialize request from LSP client to LSP server and awaiting response
[2025-10-16 13:58:28] [INFO] Server serena stderr: INFO 2025-10-16 13:58:28,728 [LSP-stdout-reader] solidlsp:window_log_message:207 - LSP: window/logMessage: {"type": 3, "message": "Using Typescript version (workspace) 5.3.3 from path "/Users/shelltraining/Documents/baidu/ide/comate-plugin-host/node_modules/typescript/lib/tsserver.js"}
[2025-10-16 13:58:29] [INFO] Server serena stderr: INFO 2025-10-16 13:58:29,735 [SerenaAgentExecutor_0] solidlsp_start_server:246 - Timeout waiting for TypeScript server to become ready, proceeding anyway
[2025-10-16 13:58:29] [INFO] Server serena stderr: INFO 2025-10-16 13:58:29,735 [SerenaAgentExecutor_0] serena.agent:stop:336 - Language server initialization completed in 1.971 seconds
[2025-10-16 13:58:29] [INFO] Server serena stderr: INFO 2025-10-16 13:58:29,735 [SerenaAgentExecutor_0] serena.agent:stop:336 - Task-1[init_language_server] completed in 1.972 seconds

```

## 🔗 MCP 启动时常见错误

### 1. 环境问题 未找到命令 (spawn uvx ENOENT)



MCP Server 可以使用多种语言实现，例如 JavaScript、Python、Java 等。要运行这些服务器，需要在本地预先安装相应的运行环境和工具。有时会遇到上图中类似“spawn uvx ENOENT”的报错，这意味着系统中缺少名为 uvx 的可执行文件（例如 uv 提供的命令行工具）。同理，也可能出现“spawn bunx ENOENT”等错误，提示缺少 bunx 工具。这些报错的原因本质上都是本地没有安装相应的运行环境或工具，解决办法是在系统中正确安装所需的解释器或包管理工具（如 Python、Node.js、Bun、uv 等）。下面整理了一些常见运行环境的安装文档。

- uv: <https://docs.astral.sh/uv/getting-started/installation>
- bun: <https://bun.sh/docs/installation>
- node: <https://nodejs.org/en/download>
- python: <https://www.python.org/downloads>

### 开发机确认环境已安装但仍找不到

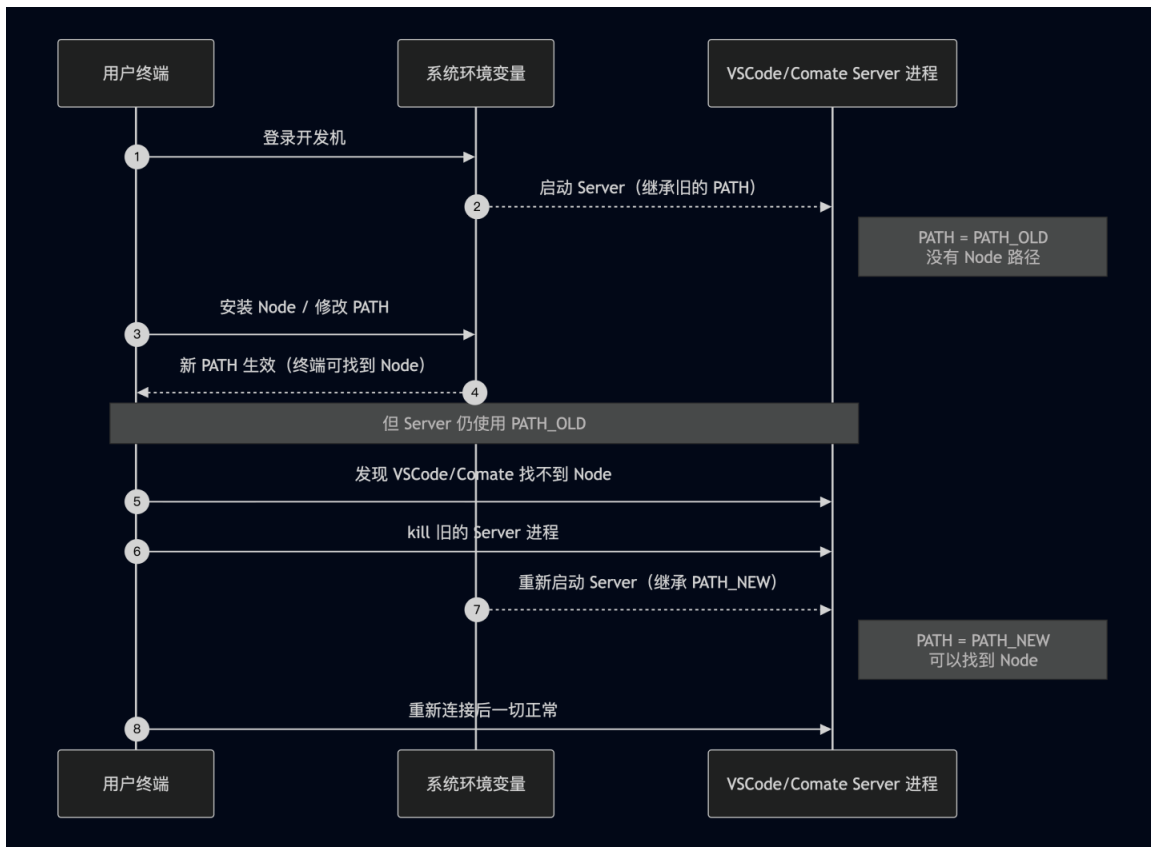
由于开发机上的 VSCode 或者 Comate 进程是启动比较早的，最开始启动时继承的环境变量里面并不包含 node 的路径等，所以在你登录开发机后且安装环境变量后，再次检查环境还是会出现找不到的情况，你需要重启 vscode-server 或者 comate-server 进程解决这个问题，命令如下：

```

pgrep -u $(whoami) -f comate-server | xargs kill -9 # 如果使用 Comate 登录开发机
pgrep -u $(whoami) -f vscode-server | xargs kill -9 # 如果使用 icoding 登录开发机

```

重启后即可找到对应环境变量。



## 2. node 类型 server 常见问题 存在 npx，但是执行时报错 You must supply a command.

```
server-sequential-thinking
work@ljjf-dev ~/.nvm/versions/node/v24.11.0/lib/node_modules/@modelcontextprotocol ((v0.39.7))
❶ $npx -y @modelcontextprotocol/server-sequential-thinking

ERROR: You must supply a command.

Execute binaries from npm packages.

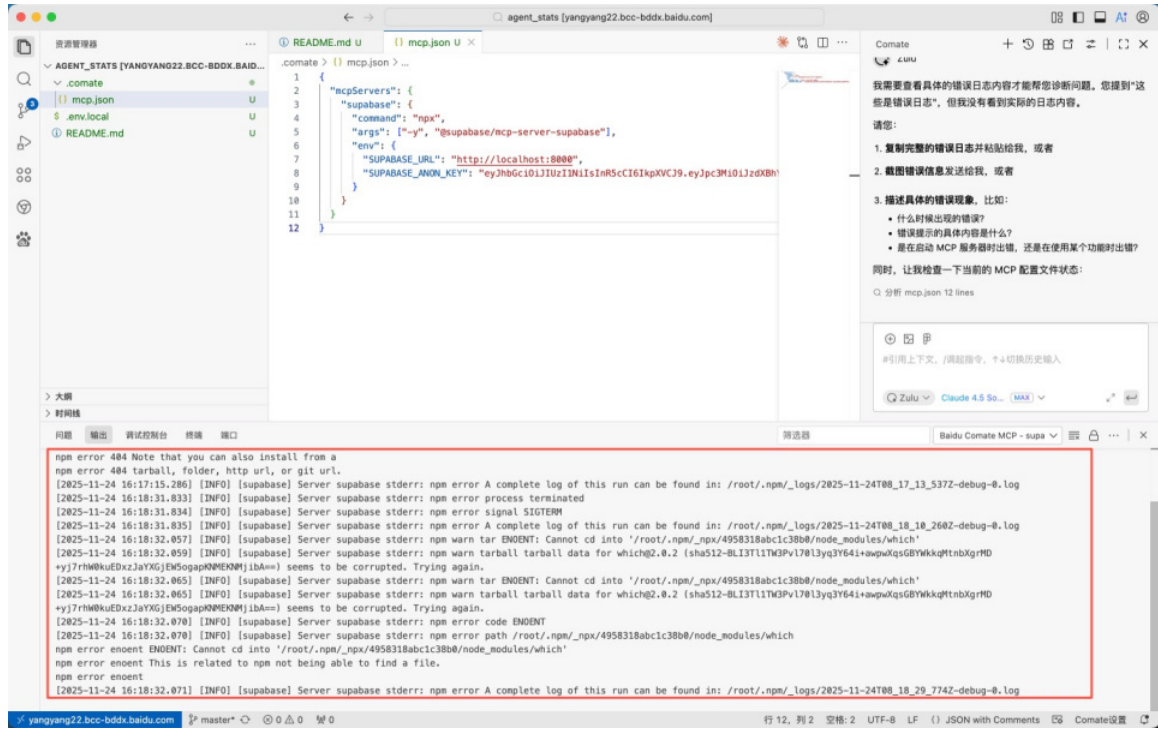
npx [options] <command>[@version] [command-arg]...
npx [options] [-p|--package <package>]... <command> [command-arg]...
npx [options] -c '<command-string>'
npx --shell-auto-fallback [shell]

Options:
  --package, -p      Package to be installed.           [string]
  --cache            Location of the npm cache.           [string]
```

上图这种情况有可能是因为 node 和 npx 版本不匹配导致的，比如 24 版本的 node 对应了一个 10 版本的 npx（正常来说应该是 11 版本），可以通过升级 npm 版本来解决，先尝试 `npm install -g npm@latest` 升级一下，如果遇到下面这样的报错（EEXIST），尝试 `npm install -g --force npm`。

```
npm error code EEXIST
npm error syscall symlink
npm error path ../.././lib/node_modules/npm/man/man1/npx.1
npm error dest /home/work/.nvm/versions/node/v24.11.0/share/man/man1/npx.1
npm error errno -17
npm error EEXIST: file already exists, symlink '.././lib/node_modules/npm/man/man1/npx.1' ->
'/home/work/.nvm/versions/node/v24.11.0/share/man/man1/npx.1'
npm error File exists: /home/work/.nvm/versions/node/v24.11.0/share/man/man1/npx.1
npm error Remove the existing file and try again, or run npm
npm error with --force to overwrite files recklessly.
npm error A complete log of this run can be found in: /home/work/.npm/_logs/2025-10-30T08_36_49_379Z-debug-0.log
```

### npm 缓存损坏问题



遇到下面类似的错误日志 (ENOENT) ，很有可能就是 npm 缓存损坏，问题，修复方式非常简单，执行 `npm cache clean --force && rm -rf ~/.npm/_npx`，然后重启 server 即可

```

npm error 404 Note that you can also install from a
npm error 404 tarball, folder, http url, or git url.
[2025-11-24 16:17:15.286] [INFO] [Supabase] Server supabase: stderr: npm error signal SIGTERM
[2025-11-24 16:18:31.833] [INFO] [Supabase] Server supabase: stderr: npm error process terminated
[2025-11-24 16:18:31.834] [INFO] [Supabase] Server supabase: stderr: npm error signal SIGTERM
[2025-11-24 16:18:32.052] [INFO] [Supabase] Server supabase: stderr: npm error A complete log of this run can be found in: /root/.npm/_logs/2025-11-24T08_17_13_537Z-debug-0.log
[2025-11-24 16:18:32.055] [INFO] [Supabase] Server supabase: stderr: npm warn tarball tarball data for which@2.0.2 (shas512-BL3T1L3WP3VTb2yL3g9Y6li4+pwpvXgSGBYWkkQMtNbx9rMD
+yj7rHm0kuEdx2JaYXg)EM50appKMEKQM]ibA==) seems to be corrupted. Trying again.
[2025-11-24 16:18:32.065] [INFO] [Supabase] Server supabase: stderr: npm warn tarball tarball data for which@2.0.2 (shas512-BL3T1L3WP3VTb2yL3g9Y6li4+pwpvXgSGBYWkkQMtNbx9rMD
+yj7rHm0kuEdx2JaYXg)EM50appKMEKQM]ibA==) seems to be corrupted. Trying again.
[2025-11-24 16:18:32.070] [INFO] [Supabase] Server supabase: stderr: npm error code ENOENT
[2025-11-24 16:18:32.071] [INFO] [Supabase] Server supabase: stderr: npm error path /root/.npm/_npx/4958318abc1c38b0/node_modules/which
npm error enoent ENOENT: Cannot cd into /root/.npm/_npx/4958318abc1c38b0/node_modules/which
npm error enoent This is related to npm not being able to find a file.
npm error enoent
[2025-11-24 16:18:32.071] [INFO] [Supabase] Server supabase: stderr: npm error A complete log of this run can be found in: /root/.npm/_logs/2025-11-24T08_18_29_774Z-debug-0.log

```

3. remote 类型错误 (SSE 和 Streamable HTTP 类型) mcp.json 配置 header 后找不到 header 很有可能是你的 server 有白名单过滤策略，把非白名单前缀的 header 给清理掉了，联系 server 管理员即可

一直处于连接状态 检查 type 或者 transportType 填写是否正确，目前回退仅实现了 Streamable HTTP -> SSE 的 fallback，未来会支持更多检测手段

### 4. 二进制类型 MCP问题

权限不足

如果你使用的是打包后的二进制 MCP，必须确保连接时这个 server 被赋予了执行权限，如果没有，就会遇到下面类似的问题：



输出中的日志类似

```
[2025-12-02 15:56:23.039] [INFO] 等待 mediator 初始化
[2025-12-02 15:56:23.057] [INFO] 等待 webview 初始化
[2025-12-02 15:56:24.237] [INFO] 开始初始化 MCP Servers
[2025-12-02 15:56:24.326] [INFO] 需要禁用的服务器: 无
[2025-12-02 15:56:24.326] [INFO] 需要启用的服务器: baidu-go-codeseek
[2025-12-02 15:56:24.326] [INFO] 保持状态的服务器: 无
[2025-12-02 15:56:24.356] [ERROR] 启用 Server baidu-go-codeseek 失败: spawn ./bin/baidu-go-codeseek EACCES
[2025-12-02 15:56:24.356] [INFO] MCP Servers 初始化完成
```

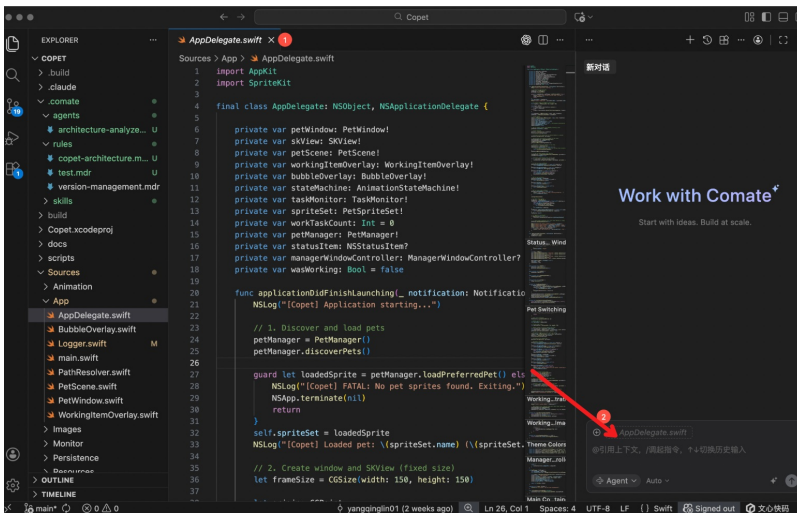
### @关联上下文

#### 自主识别上下文

Comate会根据当前正在打开的文件和你的Prompt识别对话的相关性，并自动添加到上下文中。分为3种情况：

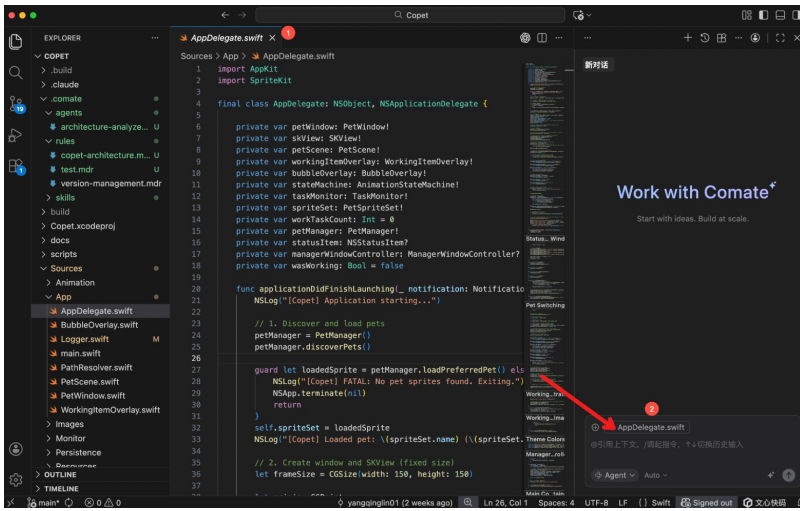
- 弱引用整个文件：Comate自动捕捉当前的打开的文件，并在输入框中以「虚线」的方式提示你这个文件被弱引用。

文件的弱引用是指这个文件的路径会被组装到Agent的上下文中，文件的内容不会被组装，当Agent根据你的输入判断需要获取文件内容时，它会根据文件路径读取内容。



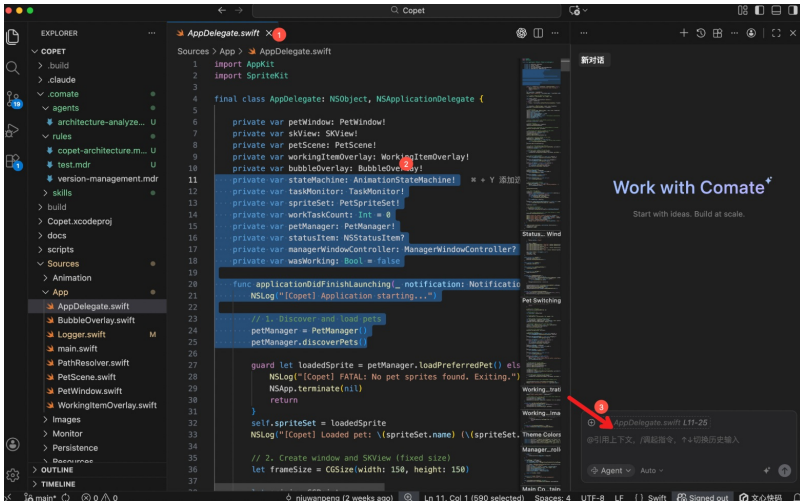
- 强引用整个文件：你可以直接在弱引用的文件（虚线文件）上点击，点击后将变成强引用，「虚线」变成「实线」。

强引用是指这个文件的路径和内容都会被组装到Agent的上下文中。

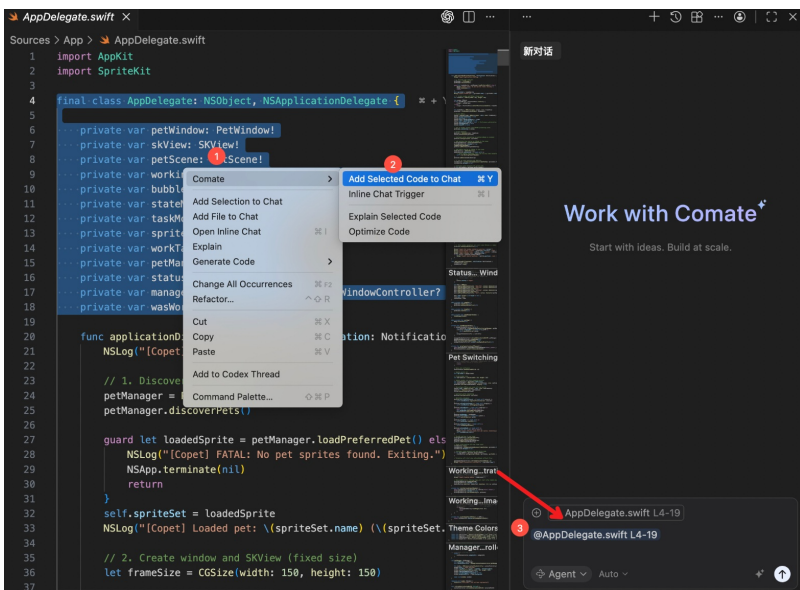


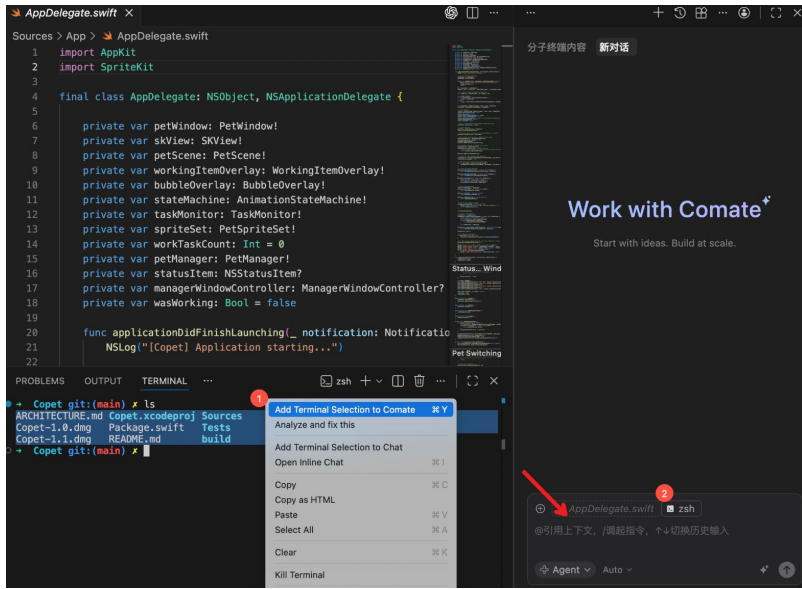
- 弱引用某段代码块：通过光标选中正在打开的文件中的某段代码，Comate会自动捕捉当前代码片段，并在输入框中以「虚线」的方式提示你这段代码被弱引用。

代码块的弱引用是指这个文件的路径和选中的代码块行号会被组装到Agent的上下文中，文件的内容不会被组装，当Agent根据你的输入判断需要获取代码块内容时，它会根据上述读取内容。



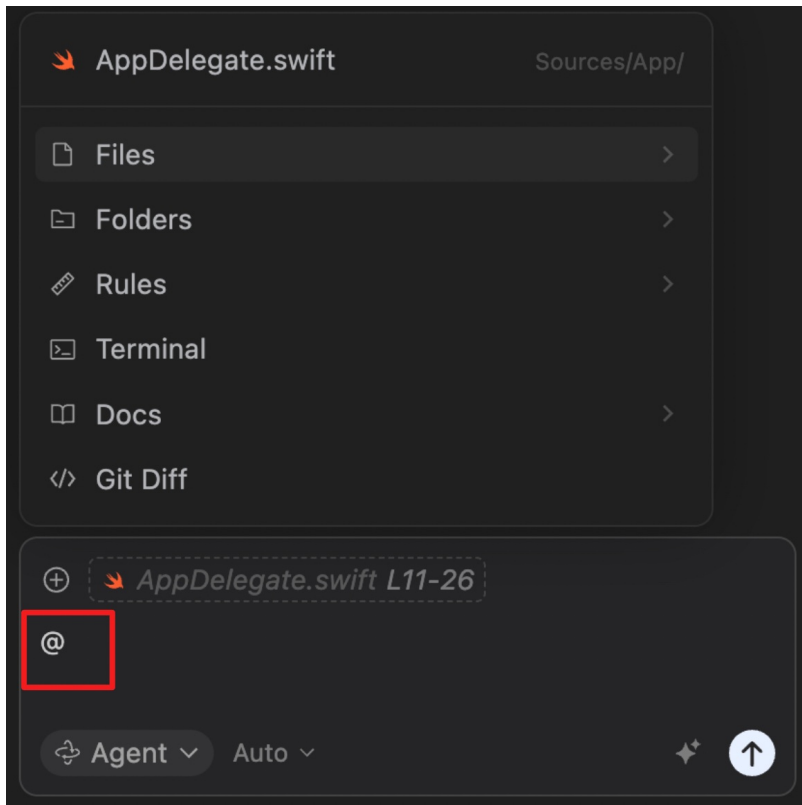
- 强引用某段代码块：通过光标选中正在打开的文件中的某段代码或终端输出，然后「右键」选择Add Selected Code To Chat或Add Selected Terminal To Chat，此时选中的内容将被强引用。

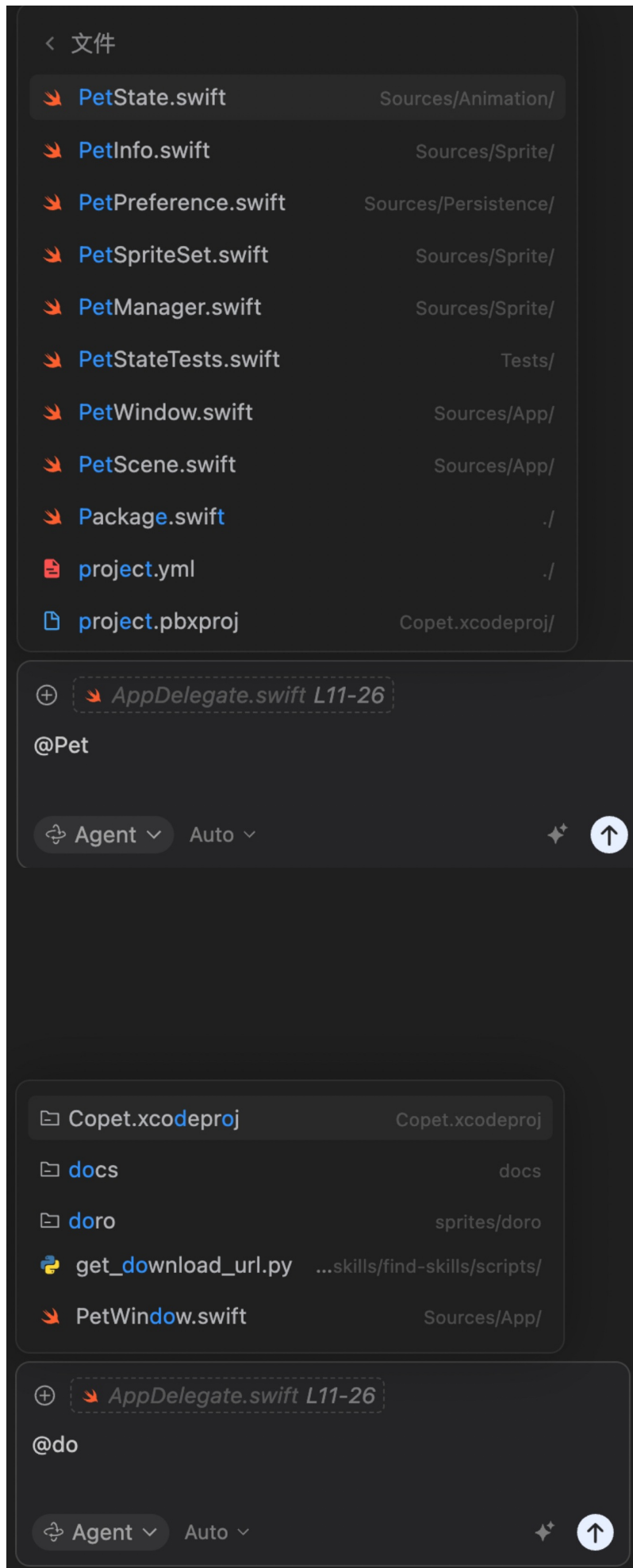




### 手动添加上下文

你还可以手动为Comate添加更多的上下文，以帮助Agent更准确地理解当前任务，使用@快速调起可添加的上下文列表。



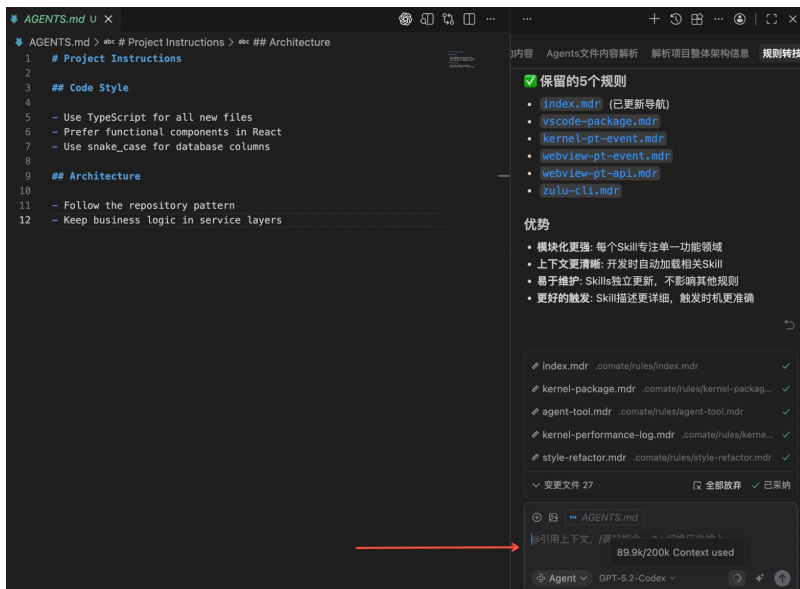


类型	作用
@Files	指定具体的文件，给出 Agent 需要明确参考或者理解的代码范围，避免 Agent 发散理解。
@Folders	指定具体的目录，给出 Agent 需要明确参考或者理解的代码范围，避免 Agent 发散理解。
@Terminal	获取当前终端的信息，包括命令执行结果、日志输出或报错信息。Agent 可以基于真实的运行反馈进行分析、定位问题并给出下一步操作建议，而不是仅凭静态代码推断。
@Git Diff	只针对Git仓库有效，用于获取git diff信息，通过 Git Diff，Agent 可以聚焦于“发生了什么变化”，更准确地进行代码评审、问题定位或后续修改，减少对无关代码的干扰。
@Rules	用于手动关联Rules文件，用于约束 Agent 的行为方式和输出结果，包括编码规范、技术栈限制、架构原则、安全要求或业务规则等。规则类上下文会在整个任务过程中持续生效，确保生成结果符合既定约定。
@Docs	@Docs功能可以让你利用文档来帮助编写代码，将本地文档上传到知识集中，即可在Comate中引用，可以引用一个知识集或某个具体的文件。

## 🔗 管理上下文

随着对话持续深入、引用越来越多上下文（文件、日志、文档等），模型 *Tokens*（消耗会快速增长，进而影响 Agent 响应质量、记忆能力和整体稳定性）。Comate 提供可视化的 *Tokens* 使用反馈与手动/自动压缩上下文的能力，帮助你直观监控对话“记忆”状态并主动控制上下文规模。你可以通过三个维度去使用这个能力：

- **Tokens 用量可视化**：在对话界面的输入区域内，实时展示当前会话已占用的上下文比例，用于直观反映模型当前“可记住的信息量”。
- **自动压缩上下文**：随着对话不断增长，Comate 会持续计算上下文占比，并在合适的时机自动进行上下文压缩。系统会保留关键结论和必要信息，对冗余细节进行总结或裁剪，在释放上下文空间的同时尽量保持对话语义的连续性。
- **手动管理上下文**：当上下文占比超过 70% 时，Comate 会在界面中进行提示，引导开发者主动管理上下文，例如触发压缩或调整当前对话内容，以避免上下文过载影响后续交互质量。

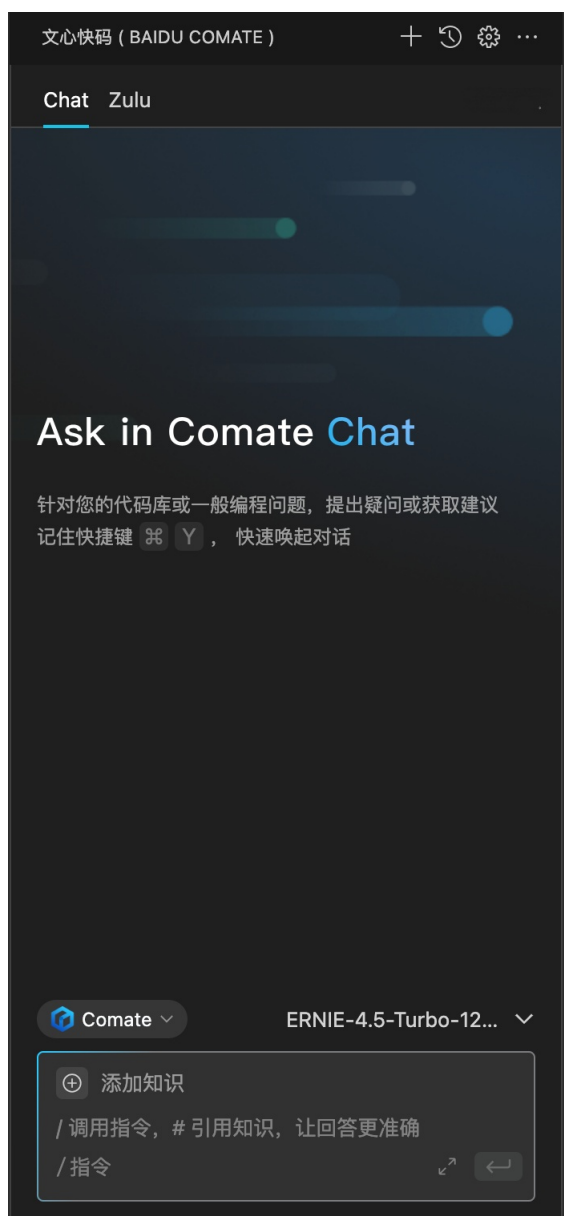


## 智能问答

代码问答能够深入理解您的代码库，并基于精准的上下文感知能力，为您提供准确且高度相关的代码解答和辅助。您可以自由选择不同的模型，并通过多种灵活的方式指定上下文，确保每一次对话都紧贴您的开发场景。

## 🔗 快速开始

1. 点击侧边栏图标: 在您的IDE侧边栏，找到并点击文心快码的图标。
2. 切换至 Chat Tab: 在展开的文心快码面板顶部，点击“Chat”标签，即可进入代码问答的对话界面。



## 🔗 模型选择

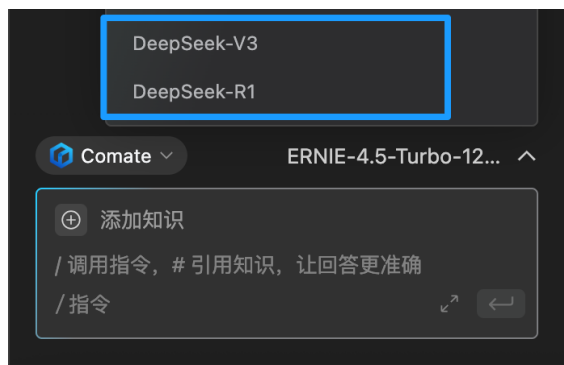
文心快码代码问答目前支持多种模型，您可以根据实际需求进行选择，以获得最佳的代码问答体验。

免费用户可以使用模型：

- ERNIE 4.5
- ERNIE 4.0 Turbo
- ERNIE 3.5
- ERNIE Speed
- ERNIE-X1

付费用户除免费用户的可以使用全部模型外，还可以使用：

- DeepSeek V3
- DeepSeek R1

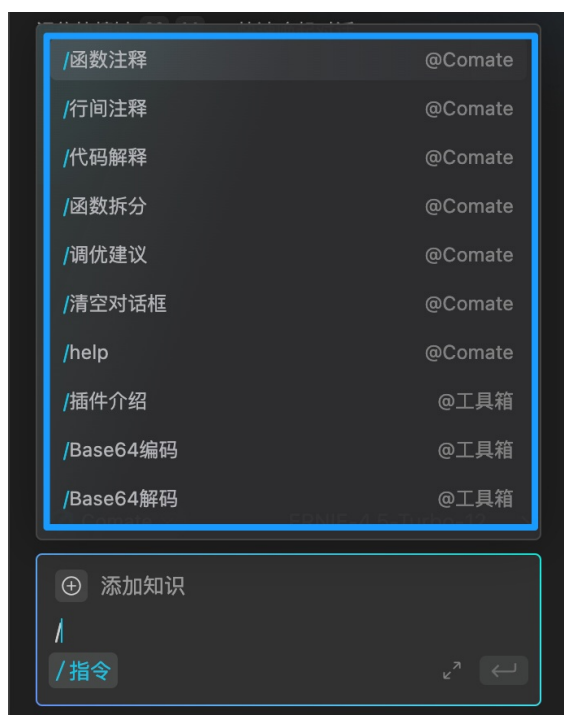


另外，企业用户还可以添加更多第三方模型提供商的API，比如[百度智能云 - 千帆ModelBuilder - API列表](#)

## 🔗 斜线命令

在代码问答对话框中，输入 / 即可触发斜线命令。文心快码内置了一系列常用的斜线命令，帮助您快速发起特定类型的提问，例如：/解释代码: 快速获得对当前代码片段的解释说明，帮助您理解代码逻辑和功能。

只需输入 / 即可查看完整列表。



## 🔗 插件扩展

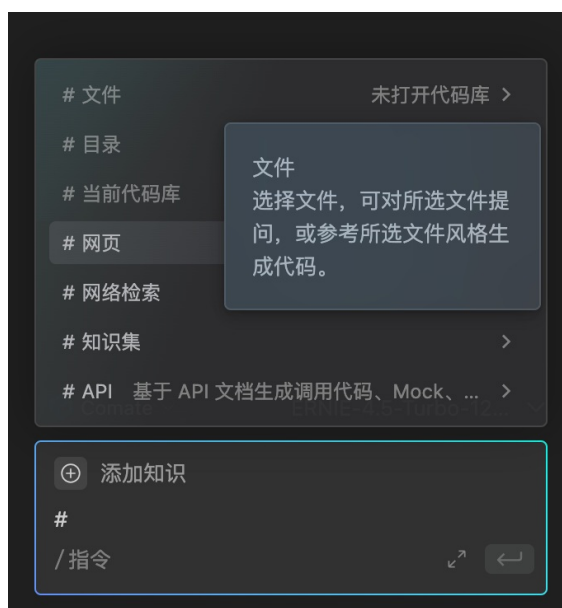
文心快码的代码问答能力，还可以通过丰富的插件进行扩展。输入 @ 即可触发插件列表，您也可以直接点击插件名称进行切换。例如，Git插件为您提供了版本控制相关的功能，其中就包括根据 Git Diff 快速生成提交信息的斜线命令。



### 🔗 引用知识（上下文）

为了让代码问答更准确地理解您的问题，并给出与当前开发环境高度相关的答案，文心快码提供了强大的上下文引用能力。

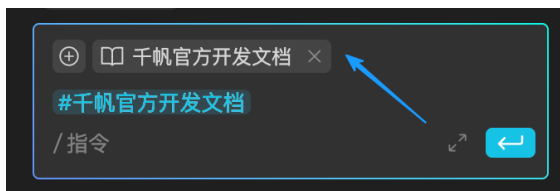
🔗 触发或点击“+”按钮: 在对话框中输入 # 或点击对话框顶部的“+”按钮，即可打开知识选择面板。



灵活选择知识范围: 您可以选择 当前代码文件、整个代码库 等作为知识范围，也可以通过“自定义知识”上传第三方文档，例如公司的编码规范文档，让文心快码基于您指定的知识进行问答。



知识展示与移除: 选中的知识集会出现在对话框顶部, 清晰可见。您可以随时点击知识集旁边的“x”将其移除, 灵活调整上下文范围。 <插入对话框顶部展示已选知识集的截图或占位符, 突出显示知识集和“x”移除按钮>



智能上下文感知: 文心快码还具备强大的智能上下文感知能力, 默认会自动分析您的问题, 并判断需要添加哪些知识作为上下文, 最大程度保证回答与当前环境的相关性和准确性。

### 🔗 代码库理解能力

文心快码拥有强大的代码库理解能力, 让您能够快速掌握项目全局信息。

使用 #当前代码库: 在对话框中输入 #当前代码库, 即可指定整个代码库作为知识范围, 让文心快码基于代码库的全局信息进行检索和分析。

示例: 订单处理逻辑查找: 例如, 当您想了解订单处理逻辑时, 可以在代码问答中输入 #代码库 订单处理逻辑在哪里?。文心快码将深入代码库, 快速定位并告知您订单处理相关的服务接口、服务实现、控制器等关键代码位置。

### 🔗 上下文关联生成

文心快码生成的代码是与当前开发环境高度关联的, 有效避免了“幻觉”代码的产生, 确保生成的代码可以直接应用到您的项目中。

示例: 添加删除订单方法: 例如, 您可以在代码问答中输入 添加删除订单的方法, 接受POST请求, 参数为订单id列表。文心快码会结合当前代码库的上下文, 智能分析并生成贴合您项目结构的实现步骤, 包括:

- 在 Controller 中添加处理删除订单请求的方法。
- 确保 Service 接口中存在删除订单的方法定义。
- 并分别生成 Controller 和 Service 接口的完整代码示例。
- 最终, 文心快码还会生成一个代码关系图谱, 清晰展示各个类之间的关系。

### 🔗 智能采纳

点击“采纳”按钮：在代码问答生成的代码块下方，点击“采纳”按钮。



代码自动合并：点击“采纳”后，您会看到代码文件从上到下有一个刷新效果，这表示文心快码已将生成的代码自动合并到当前文件的准确位置。整个过程快速流畅，位置准确，没有出错。

## 代码补全

代码补全是一种智能辅助功能，它能够根据您当前的代码上下文，自动预测并建议您接下来可能要写的代码，从而显著提高编码效率和准确性。

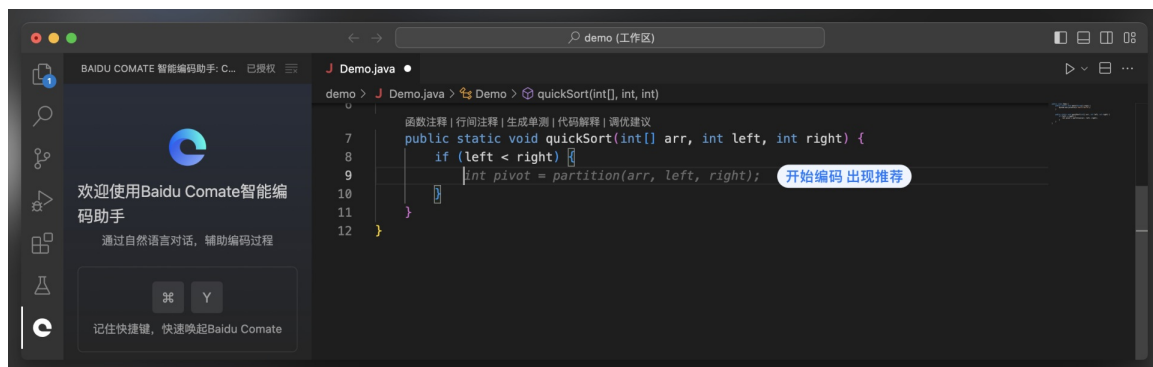
### 单行补全

单行补全是文心快码中最直接、最高效的补全方式。当您打开一个代码文件并开始编码时，只需稍微停顿一下，文心快码就会根据您的代码上下文，自动建议一行代码的补全内容。

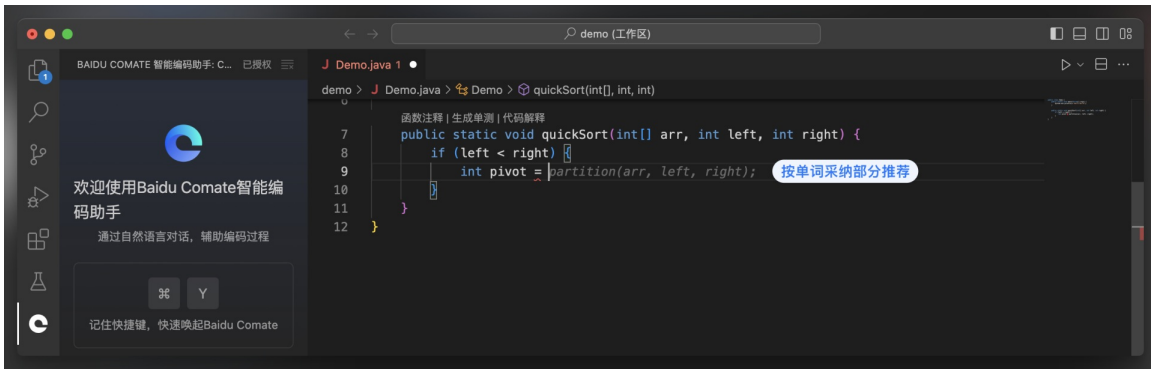
如何使用：

- 如果建议的内容符合您的预期，按 Tab 键接受补全。
- 如果建议的内容不符合需求，按 ESC 键取消补全。
- 您还可以按单词接受部分补全，方便根据您的意图灵活修改代码。

开始编码时推荐



按单词采纳部分推荐



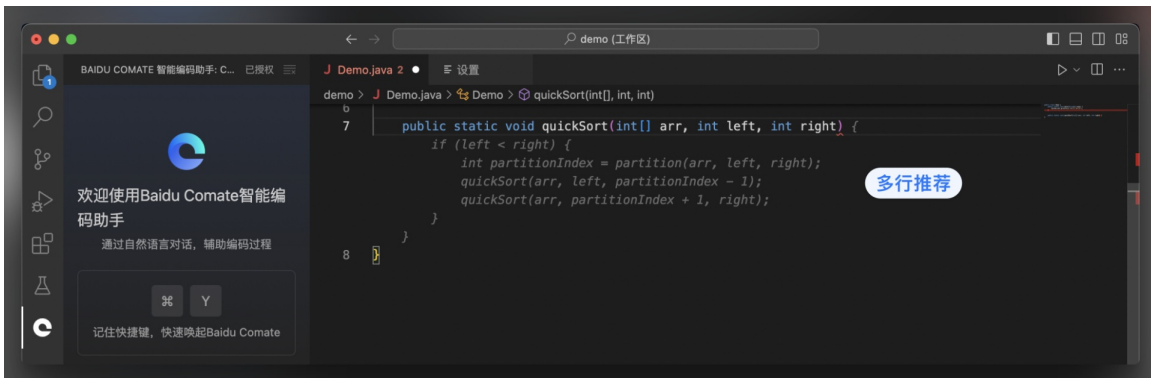
## ☞ 多行补全

多行补全与单行补全的使用方式类似，但在特定场景下（如函数定义 `{` 处、`for` 循环、`if` 判断等），文心快码会优先尝试计算多行补全逻辑。如果逻辑完整，会展示多行补全结果。

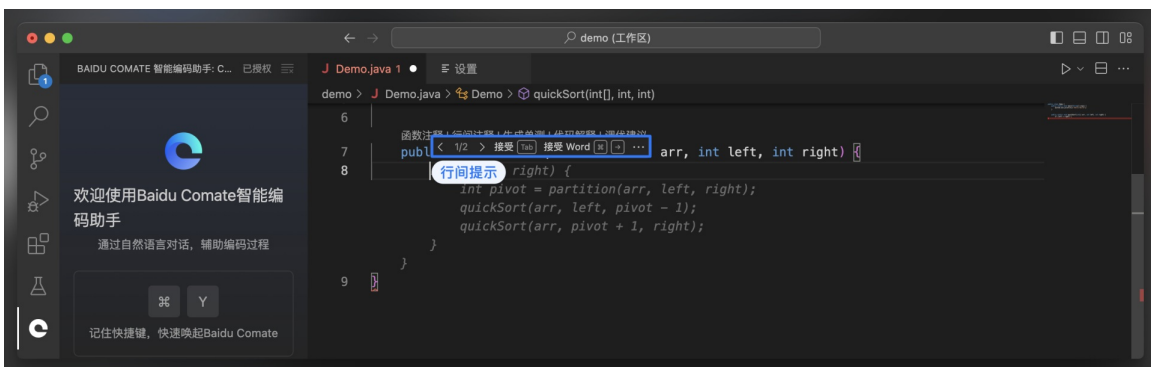
如何使用：

- 如果建议的多行内容符合预期，按 `Tab` 键接受全部补全。
- 如果需要逐步调整，使用快捷键 `⌘ + ↓` 按行接受部分补全。
- 若不符合需求，按 `ESC` 键取消。

多行推荐



行间提示



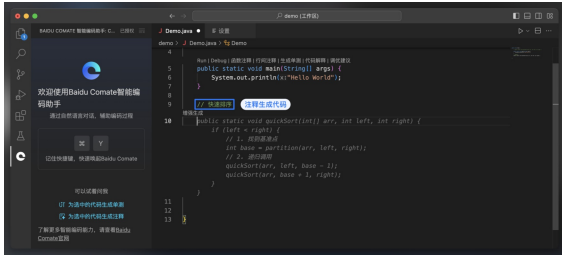
## ☞ 注释生成代码

注释生成代码功能可以根据您写的注释，自动生成相关的代码片段。这一功能特别适用于能用一句话描述清楚需求的情况，或生成常见的算法片段。

如何使用：

- 在代码中输入注释，例如 `# 实现一个快速排序算法。`
- 文心快码会根据注释内容生成相应的代码片段。

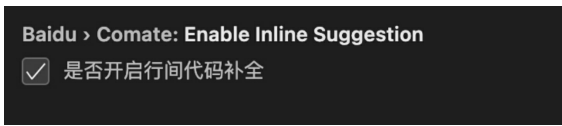
- 按 Tab 键接受生成结果，或按 ESC 键取消。



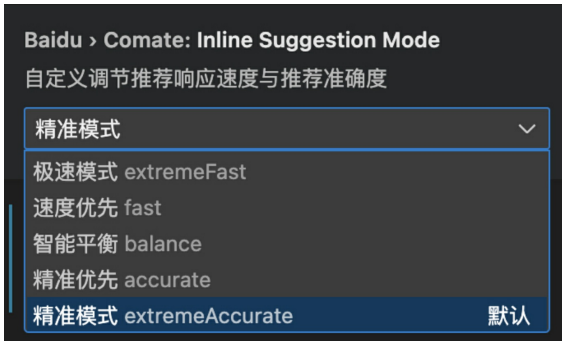
### 可配置项

代码补全提供了一些灵活的可配置项，让您可以根据个人习惯调整补全行为。这些选项可以在设置界面中找到并调整，以下是具体的配置内容及其用途：

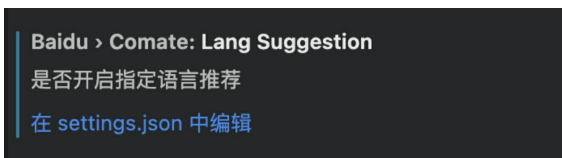
是否开启行间代码补全，选择是否在编码过程中自动触发补全建议。



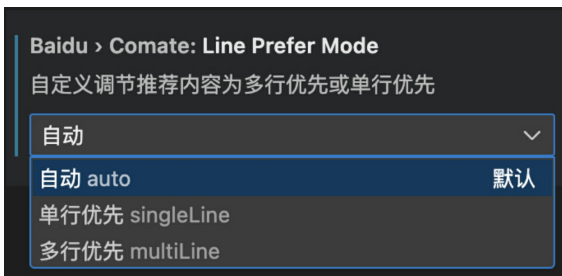
自定义调节推荐响应速度与推荐准确度，根据您的喜好，调整补全建议的响应速度和准确度。



是否开启指定语言推荐，按语言设置是否进行自动补全，例如只在 Python 文件中启用补全功能。



自定义调节推荐内容为多行优先或单行优先，选择补全建议默认提供多行补全还是单行补全。



超级代码补全是一种高级的代码辅助工具，它不仅可以补全代码，还能智能地修改现有代码、预测光标位置，并支持多行代码的改写，帮助开发者更高效地完成编码工作。

目前所有用户均可使用该功能，免费用户将有有限的体验次数。

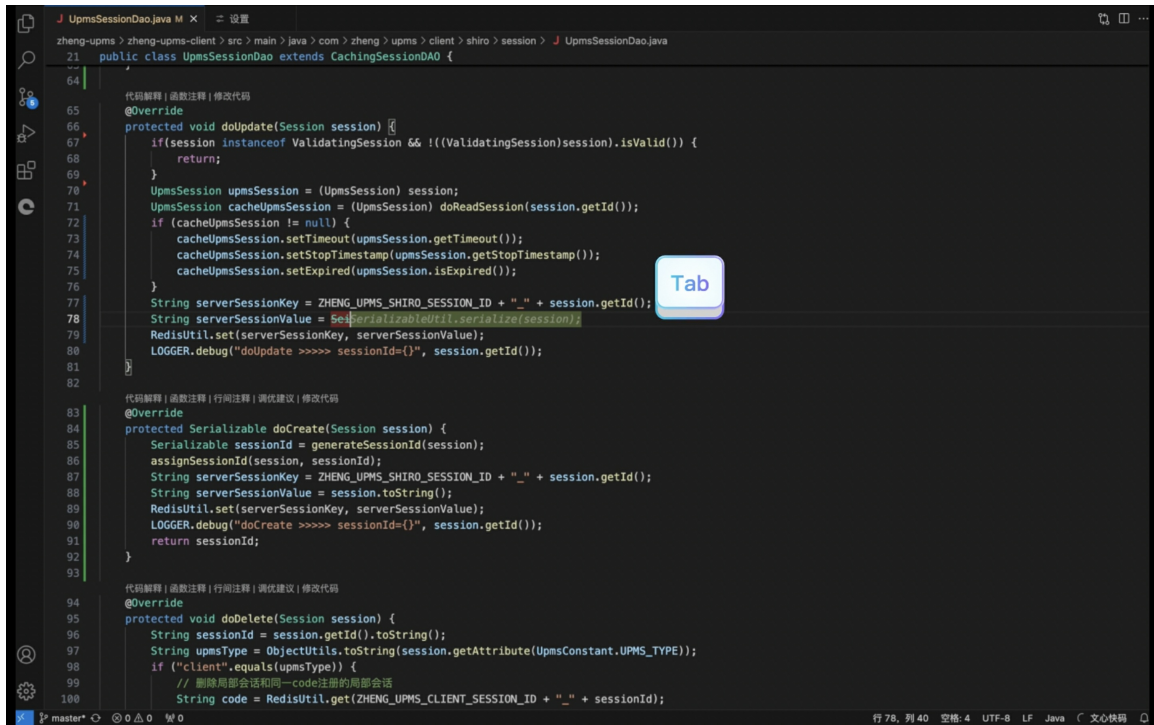
### 代码改写

相比于普通补全仅在光标停留处新增代码，超级代码补全的代码改写功能能够智能地修改现有代码。

如下图示例，当开发者输入一个错误的变量名（如“sei”），超级代码补全会自动检测并建议正确的拼写

(如“SerializableUtil.serialize(session);”)，只需接受建议即可完成修正。

## 输入拼写错误

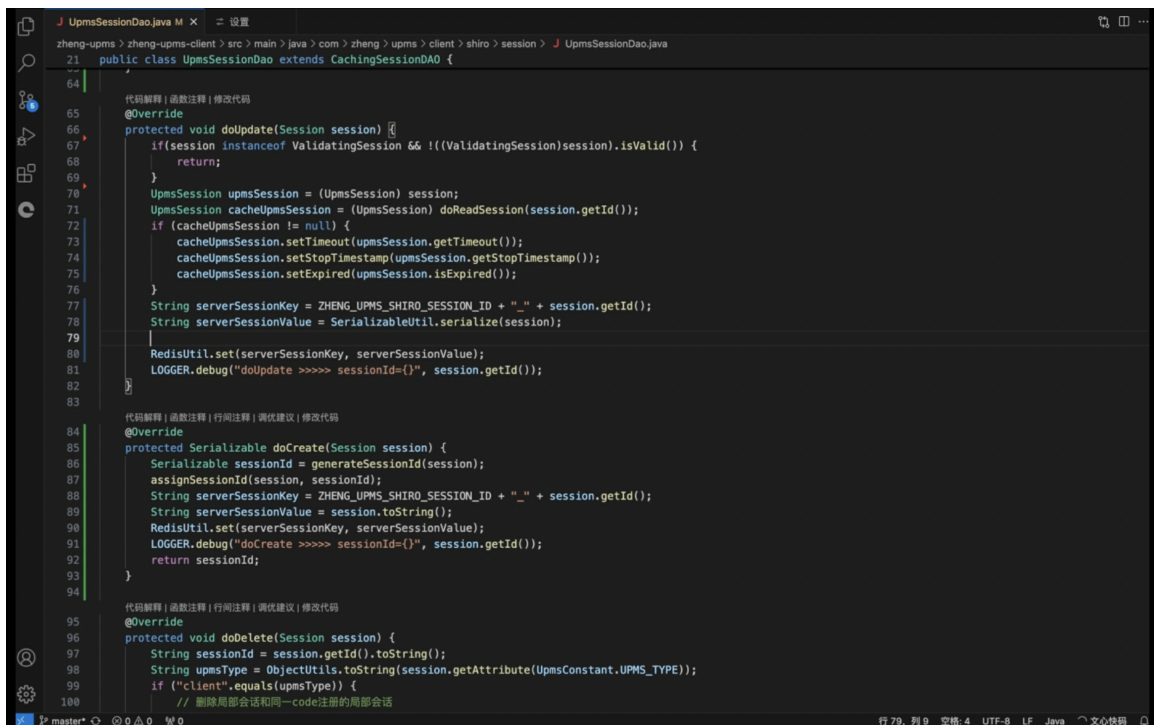


```

21 public class UpmsSessionDao extends CachingSessionDAO {
64
65     代码解释 | 函数注释 | 修改代码
66     @Override
67     protected void doUpdate(Session session) {
68         if(session instanceof ValidatingSession && !((ValidatingSession)session).isValid()) {
69             return;
70         }
71         UpmsSession upmsSession = (UpmsSession) session;
72         UpmsSession cacheUpmsSession = (UpmsSession) doReadSession(session.getId());
73         if (cacheUpmsSession != null) {
74             cacheUpmsSession.setTimeout(upmsSession.getTimeout());
75             cacheUpmsSession.setStopTimestamp(upmsSession.getStopTimestamp());
76             cacheUpmsSession.setExpired(upmsSession.isExpired());
77         }
78         String serverSessionKey = ZHENG_UPMS_SHIRO_SESSION_ID + "_" + session.getId();
79         String serverSessionValue = SerializableUtil.serialize(session);
80         RedisUtil.set(serverSessionKey, serverSessionValue);
81         LOGGER.debug("doUpdate >>>> sessionId={}", session.getId());
82     }
83
84     代码解释 | 函数注释 | 行间注释 | 调试建议 | 修改代码
85     @Override
86     protected Serializable doCreate(Session session) {
87         Serializable sessionId = generateSessionId(session);
88         assignSessionId(session, sessionId);
89         String serverSessionKey = ZHENG_UPMS_SHIRO_SESSION_ID + "_" + session.getId();
90         String serverSessionValue = session.toString();
91         RedisUtil.set(serverSessionKey, serverSessionValue);
92         LOGGER.debug("doCreate >>>> sessionId={}", session.getId());
93         return sessionId;
94     }
95
96     代码解释 | 函数注释 | 行间注释 | 调试建议 | 修改代码
97     @Override
98     protected void doDelete(Session session) {
99         String sessionId = session.getId().toString();
100        String upmsType = ObjectUtils.toString(session.getAttribute(UpmsConstant.UPMS_TYPE));
101        if ("client".equals(upmsType)) {
102            // 删除局部会话和同一code注册的局部会话
103            String code = RedisUtil.get(ZHENG_UPMS_CLIENT_SESSION_ID + "_" + sessionId);
104        }
105    }

```

按下tab修正，并自动补全



```

21 public class UpmsSessionDao extends CachingSessionDAO {
64
65     代码解释 | 函数注释 | 修改代码
66     @Override
67     protected void doUpdate(Session session) {
68         if(session instanceof ValidatingSession && !((ValidatingSession)session).isValid()) {
69             return;
70         }
71         UpmsSession upmsSession = (UpmsSession) session;
72         UpmsSession cacheUpmsSession = (UpmsSession) doReadSession(session.getId());
73         if (cacheUpmsSession != null) {
74             cacheUpmsSession.setTimeout(upmsSession.getTimeout());
75             cacheUpmsSession.setStopTimestamp(upmsSession.getStopTimestamp());
76             cacheUpmsSession.setExpired(upmsSession.isExpired());
77         }
78         String serverSessionKey = ZHENG_UPMS_SHIRO_SESSION_ID + "_" + session.getId();
79         String serverSessionValue = SerializableUtil.serialize(session);
80         RedisUtil.set(serverSessionKey, serverSessionValue);
81         LOGGER.debug("doUpdate >>>> sessionId={}", session.getId());
82     }
83
84     代码解释 | 函数注释 | 行间注释 | 调试建议 | 修改代码
85     @Override
86     protected Serializable doCreate(Session session) {
87         Serializable sessionId = generateSessionId(session);
88         assignSessionId(session, sessionId);
89         String serverSessionKey = ZHENG_UPMS_SHIRO_SESSION_ID + "_" + session.getId();
90         String serverSessionValue = session.toString();
91         RedisUtil.set(serverSessionKey, serverSessionValue);
92         LOGGER.debug("doCreate >>>> sessionId={}", session.getId());
93         return sessionId;
94     }
95
96     代码解释 | 函数注释 | 行间注释 | 调试建议 | 修改代码
97     @Override
98     protected void doDelete(Session session) {
99         String sessionId = session.getId().toString();
100        String upmsType = ObjectUtils.toString(session.getAttribute(UpmsConstant.UPMS_TYPE));
101        if ("client".equals(upmsType)) {
102            // 删除局部会话和同一code注册的局部会话
103            String code = RedisUtil.get(ZHENG_UPMS_CLIENT_SESSION_ID + "_" + sessionId);
104        }
105    }

```

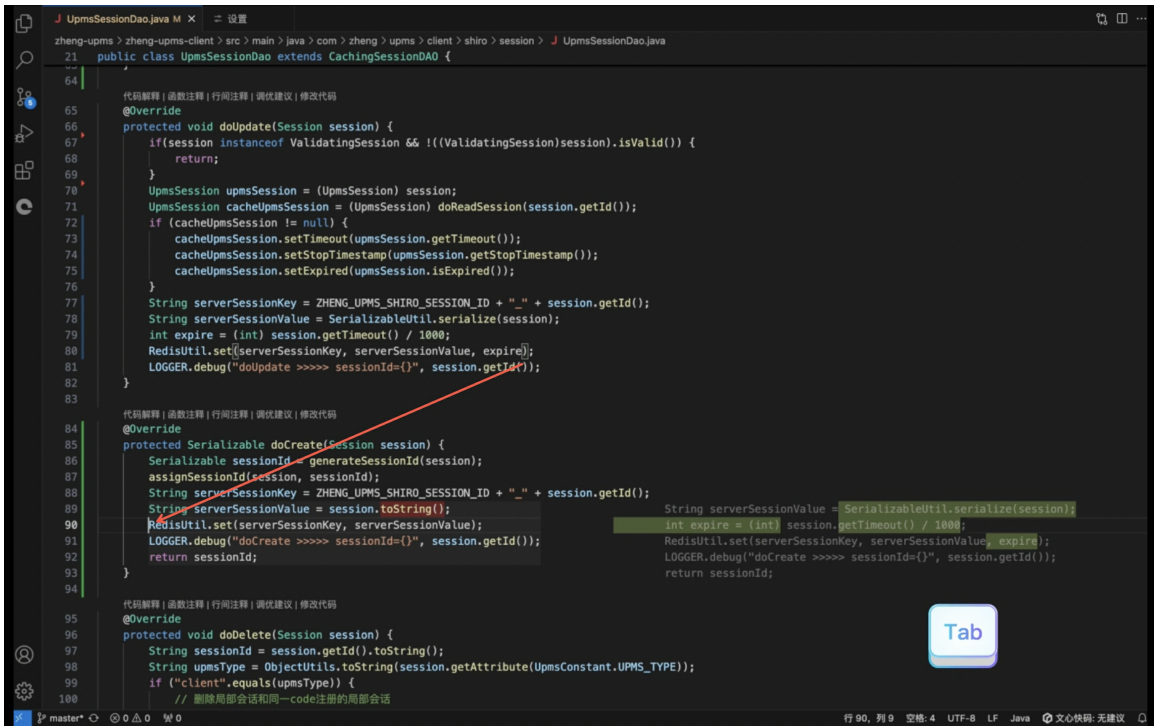
## 如何使用

- 在编码时，若出现拼写错误或需要调整代码，超级代码补全会自动提供修改建议。
- 通过键盘确认（Tab键）接受建议，代码即刻得到修正和补全。

## 👁️ 光标预测

超级代码补全能根据您的编辑历史，智能预测下一处需要修改的代码位置。只需按下 Tab 键，光标就会自动跳转到预测位置，减少手动寻找的麻烦。

下图示例中，开发者修改了一段代码后按下 Tab 键，光标自动跳转到下一处需要调整的相似代码块，省去了手动定位的时间。



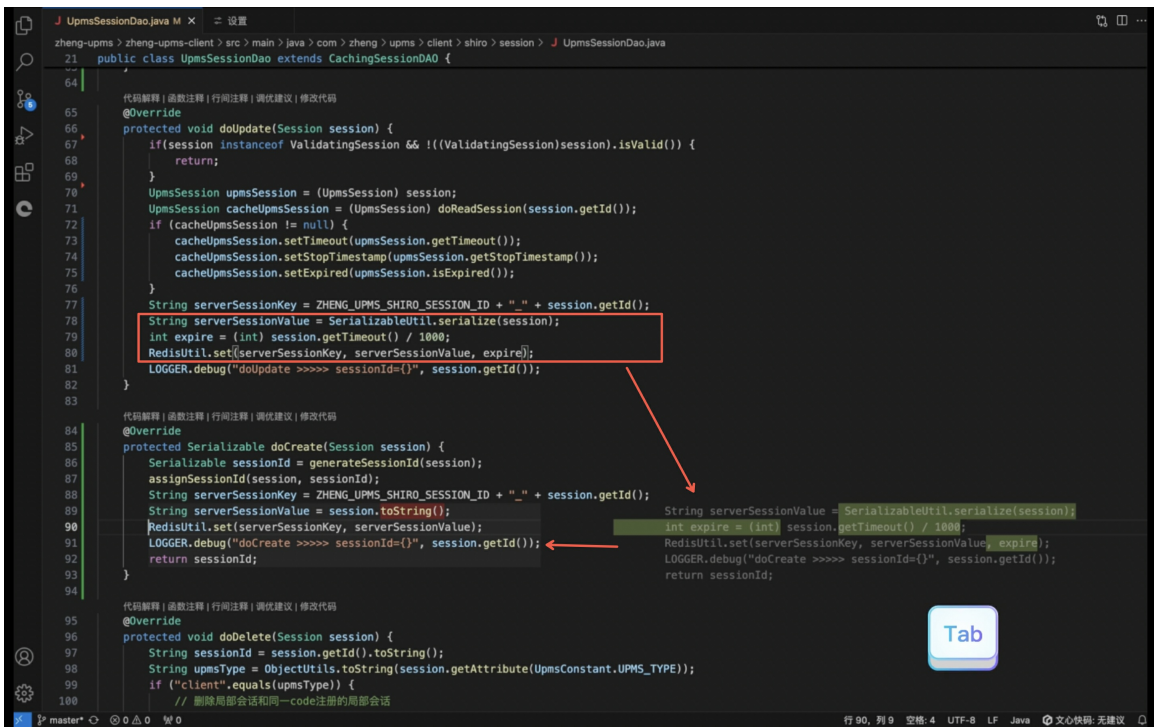
### 如何使用

- 完成当前代码修改后，按下 Tab 键。
- 超级代码补全会根据上下文预测并移动光标至下一处编辑位置。

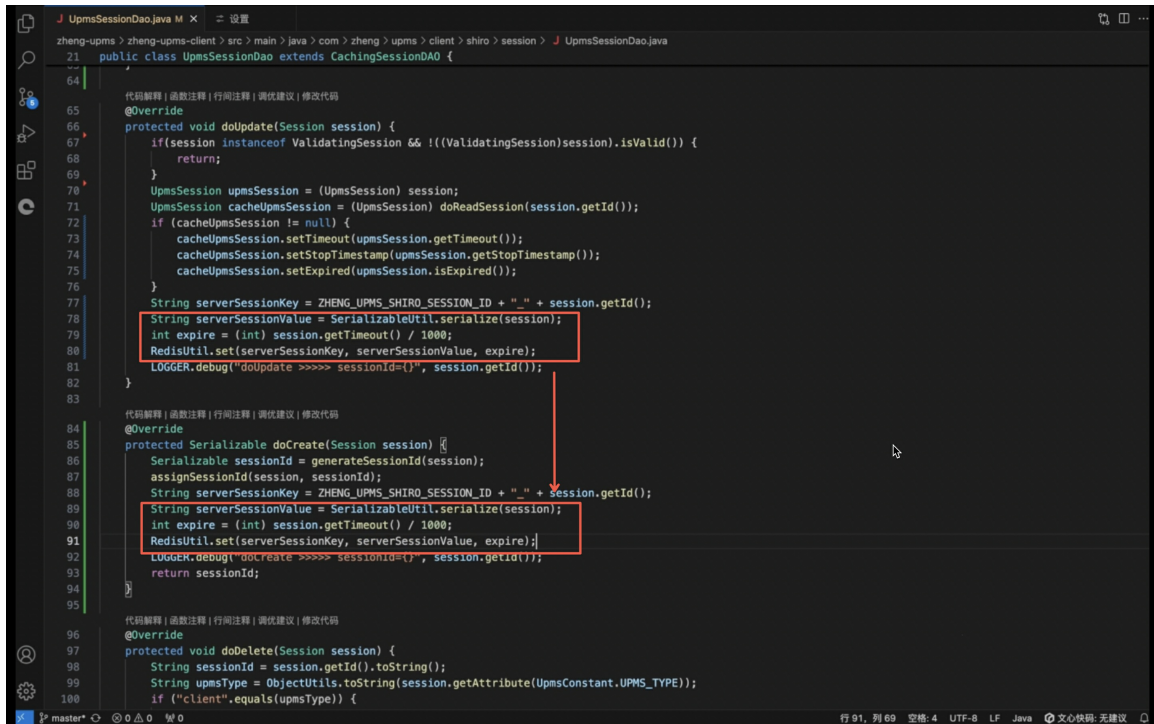
### 多行改写

当您在多处类似的代码块中手动修改了第一处后，只需按下 Tab 键，超级代码补全会自动将相同的修改应用到其他相关代码块，大幅减少重复工作。下图示例中，开发者对代码块进行了3处修改，按下 Tab 键，超级代码补全会自动将相同的修改应用到第二个相似的代码块。

### 预览第一个代码库的修改



按下tab后应用到第二个相似代码块



## 如何使用

- 手动修改一处代码后，按下 Tab 键。
- 超级代码补全会识别相似代码块并自动完成其他部分的改写。

## 智能补全

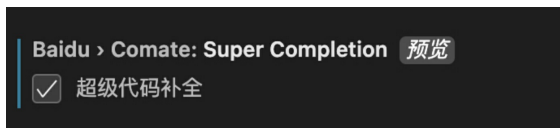
智能导入补全帮助用户在复杂的代码库中快速完成引入、修改和跨文件协同，减少重复劳动，提高开发效率。在代码编辑时，系统会自动弹出补全建议。按 Tab 键采纳结果，可直接跳转到对应文件或位置。按 ESC 取消补全。

适用范围如下：

- **Import补全**自动识别相同代码库、三方依赖和业务依赖，提供快速导入建议，避免手动查找和遗漏。
- **跨文件补全**支持跨文件跳转与修改，当方法名或类发生变更时，可联动多个文件同步更新，减少维护成本。
- **相似实现推荐**当检测到已有相似文件或逻辑时，推荐参考修改，保持风格与实现一致。
- **多文件协同**针对单一需求涉及多文件的情况，自动生成修改清单，并提供统一修改建议，避免遗漏。

## 可配置项

超级代码补全功能支持个性化设置，您可以根据需要选择开启或关闭。

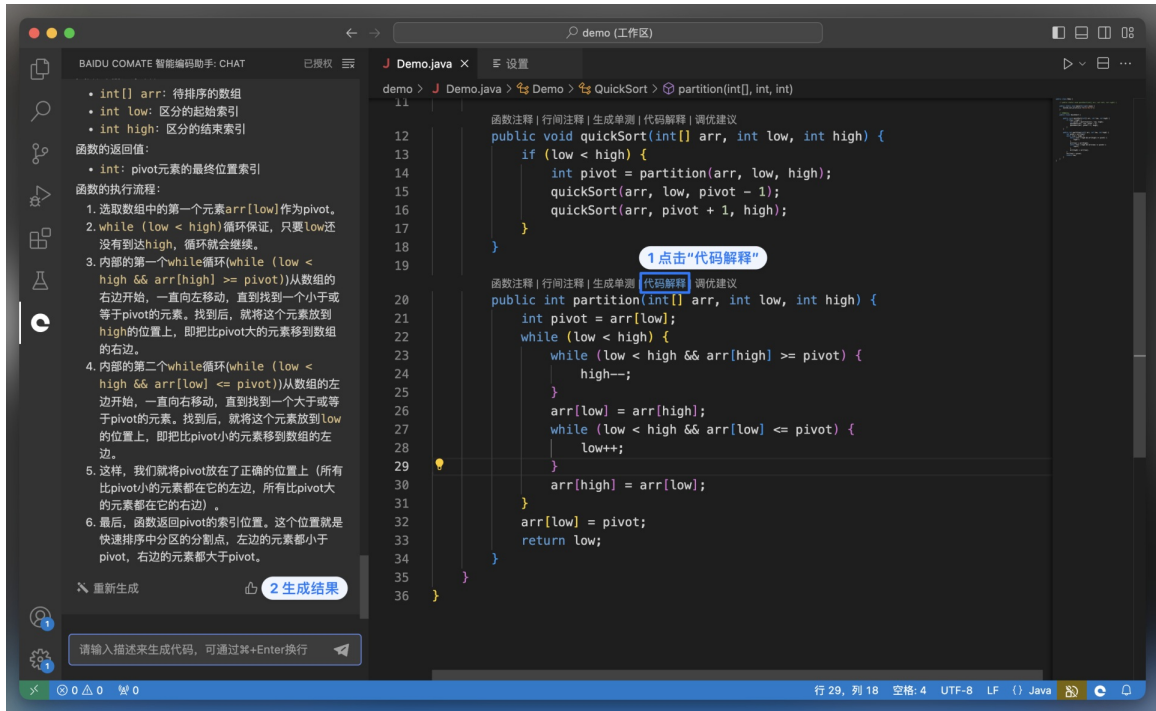


## 行间命令

行间命令，顾名思义，是在代码行间，特别是函数和方法定义上方显示的快捷操作入口。文心快码将一系列常用的代码辅助功能，如代码解释、注释生成、日志添加、调优建议、函数拆分和单测生成等，以简洁的文字或按钮形式展现在您的代码编辑界面。您无需离开代码编辑器，即可快速触发这些功能。

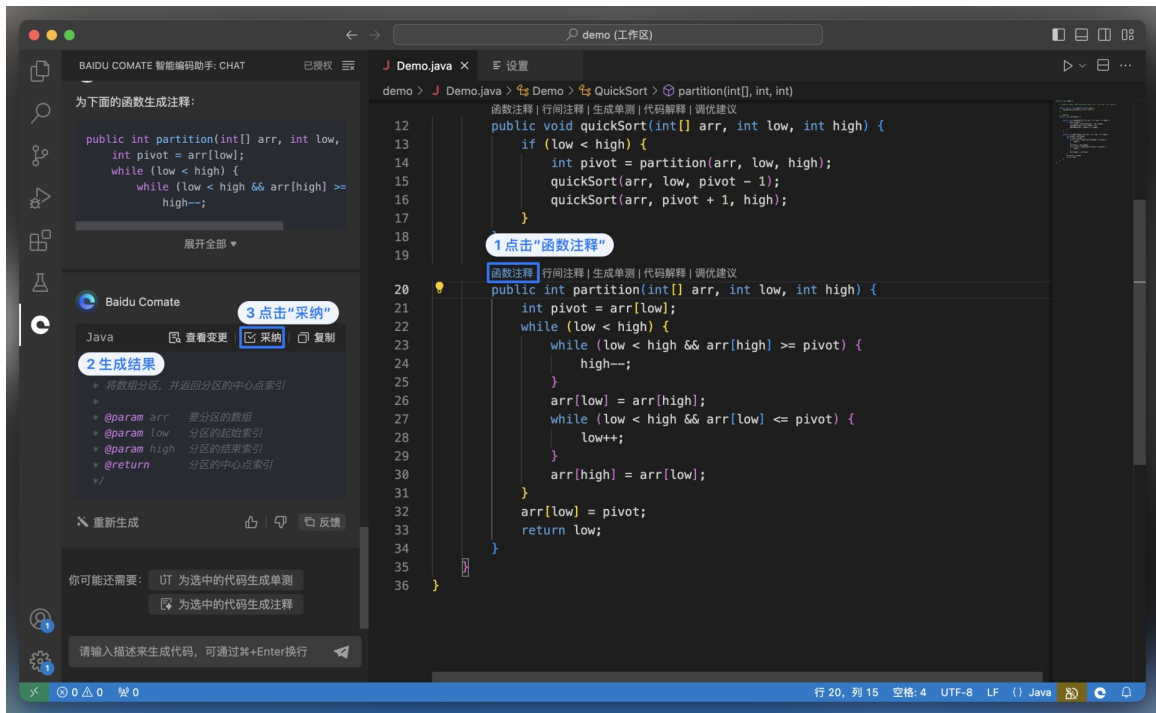
## 代码解释

代码解释使用先进的大语言模型，深入分析您选定的代码片段（目前主要针对函数或方法），并生成清晰易懂的自然语言解释。当您接手 legacy 代码、阅读复杂逻辑或快速了解他人代码时，这项功能将大大节省您的时间和精力，助您快速掌握代码脉络。



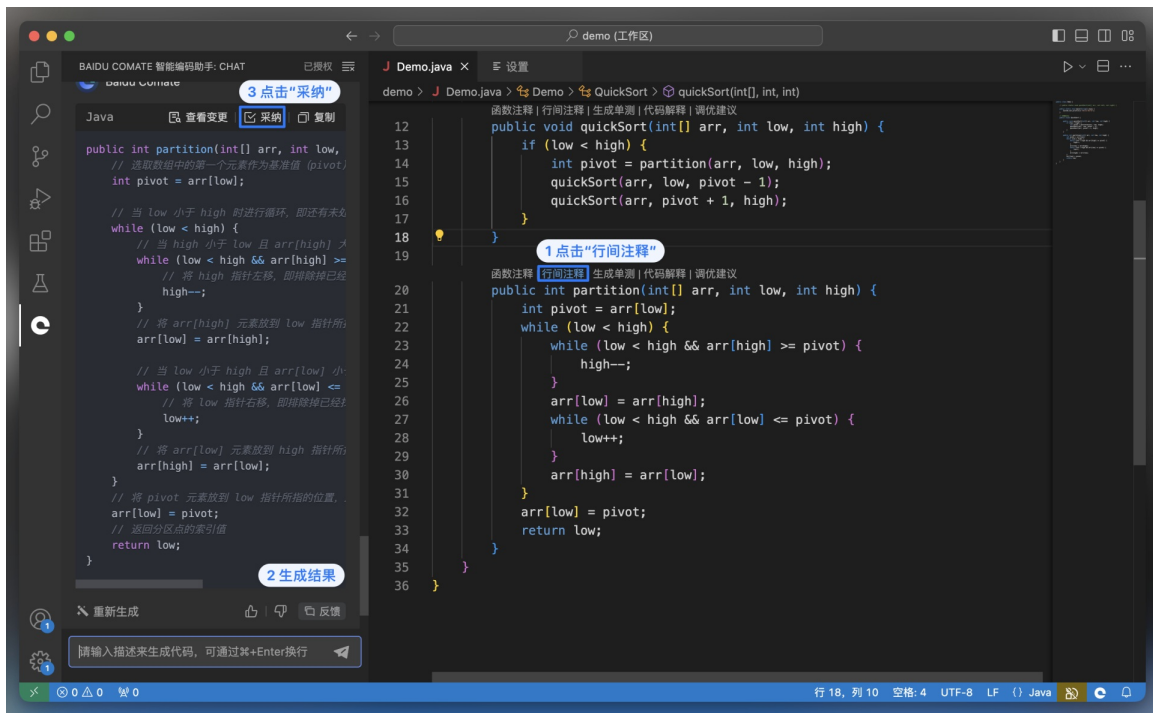
### 函数注释

函数注释 功能为函数或方法生成文档注释（Docstring），即位于函数定义下方，用于描述函数功能、参数、返回值等的注释。



### 行间注释

行间注释 功能为代码块或段落生成行内注释，帮助解释代码的具体逻辑和实现细节，使代码更易于理解和维护。请注意：当函数或方法的代码字符数少于 4500 个字符时，此功能可用。

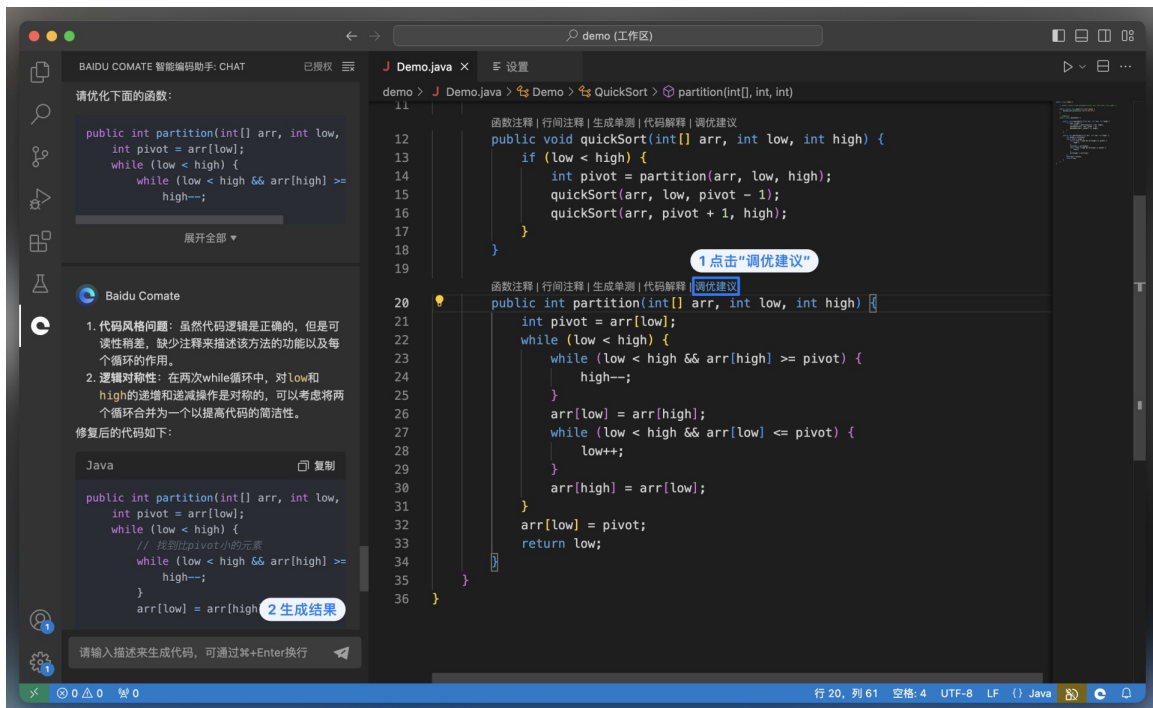


### 添加日志

调试是开发过程中不可或缺的环节。添加日志功能让您能够快速在代码中插入日志语句，用于追踪程序运行状态、排查错误，而无需手动编写繁琐的日志代码。

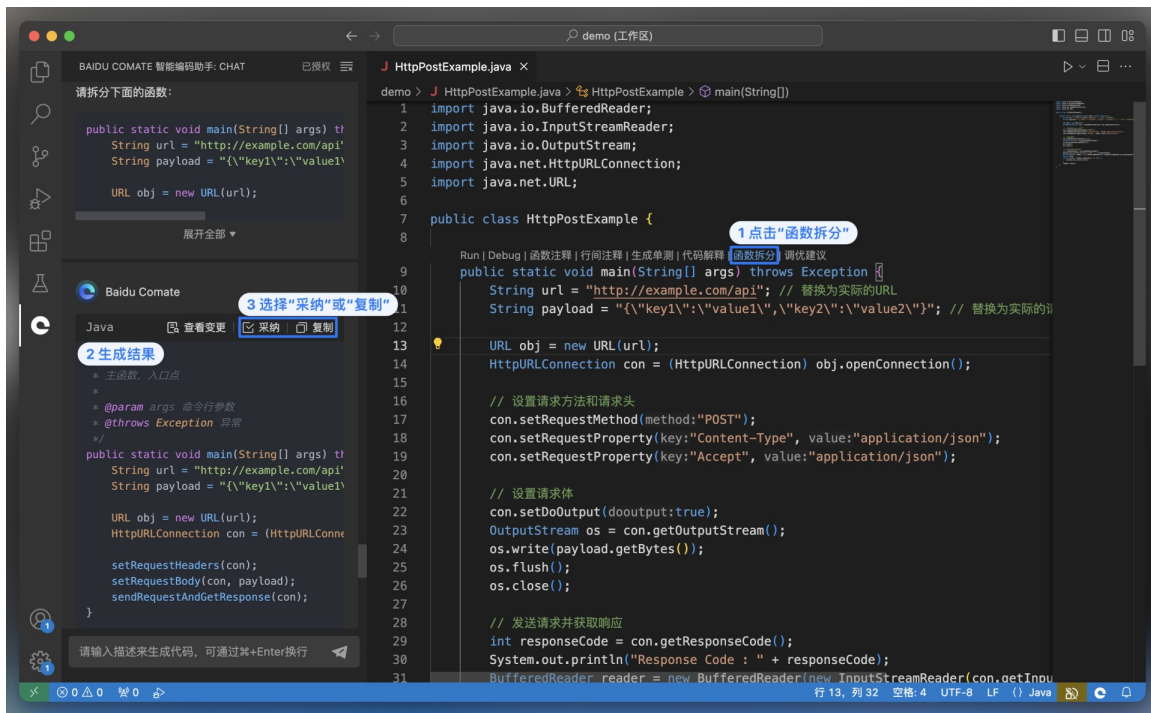
### 调优建议

调优建议利用大语言模型深度分析您选定的函数或方法，从代码规范、可读性、执行效率、资源消耗等多个维度，为您提供有针对性的优化建议，帮助您编写更高质量的代码。



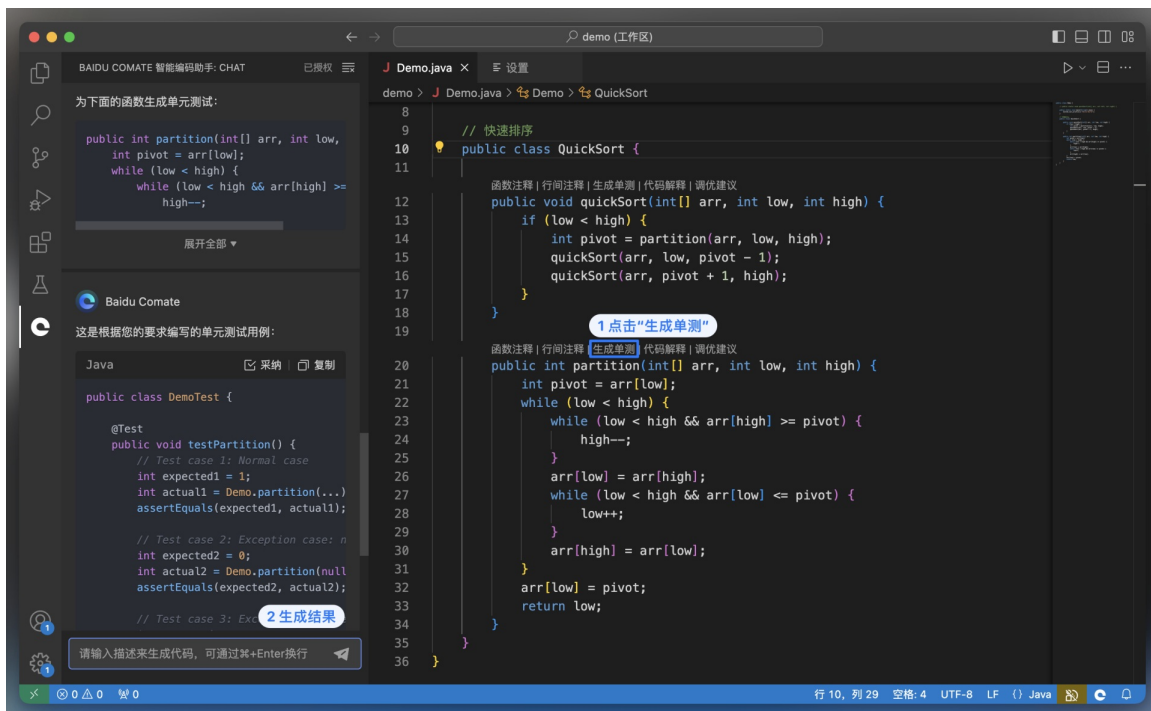
### 函数拆分

当函数或方法过于冗长时，会降低代码的可读性和可维护性。函数拆分功能能够智能分析长函数，将其拆分成多个更小、更专注的函数或方法，使代码结构更清晰，逻辑更易理解，也更方便复用和维护。请注意：当函数或方法代码行数大于等于 20 行，且字符数少于 4500 个字符时，此功能可用。



## 生成单测

单元测试是保障代码质量、预防 bug 的重要手段。生成单测 功能可以针对您选定的函数或方法，自动生成相应的单元测试代码框架，您只需在此基础上完善测试用例，即可快速构建完善的单元测试体系。



## 行间对话

通过快捷键 CMD+I (macOS) 或 Ctrl+I (Windows/Linux)，您可以在代码编辑区的任意位置唤出一个命令输入框，输入您想要的指令，文心快码将根据您的指令和上下文代码，智能生成结果。

例如，代码中有一个未使用的perm\_dict变量，您可以选中这段代码，输入 将perm\_dict这个变量使用起来，文心快码会直接在编辑器中刷新代码，省去侧边栏提问和采纳的流程。



## 配置选项

行间命令按钮样式：

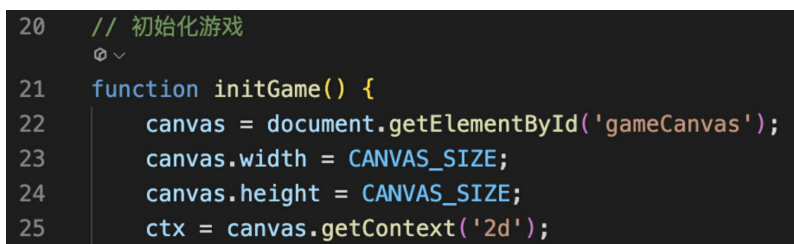
- 文字平铺在函数名上方：默认样式，以简洁的文字形式直接显示在函数名上方，例如“代码解释”、“函数注释”等。这种方式简洁直观，不占用过多空间。



- 显示为按钮，点击展开菜单：另一种样式，行间命令以一个按钮形式显示在函数名上方，点击按钮后展开命令功能菜单。这种方式更加紧凑，界面更整洁，适合需要更少干扰的用户。



行间命令功能可见性：您可以自由配置哪些行间命令功能在您的编辑器中显示。例如，如果您主要关注代码解释和注释生成，您可以只显示“代码解释”、“函数注释”和“行间注释”功能，隐藏其他功能，以简化界面。



## Mission Mode

Mission mode 全新上线：从“关注执行”到“专注目标”，你的研发工作台进化啦！

Comate AI IDE 不再是传统意义上的代码编辑器

全新 Mission Mode（任务模式），让用户从“关注执行”，转变为“专注目标”

维度	Code Mode (传统模式)	Mission Mode (任务模式)
你的关注	AI 执行进度	任务完成进度
你看到的	代码结构、文件树	任务列表 + 对话 + 知识
你的角色	描述目标、关注执行、检查 AI 代码	描述目标、AI 全程推进、专注结果
基本单元	文件 / 分支 / PR	任务

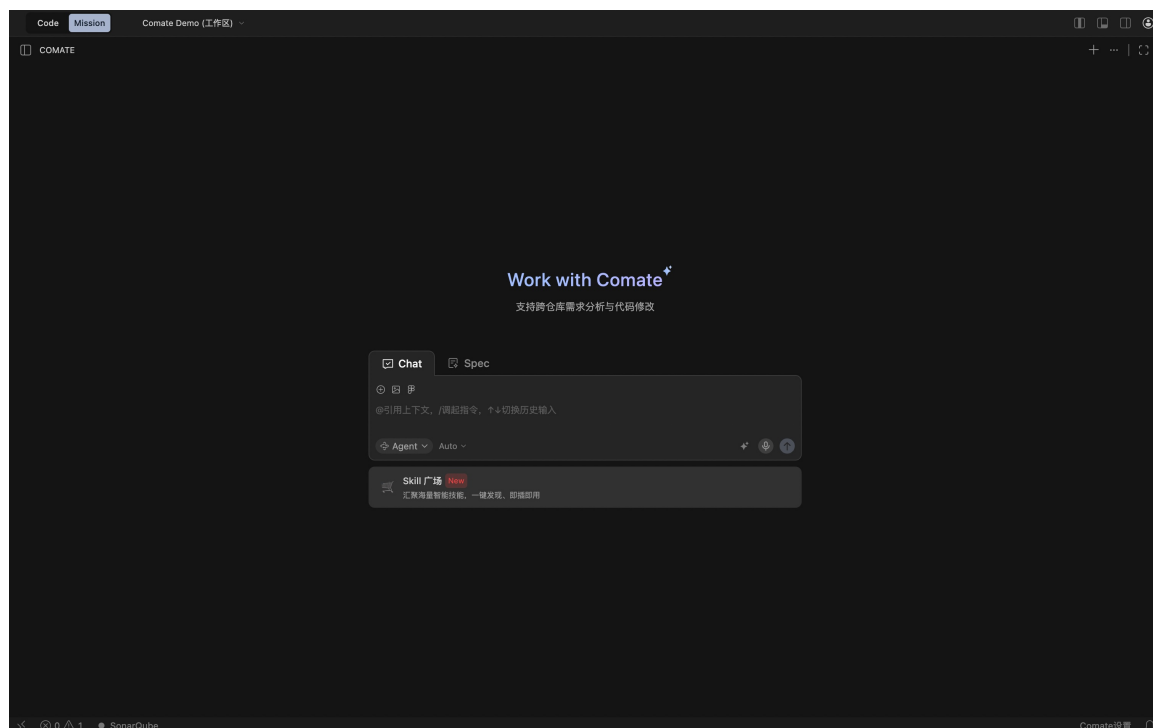
**\*\*一、Mission Mode 是什么？\*\*** 传统模式下，工程师围绕文件、分支、PR 工作，AI 只是“编程助手”

**Mission Mode 则颠覆了这一逻辑：**

以前：人围绕代码工作，Agent 辅助完成代码开发

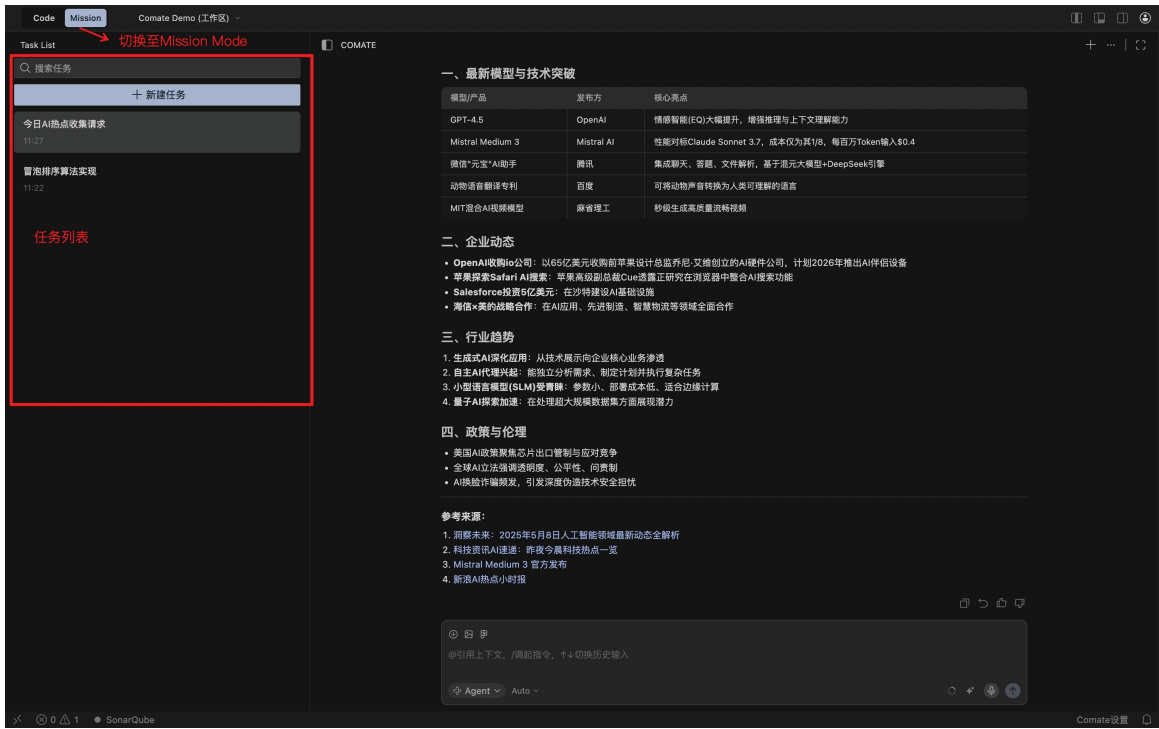
现在：人只负责提出目标，Agent 完成任务全流程推进

随着模型能力增强，工程师的核心竞争力，正从“写代码的速度”转向“高质量组织任务、控制风险、加速交付”，Mission Mode 正为此而生。

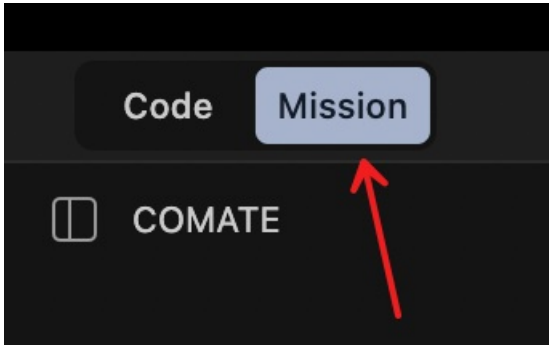


**\*\*二、Mission 界面介绍\*\*** 在Commate IDE 左上角进行Code/Mission 模式的切换，进入 Mission Mode 后，你会看到一个焕然一新的布局：

- 左侧：任务列表 + 文件管理
- 中部：对话区 + 代码变更预览
- 底部 / 右侧：Terminal、插件列表等

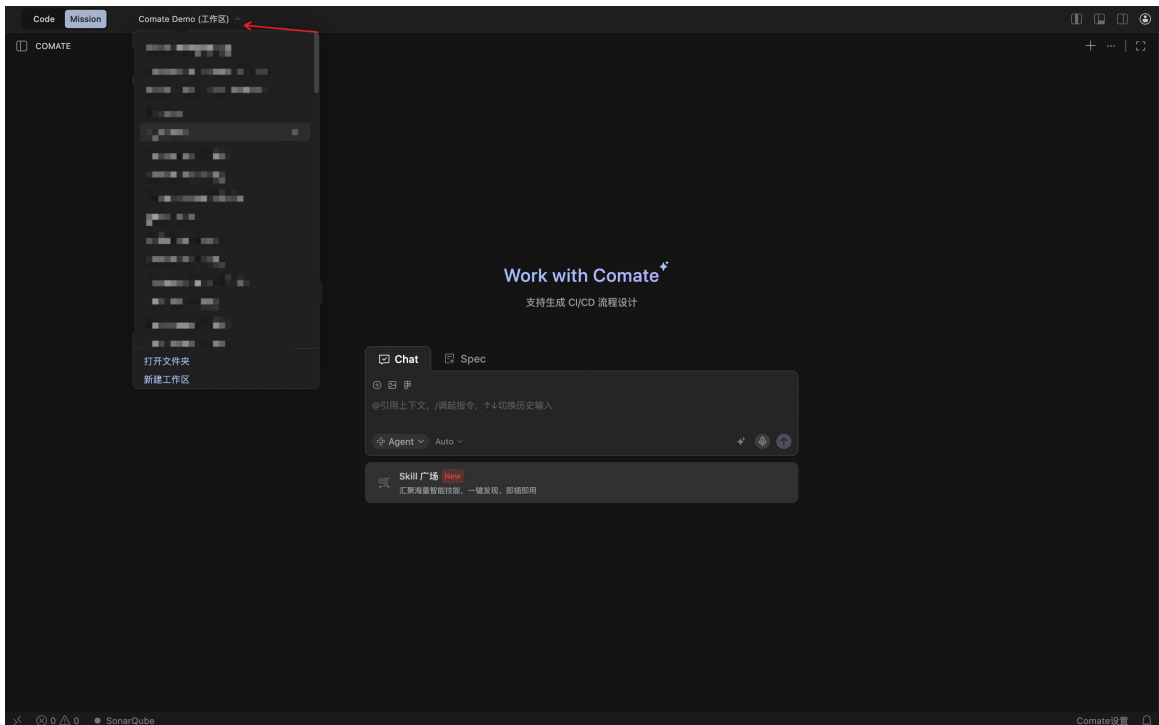


三、快速开启第一个任务 Step 1：切换到 Mission Mode IDE 左上角点击模式切换按钮，进入任务模式

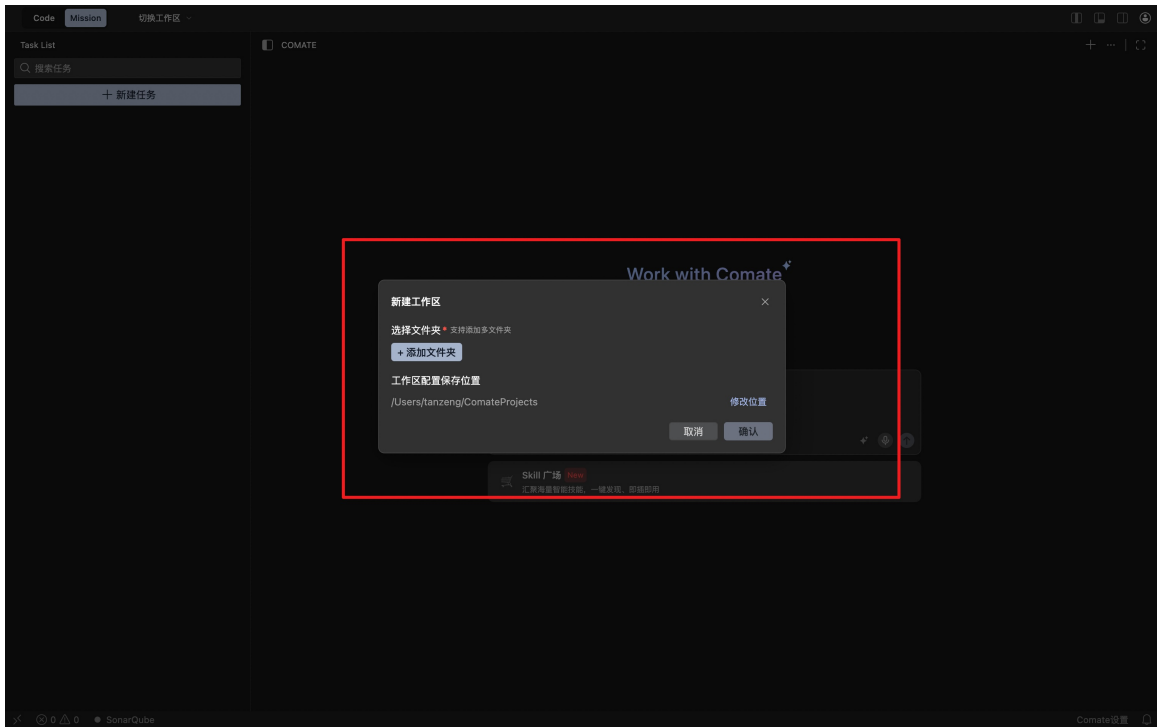


Step 2：创建 / 选择工作区 (Workspace)

- 创建入口1：Mission Mode 视图内通过「切换工作区」新建

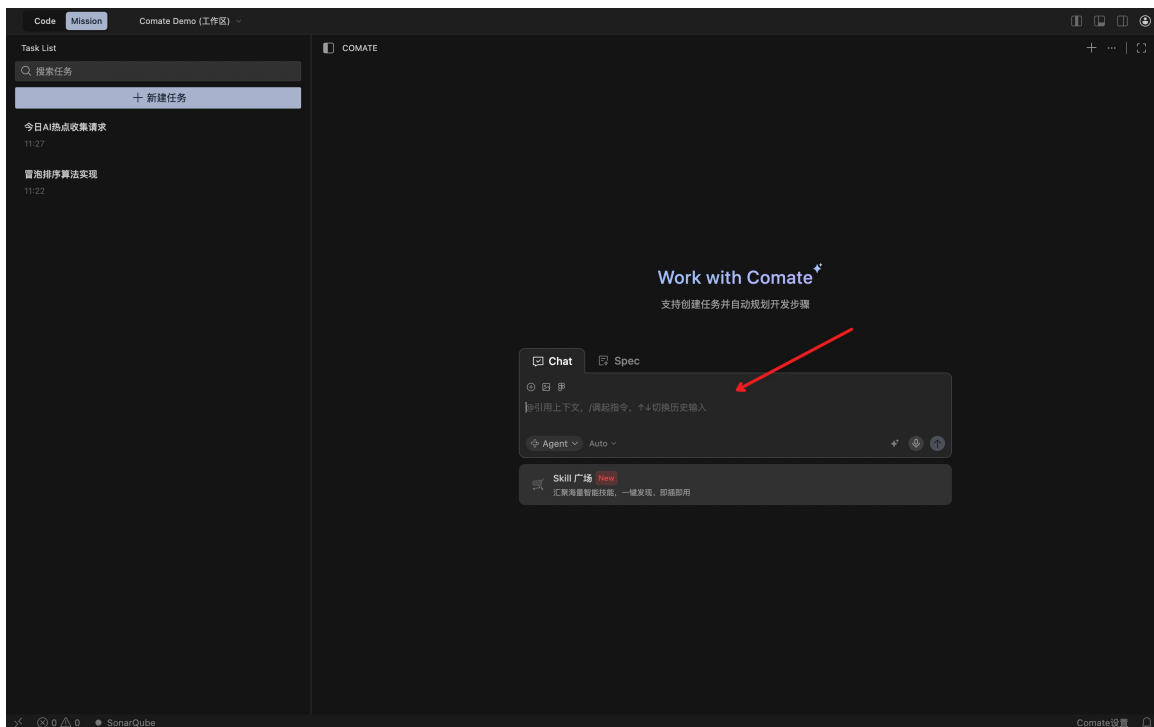


- 创建入口2：初始界面新建



### Step 3 : 描述任务

在对话区输入任务，例如：“帮我收集今天的AI 热点资讯” Agent 自动执行，任务列表同步更新进度



### Step 4 : 查看进度与结果

- 左栏任务列表实时追踪状态（运行中 / 已完成 / 待确认）
- 任务完成时系统弹窗提示（多窗口也能知道是哪个项目更新）
- 前端任务自动打开浏览器预览
- 对话区上方展示变更文件列表，随时 review diff

## AI IDE

### Spec模式

背景 要让 Agent 发挥更大价值，核心是让它聚焦代码编写，人承担监督角色。但现在真实场景中，人缺乏有效指导 Agent 写代

码的手段。

因此我们给出解决方案：人多投入文档撰写的过程，负责核验 Agent 产出的文档与代码是否准确。

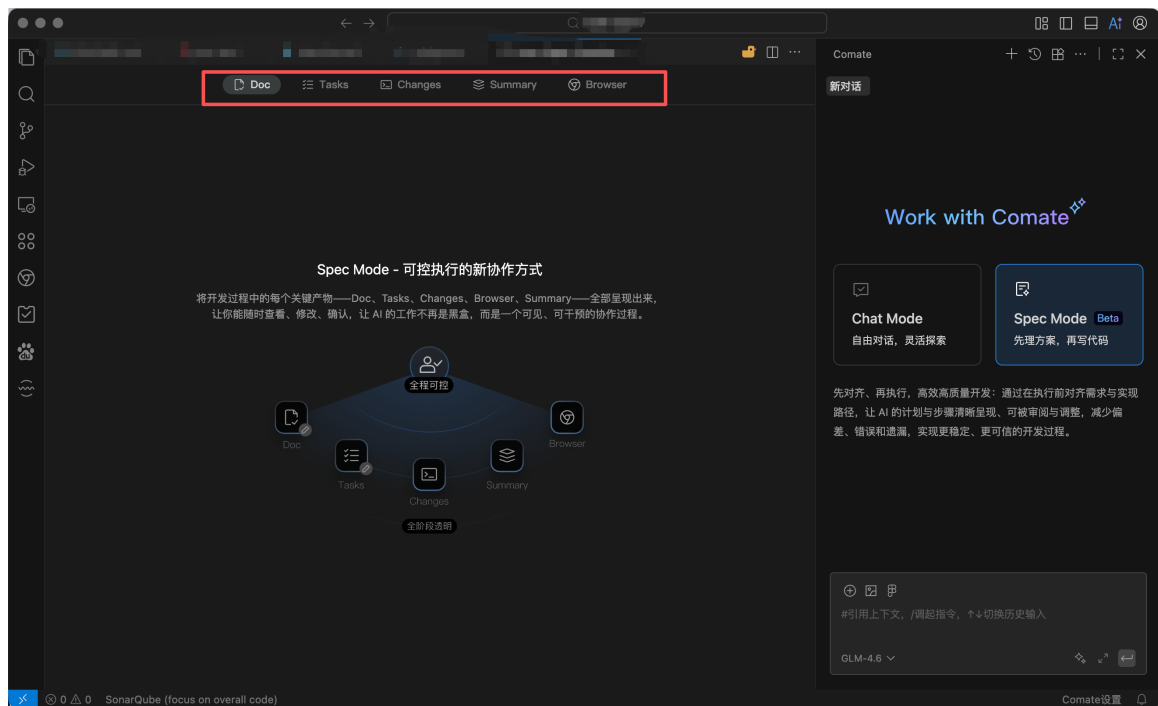
### 把错误尽可能拦在 Agent 写代码之前

**智能体Spec模式** Spec模式是一种新的协作方式：让 Agent 先把它的理解与计划写成文档，把要做的事情拆成任务，经过用户确认后，再进行相应的代码编写。

从而让Agent的代码生成质量和效果大幅提升。

**具体方案** 各阶段以产物视图 (Product View) 呈现，共设六个 Tab (越往上，用户需要关注的优先级越高)。用户从关心代码转变为只需要关心开始的文档和任务以及最后的结果

- 文档Doc：需求目标、实现方案说明 (需确认)
- 任务Tasks：任务拆解与执行计划 (需确认)
- 代码变更Changes：执行阶段的代码变更可视化与验证
- 网页预览Preview：可视化前端或最终成果预览
- 验证Verify (待实现)：通过自动化测试保证任务成功
- 任务总结Summary：任务总结与交付结果!



- **流程策略**：不显性展示固定流程，而以“隐性引导”推进 (Doc → Tasks → Summary)，确保灵活性；特殊情况下允许模型智能跳过环节。
- 聚焦“**单轮完整任务**”流程：用户在确认 Doc 与 Tasks 后，暂不支持回溯修改这两个环节，保证流程简洁清晰。

**效果** 任务一次成功率显著提升、返工大幅减少、代码质量更稳。

- ① Spec 解决了“做错”问题：需求理解错误，代码也就错误；需求理解了，代码修改位置不对，代码也写错了
- ② Spec 解决了“做漏”问题：Agent考虑不周，未有效处理分支和边界条件
- ③ Spec 解决了“做偏”问题：实现了功能，但实现方式不符合用户预期\*\*

### 🔗 基于Github的真实需求Case演示

#### 如何通过Spec模式中解决Agent 需求考虑不周，任务执行偏差的问题

原始query：

“基于nageoffer/12306代码库”

我在公共演示的时候，不希望现场的观众能够使用用户注册，修改信息、下单等功能扰乱我系统中原本的数据。所以我想增加一个demo模式，能够通过配置中设定这个demo为true，并配置写操作路径黑名单以及返回状态码、返回消息，就能够拦截这些请求

正确路径：

/api/user-service/passenger/save

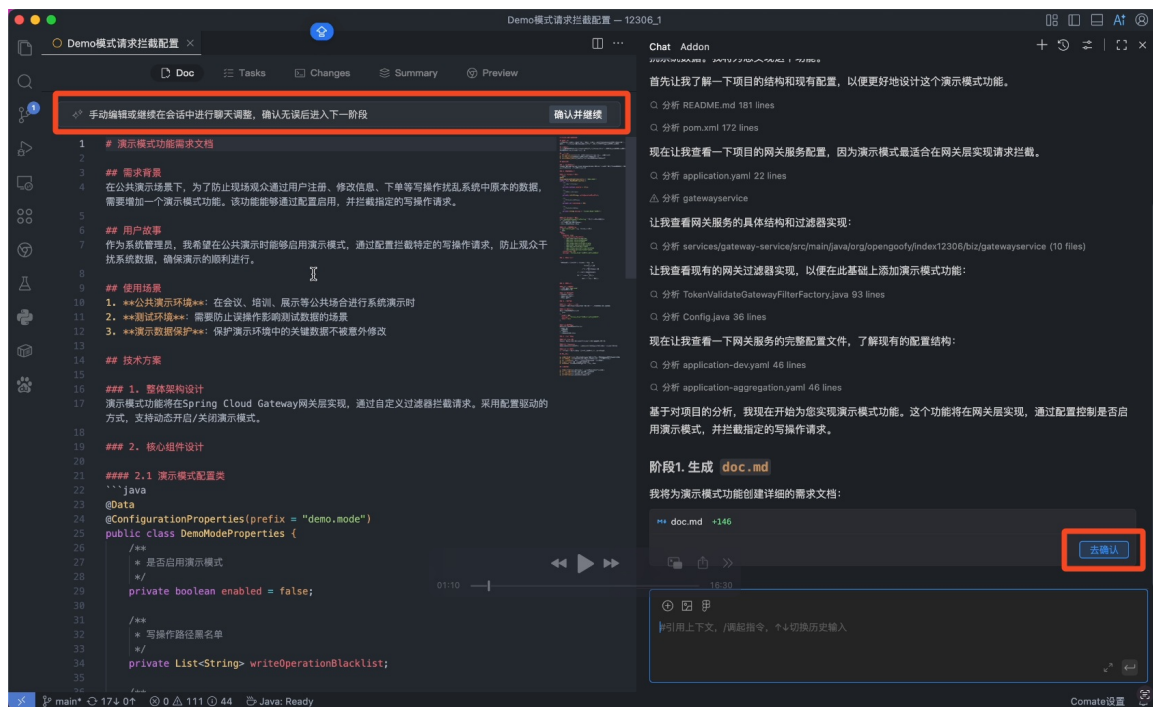
/api/user-service/passenger/update

/api/user-service/passenger/remove

/api/user-service/update

/api/user-service/deletion

Spec模式如何解决 Spec 模式通过让用户Doc确认解决了Zulu的问题：Spec 在文档阶段（仅 3 min）就能提前识别问题



Spec 在

开始执行前会生成 **Doc**，其中明确列出各种细节：

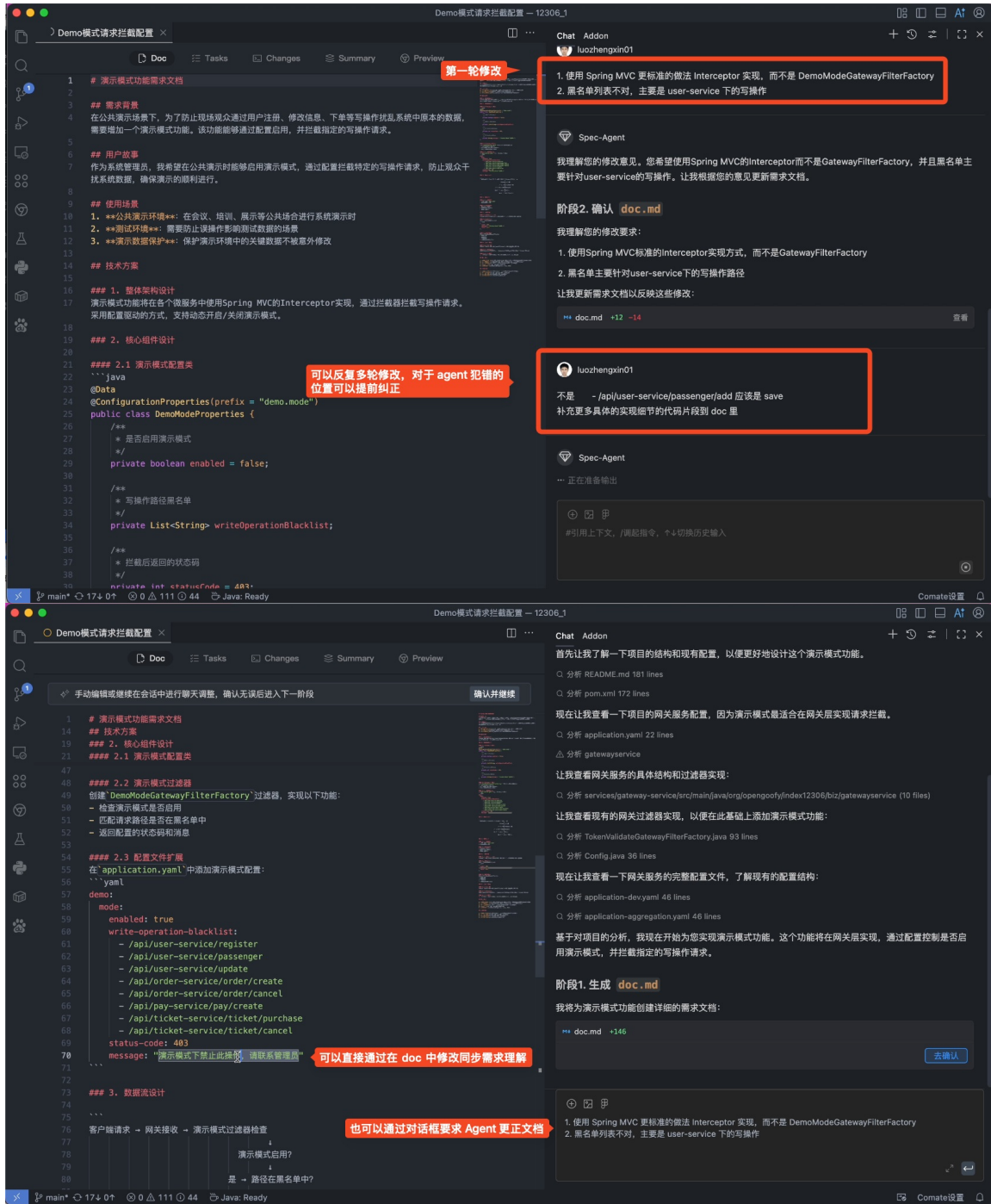
- 需要拦截的路径
- 服务边界
- 演示环境下哪些模块不能触碰
- 可能的风险点
- 精准的范围列表
- 用户可编辑、可删改的范围说明

然后用户需要 **确认 Doc** 才能进入下一步。

\*1) 通过Doc，用户发现不需要拦截的路径不对，快速的和Agent对齐意图在 Spec 的 Doc 里，AI 不再“默默选一个方案”，而是：

- 列出不同实现方式（例如：基于 Spring MVC Interceptor / 基于自定义 Filter / 基于配置驱动）；
- 标明每种方案对 **可维护性、扩展性、性能、框架一致性** 的影响；

• 用人能读懂的话解释差异。

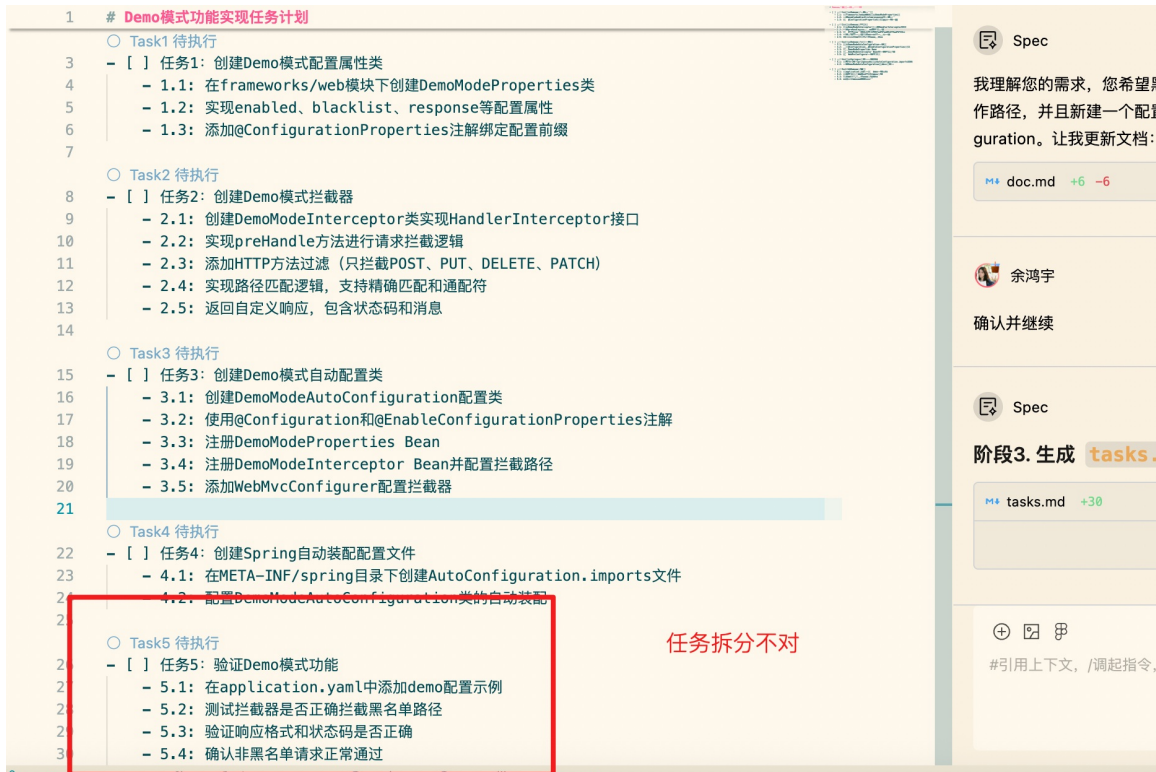


以在 Doc 里：

- 修改 AI 给出的实现方式描述；
- 明确否决某些方案。

\* (2) Tasks 快速发现Agent任务拆分不对，进行相应的纠正 在进入执行前，Spec 会把 AI 计划做的事情拆成 Tasks：\*

- 要改哪些路径
- 要动哪些文件
- 每一步预计会修改什么



以：

- 在 **Tasks** 阶段就发现路径写错了（比如某个 *api* 路径不对）；
- 直接在任务层面改正，而不是等执行完再面对一堆错误 *diff*；
- 发现不对可以不让它继续到 *Changes* 阶段，\*\*相当于在“将要执行”阶段就踩了刹车\*\*。

### 预览和元素级调试

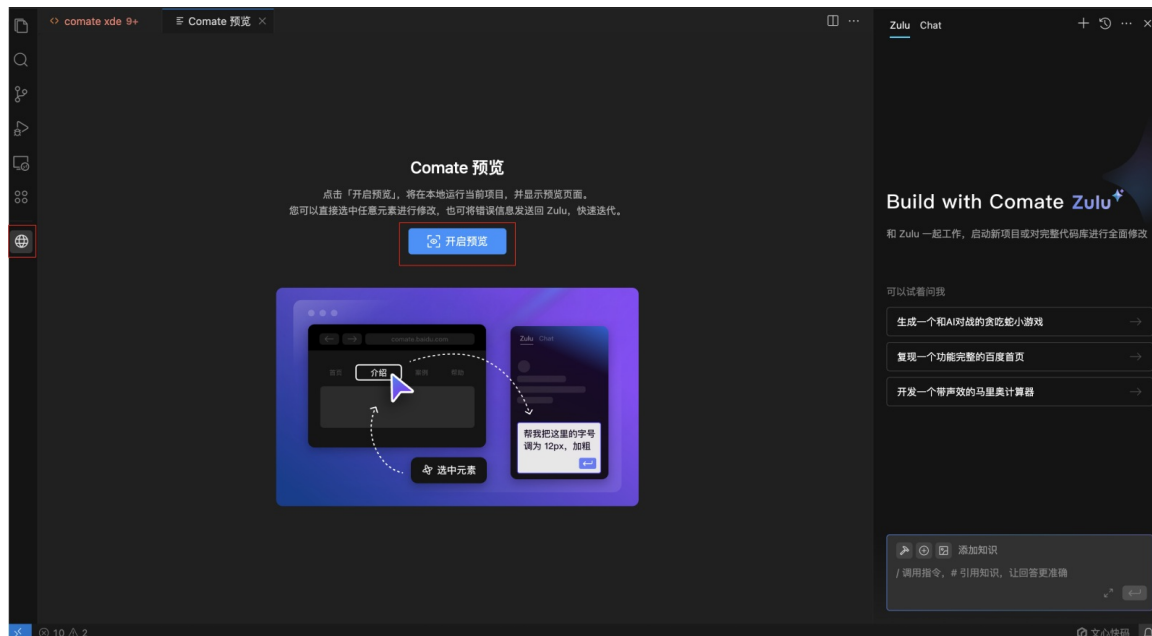
#### 🔗 功能描述

1. Comate AI IDE 可自动识别前端项目，快速启动并在 IDE 内查看运行效果，无需跳转至浏览器，实现沉浸式开发。
2. 精准操作，只需鼠标单击，即可选中待修改的组件和元素，告别繁琐的自然语言描述。
3. Comate AI IDE 内置浏览器具备网页运行时错误识别功能，一键发送到 Zulu，快速解决问题。

#### 🔗 IDE 内预览 Web 页面

Comate AI IDE 支持识别前端项目、快速启动前端项目并在 IDE 内查看运行效果，为您带来无需跳转至浏览器的无缝开发体验。您可通过以下入口在 IDE 内预览Web页面：

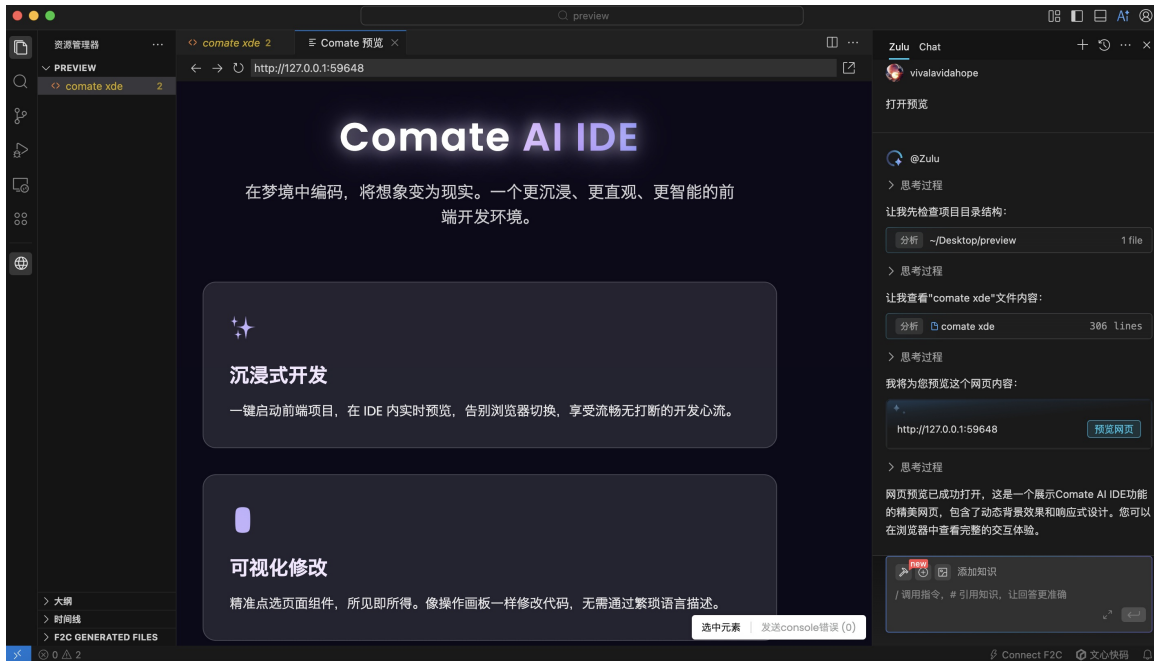
1. 打开文件后，点击预览图标即可唤起预览页面。在预览页面点击“开启预览”按钮，Zulu 会自动检查是否存在可预览实例，若有将在预览页面中即时呈现 Web 页面。



2. Zulu 分析过程中显示 预览网页 按钮，点击该按钮也可自动打开预览页面，即时查看 Web 页面。



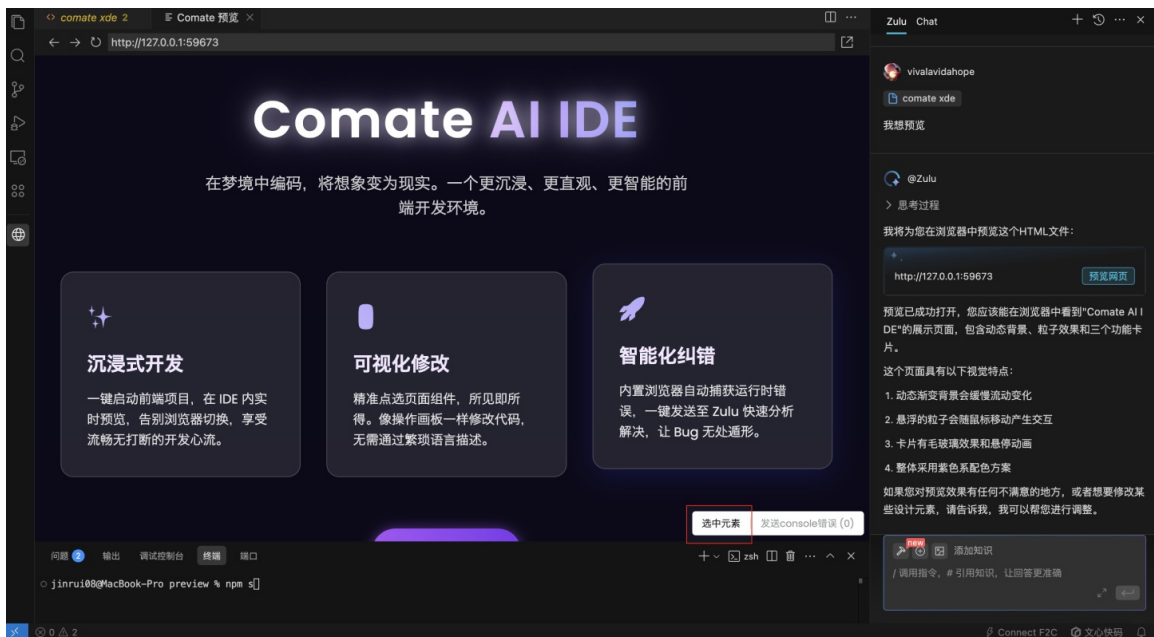
### 3. 查看预览效果



## 🔗 元素级调试

Comate AI IDE 提供精准元素级调试体验，您仅需通过鼠标单击，便能快速选中待修改的组件与元素，有效避免使用繁琐的自然语言描述来定位目标。调试方法如下：

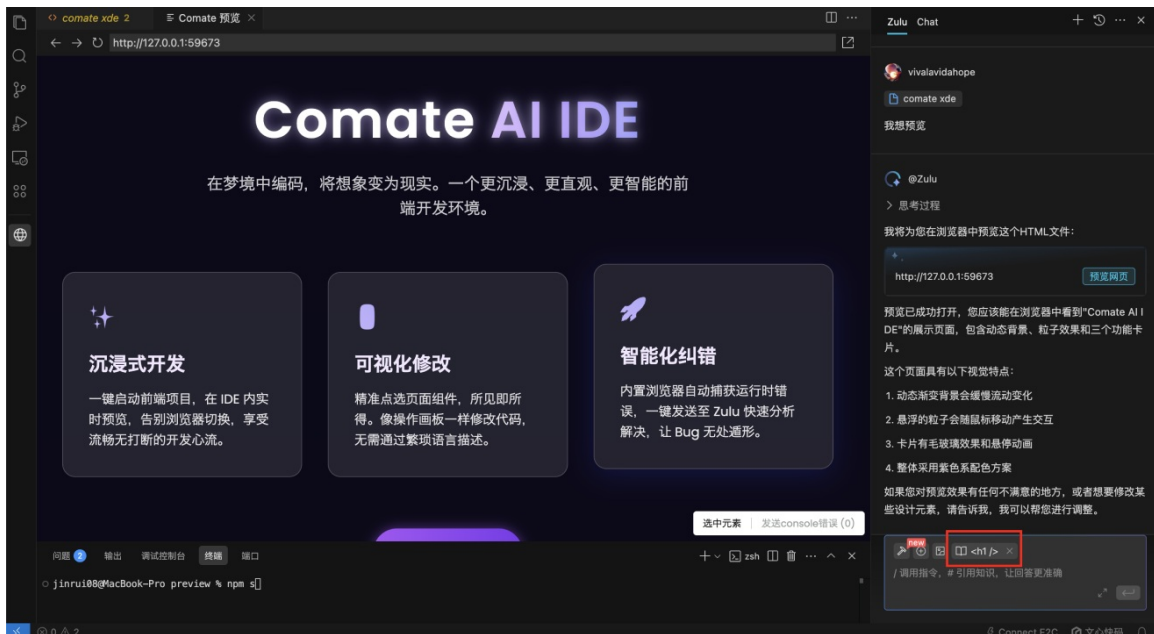
1. 预览页面中显示 web 页面后，可点击 **选中元素** 按钮；



2. 选择期望修改的单元元素；



- 选择完成后，对话框中将添加待修改元素标签，输入期望修改的指令，例如：“帮我把 IDE 改为 XDE”，点击 运行 按钮，Zulu 将对元素开始修改，并实时预览修改后效果。



## 手动安装插件

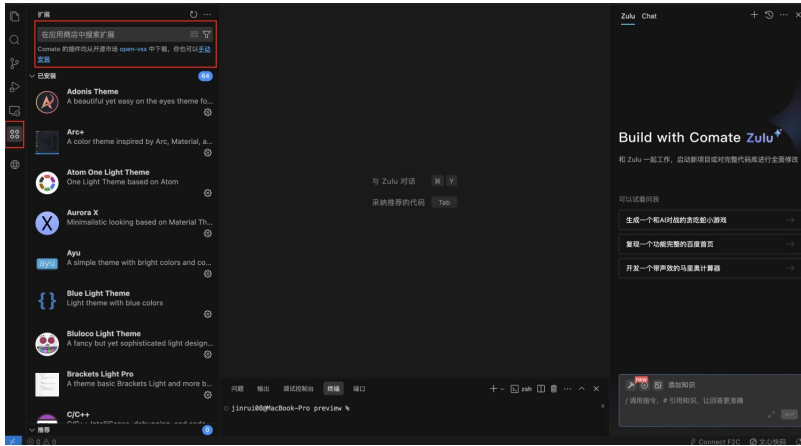
### 安装插件

您可通过两种方式为 Comate AI IDE 安装插件：

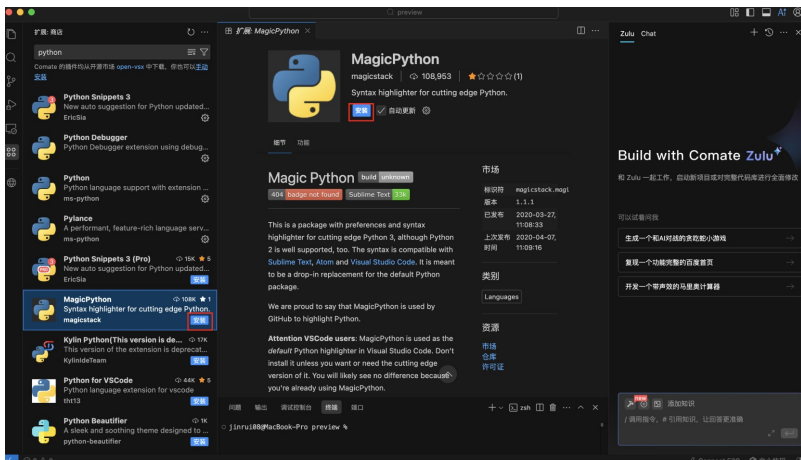
- 插件应用商店安装：直接在 Comate AI IDE 的插件应用商店下载安装所需插件。
- 导入 .vsix 文件：从 VS Code 插件市场下载或使用您本地已有的插件 .vsix 文件，导入 Comate AI IDE 即可安装。

### 从 Comate AI IDE 的插件应用商店安装

- 打开插件应用商店：在左侧导航栏中，点击 扩展 图标，打开 插件应用商店 面板。

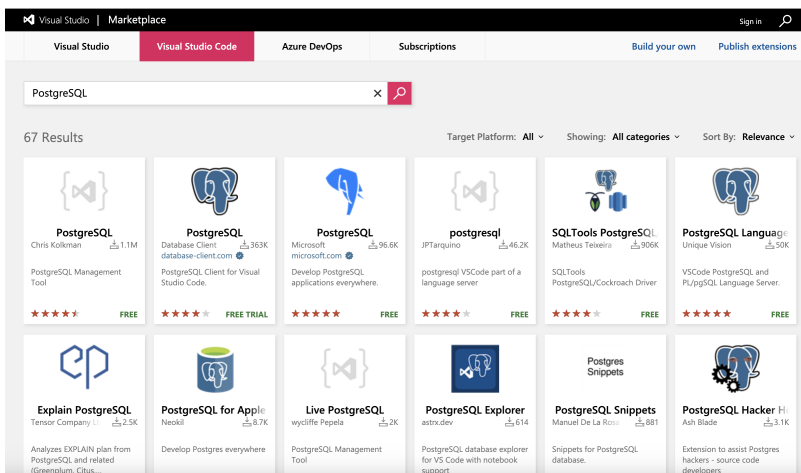


2. 查找并选择插件：在插件商店中搜索您想安装的插件，点击该插件，您将看到包含插件的详细描述、更新日志等关键信息。
3. 进行安装：点击插件右下角或详情窗口中的 **安装** 按钮，Comate AI IDE 将自动安装，安装完成后可在 **已安装** 列表中找到您刚刚添加的插件。



从 VS Code 的插件市场安装

1. 请前往 **VS Code插件市场** 进行搜索。
2. 输入您想要安装的插件名称，例如 **PostgreSQL**。



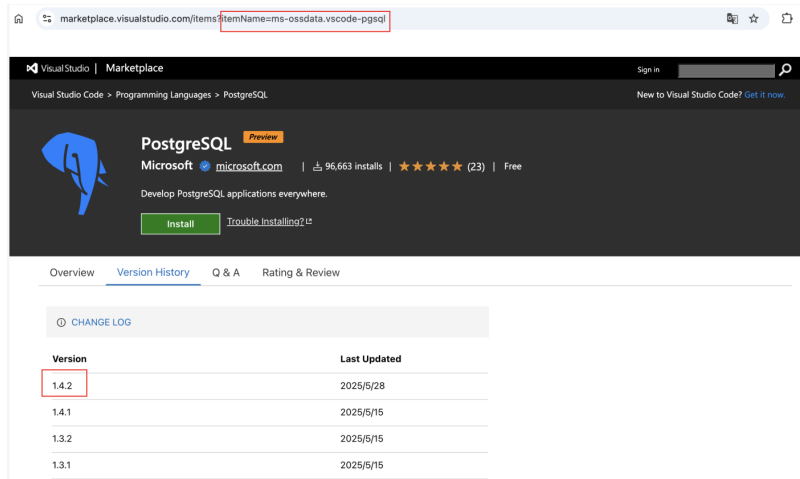
3. 在搜索结果列表中，点击您感兴趣的插件以查看其详情。
4. 在插件详情页面，找到并点击 **Version History** 选项。
5. 结合插件页的 URL 和 **Version History** 中的信息，提取出以下信息（以 PostgreSQL 为例）：

itemName: URL Query 中的 itemName 字段，如截图中的 `ms-ossdata.vscode-pgsql`，并将小数点 (.) 前后的内容分成以下字段：

fieldA: `ms-ossdata`

fieldB: vscode-pgsql

version: 例如 1.4.2



6. 使用提取出来的 3 个字段的值替换下方 URL 中的同名字段。

模板：

`https://marketplace.visualstudio.com/_apis/public/gallery/publishers/${itemName.fieldA}/vsextensions/${itemName.fieldE}`

示例：`https://marketplace.visualstudio.com/_apis/public/gallery/publishers/ms-ossdata/vsextensions/vscode-pgsql/1.4.2/vspackage`



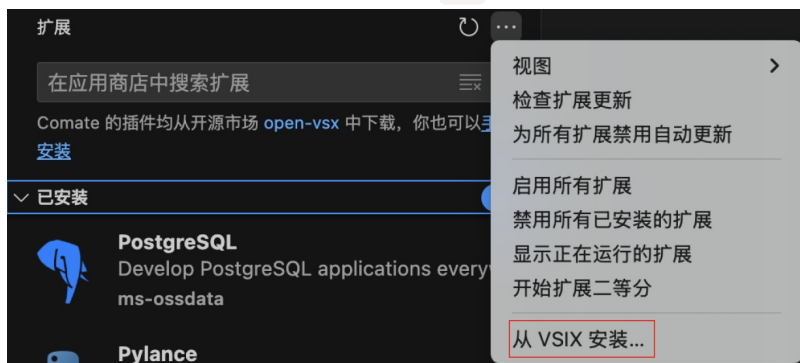
7. 在浏览器地址栏输入修改后的 URL，并按下回车键。

8. 浏览器将自动开始下载插件的 .vsix 文件。

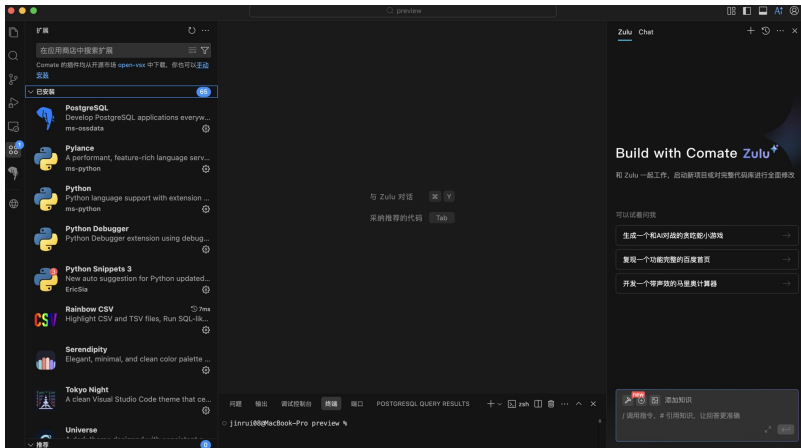
9. 下载完成后，返回 Comate AI IDE，然后打开 插件应用商店 面板。

10. 安装方式一（拖拽）：将下载好的 .vsix 文件直接拖拽至插件应用商店区域。

11. 安装方式二（选择文件）：在插件应用商店面板的右上角，点击…(更多选项)图标。在下拉菜单中选择从 VSIX 安装。\* 在弹出的文件选择窗口中，找到并选中您的 .vsix 文件导入。



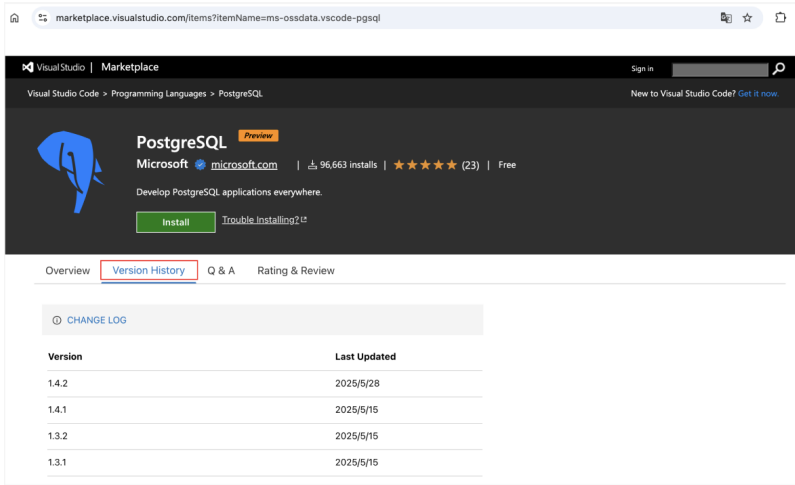
12. Comate AI IDE 将自动执行插件安装。安装成功后，该插件将显示在 已安装 列表中。



注意：C/C++ 依赖特定的平台架构的插件，无法直接通过插件市场安装，您需要手动下载并安装对应您平台的 .vsix 文件。

### 常见问题

暂不支持安装 VS Code 插件市场中某个版本的插件 当 VS Code 插件市场里特定版本的插件依赖新版 VS Code 的某些接口时，可能会出现该插件与 Comate AI IDE 不兼容的情况，您可前往该插件的 Version History（版本历史记录）页面，从中下载该插件的历史版本来解决兼容性问题。



## 开放平台

### Baidu Comate开放平台是什么？

通过Baidu Comate，对第三方开发者工具、在线服务等进行知识扩展与能力扩展，使得第三方服务可通过Comate直接提供给用户使用，将研发各个环节连接到编程现场，共同打造以编码环节为中心的生态系统，打造Comate everything。

### 如何使用

#### 一、插件配置中心

首先需要在网页端中配置：[配置中心](#)

目前支持的功能：

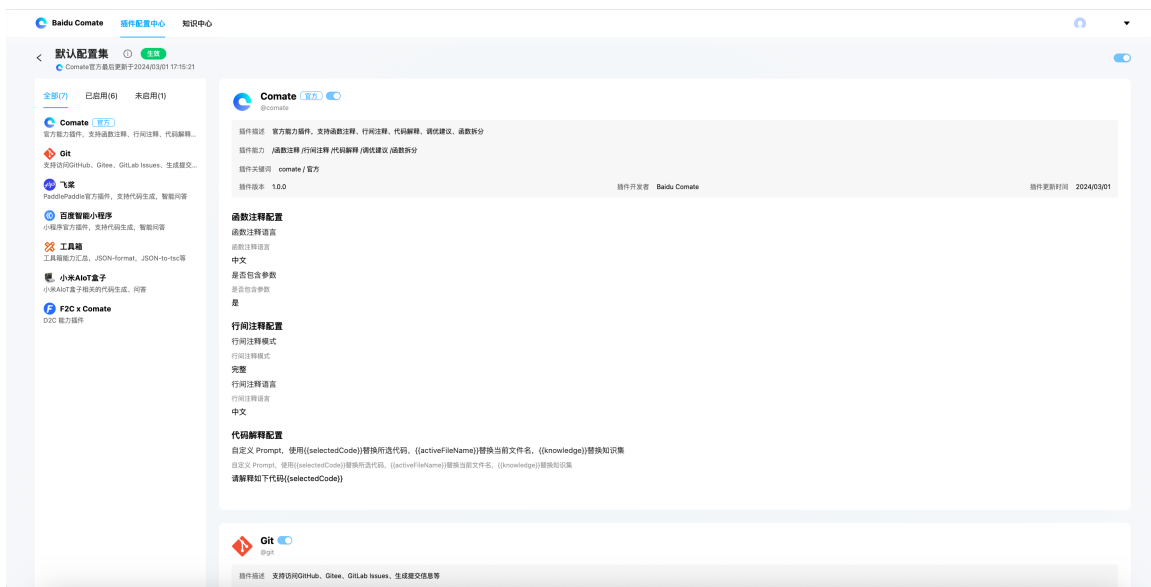
- 创建自有配置集
- 将配置集共享给团队其他成员
- 配置插件能力：本期支持Baidu Comate官方能力插件、Github 插件

#### 1. 新建配置集

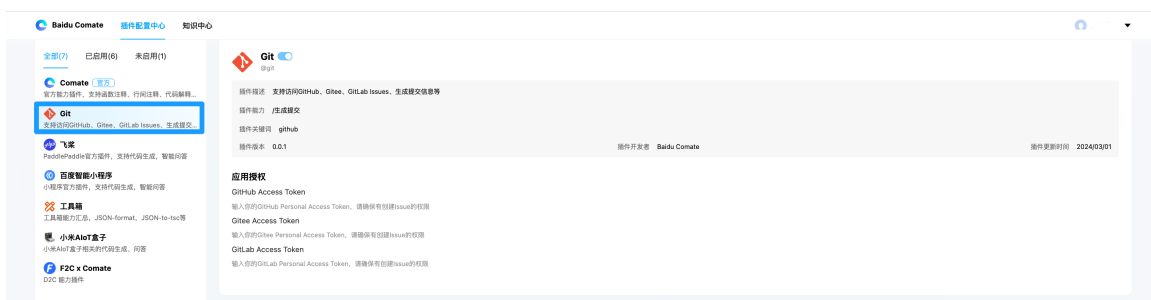


## 2. 编辑配置集，直接编辑即保存

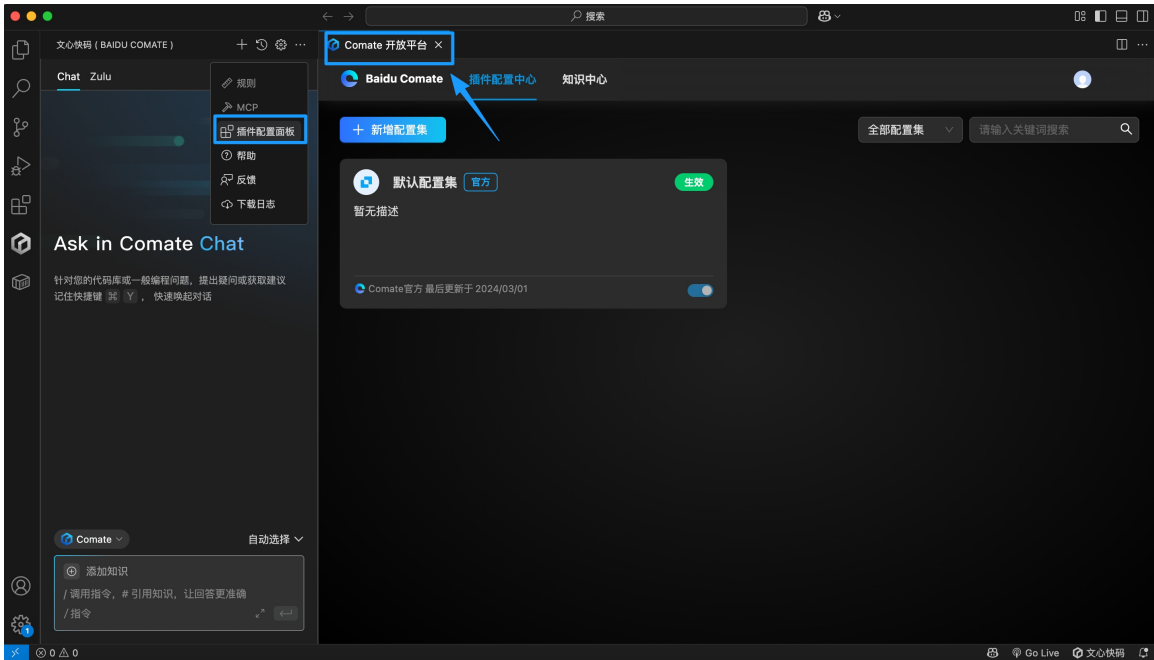
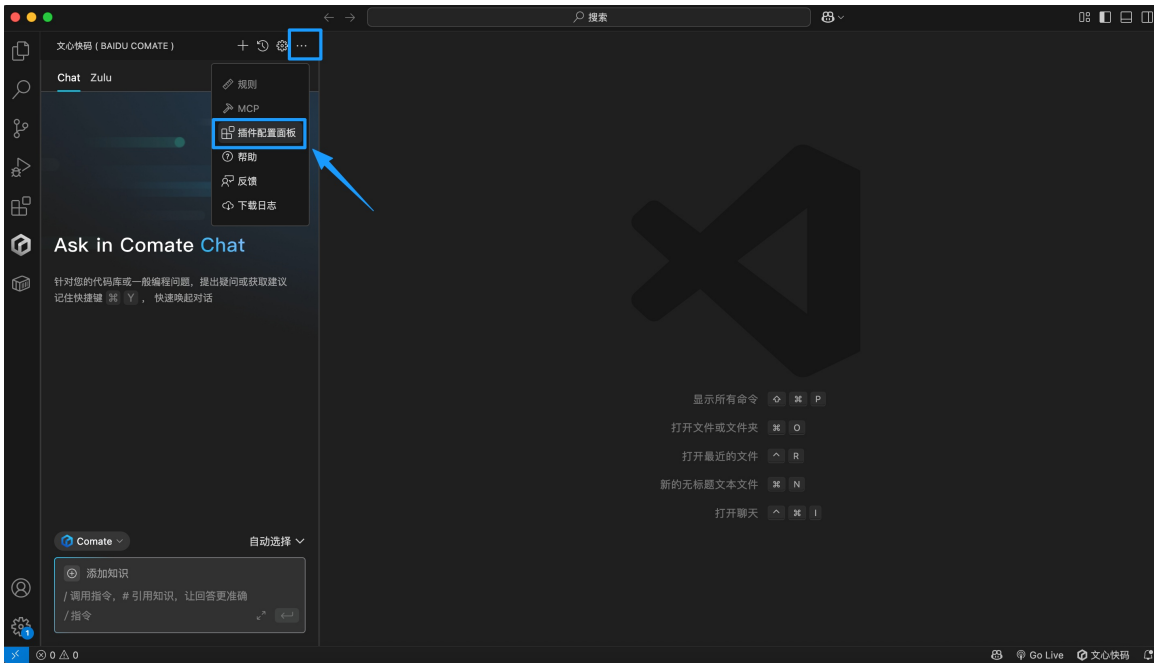
- Baidu Comate官方插件：支持为 /代码解释 能力自定义 Prompt



- Github 插件：支持访问GitHub Issues、生成提交信息等



## 3. 在 IDE 端配置 配置方式同 Web 端

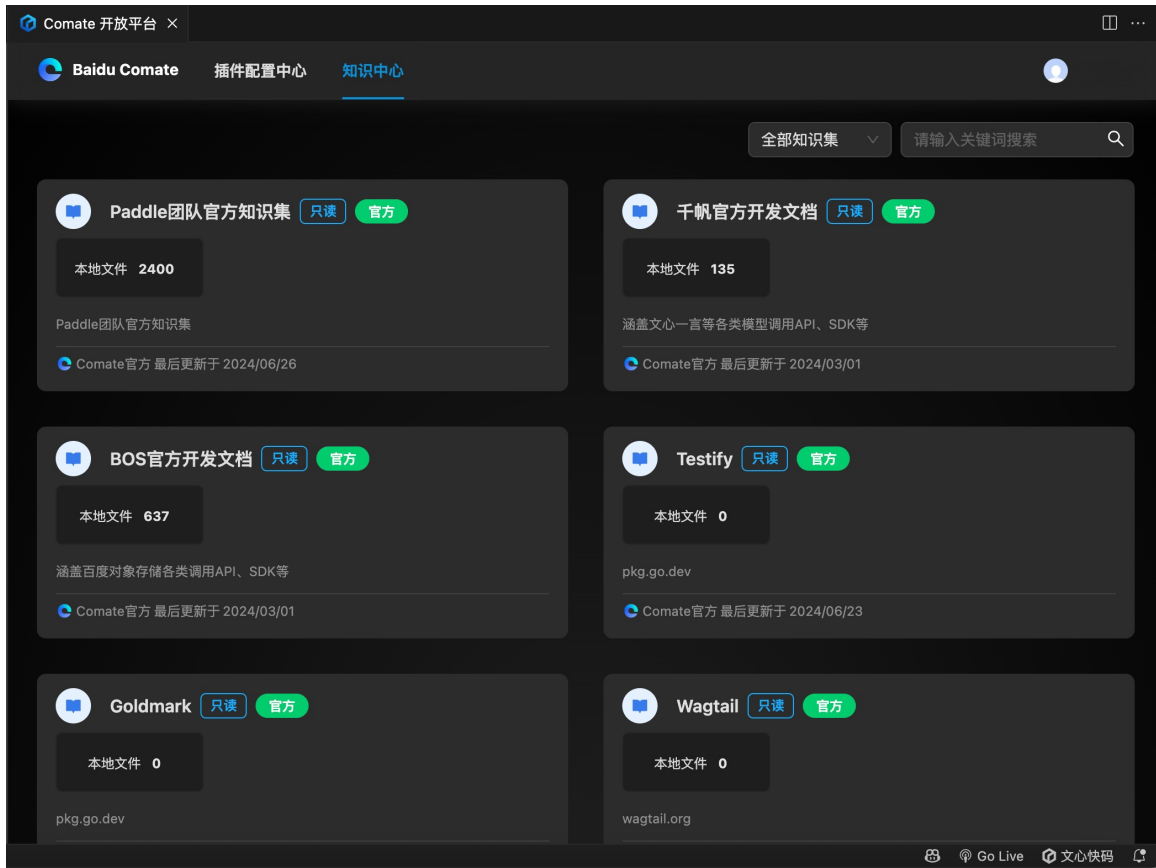


## 二、知识中心

目前支持的功能：

- 创建自有「知识集」，知识集中可以添加本地文件作为【知识】

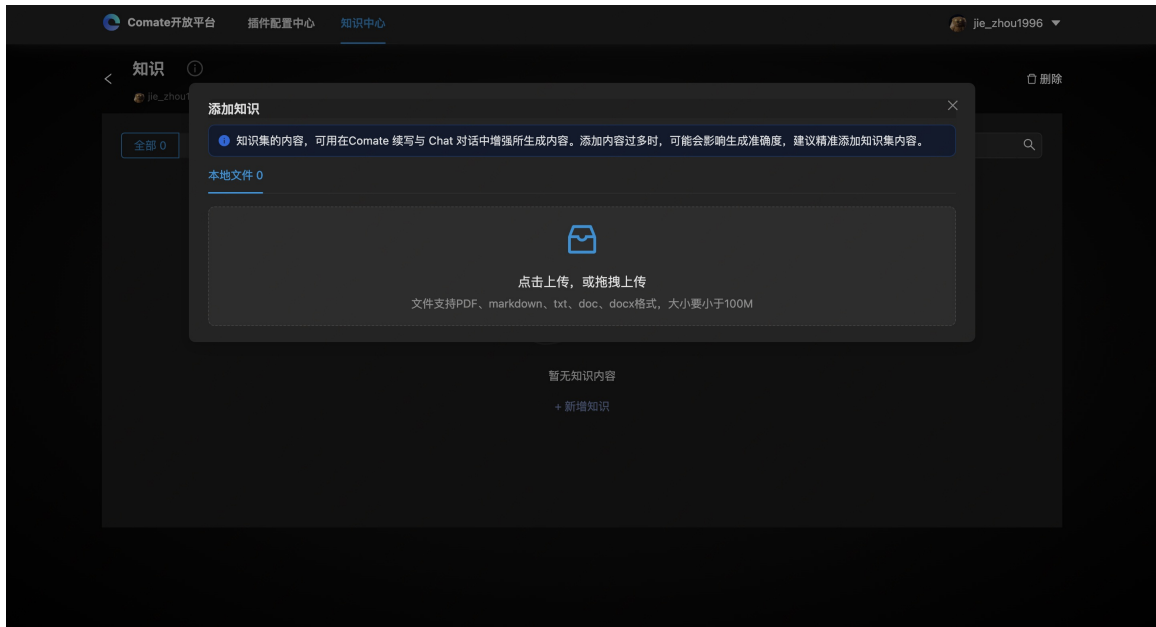
### 1. 查看现有知识集列表



## 2. 新建知识集



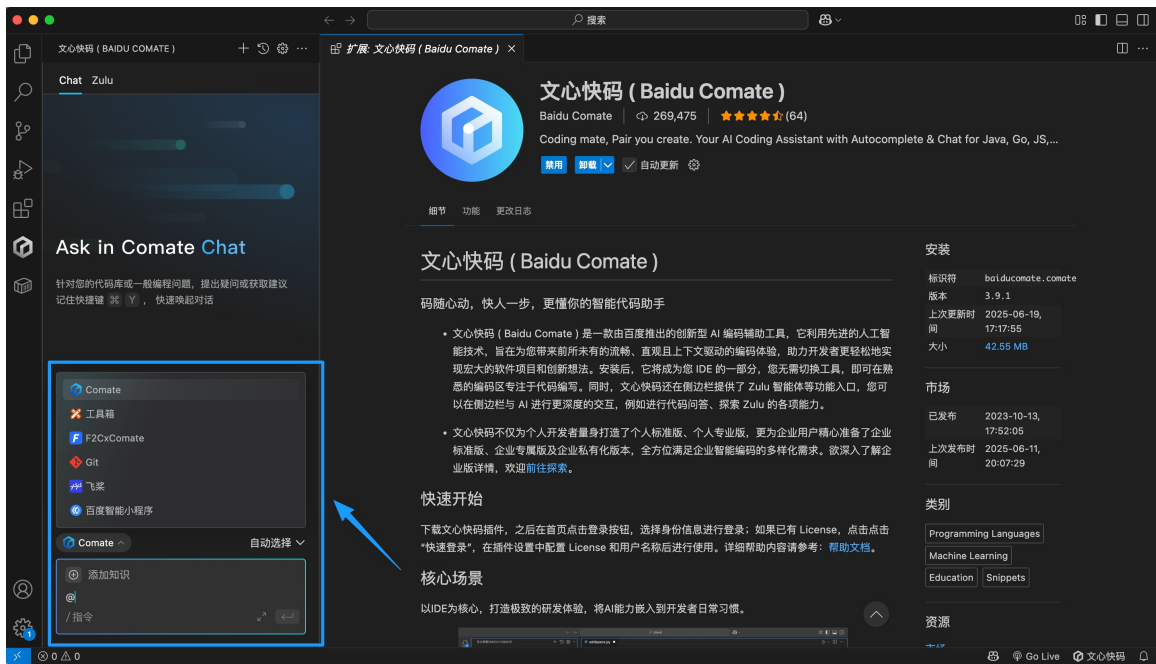
## 3. 在知识集中添加 本地文件



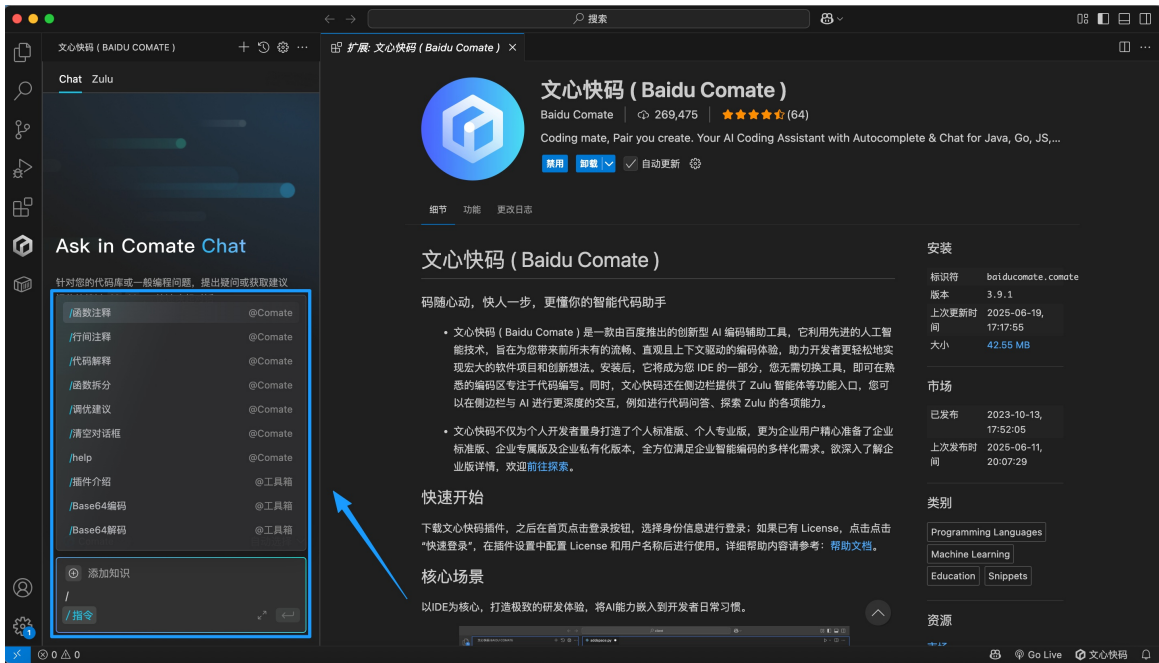
### 三、插件端使用

#### 1. 使用插件

- 通过@调起所需插件



- 通过 / 调起所需能力



2. 使用知识增强插件能力 此时分两种场景：场景一：插件开发过程中已经预置了一些知识，用户不感知知识，直接使用插件即可 场景二：用户在发起自然语言对话时，自动引入知识同时调起插件，将知识授权给插件增强生成能力 此处演示场景二：用户可感知「知识」

- 通过#调起所需知识



# 企业控制台

## 登录

企业控制台是一站式管理与洞察AI开发效能的核心平台。通过清晰的数据概览，可以实时掌握团队使用动态，包括活跃用户数、License授权状态及代码生成占比等关键指标。

控制台提供多维统计视图，支持多维度追踪使用趋势，并集成智能体资源管理。帮助管理员高效配置资源、监控使用情况，确保团队在各类主流IDE中安全、顺畅地运用AI能力，实现研发效能的持续提升与优化。

企业控制台登录方法 邮箱/工号+密码登录 输入邮箱/工号信息和密码进行登录。



## 主页

企业控制台主页以高度集约的布局呈现核心数据与全局状态。（下图为企业管理员权限登录首页展示效果）

### 1. 核心数据仪表盘

位于视觉焦点，通过卡片、图表等形式动态展示实时关键指标，如用户活跃度、资源使用率、服务健康度等，让管理者一眼掌握整体业务脉搏。

### 2. 资源与配额总览

清晰列出企业拥有的核心资源（如License许可证、智能体代金券等）的总量、已使用量和剩余量，是进行成本控制和资源规划的首要参考。

### 3. 快速操作与导航入口

提供高频管理功能的快捷入口，如成员管理、服务开通、配置修改等，并集成了文档中心、插件下载等支持资源，实现效率最



大化。

#### 一、数据概览

#### 二、个人信息

- 人员角色信息
- 企业名称

- 已授权license信息

## 团队数据概览

- 已授权 License 数（实时授权的License数量/全部有效License）
- 使用用户数（最近一周内使用过的用户数/全部加入过的用户数）
- 代码生成占比（采纳的代码行数/代码变更行数）

## 个人数据

- 今日采纳次数（当前用户今日内采纳的次数）
- 今日代码生成占比（采纳行/变更行数）
- 累计采纳次数（当前用户累计采纳的次数）
- 累计使用天数（当前用户累计使用的天数）
- 采纳次数（图表）



## 二、操作

您可以在页面右上角获取用户使用文档及插件下载信息。

## 文档中心

点击右上方文档中心进入产品介绍页面，集中展示插件和管理后台相关的说明手册，包含快速入门（插件安装手册）、插件使用手册、管理后台使用手册。



### 插件下载

点击右上角插件下载进入Comate插件下载页面，支持进行各类插件的下载。集中展示 Comate 插件版本和对应支持的 IDE 版本号，用户可以从此处下载所需的插件包到本地。

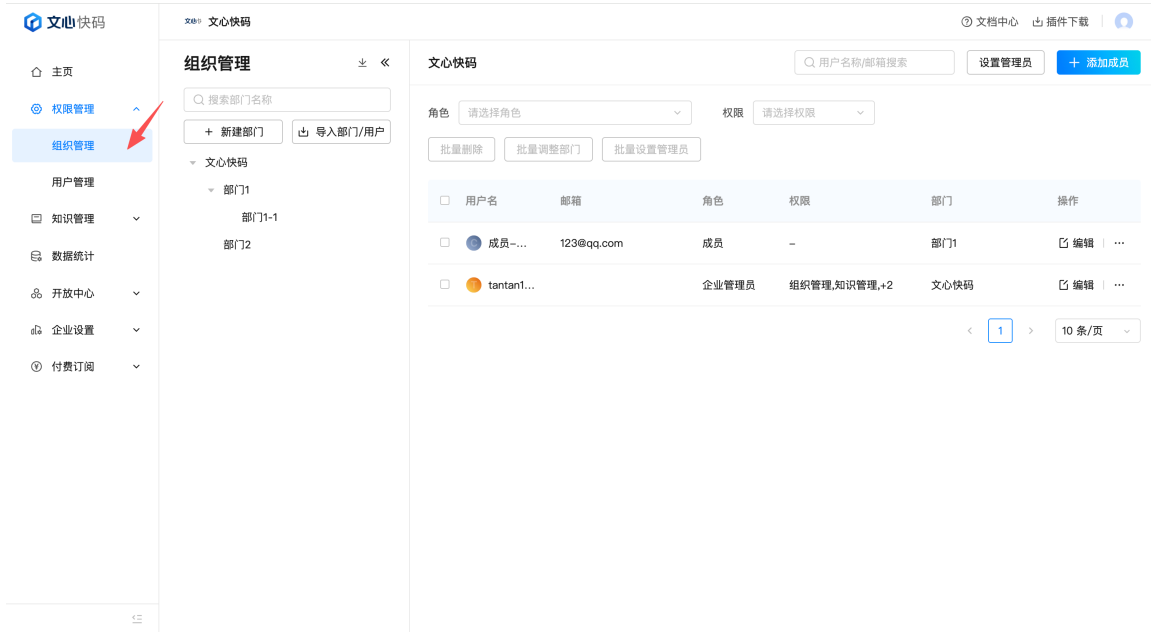


## 权限管理

### 组织管理

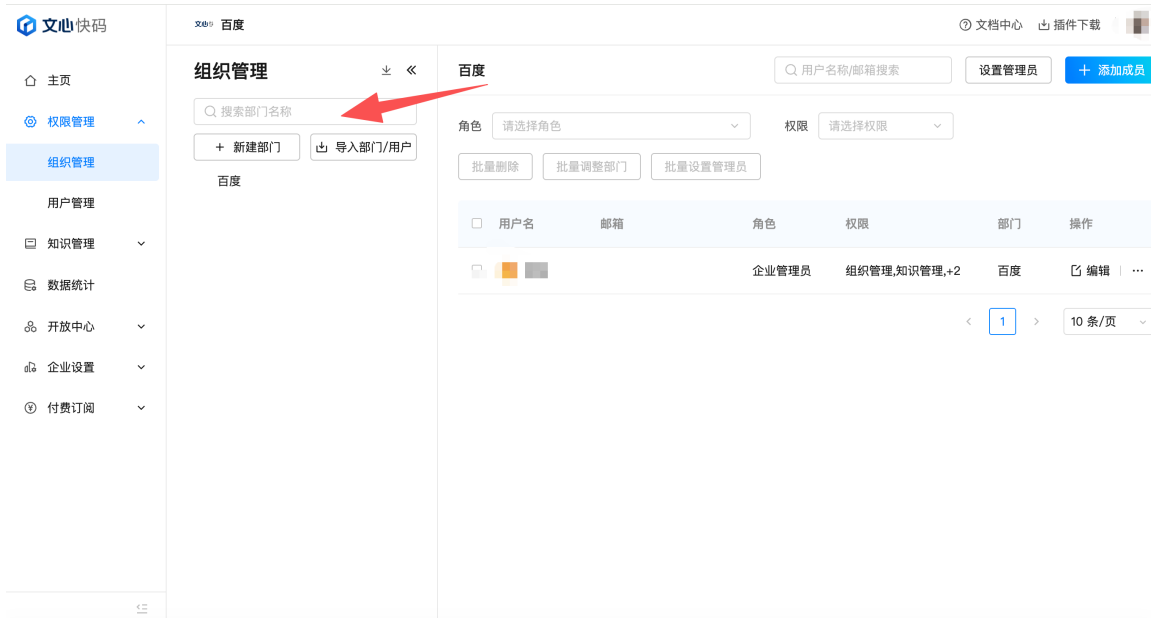
组织管理页面用于展示和管理企业内的组织架构信息和全部企业用户的基本信息。右侧表单展示的是左侧已选部门内公司用户的所有信息，管理员可以进行修改用户的角色、权限、部门、重置密码等操作。

企业管理员、部门管理员有权限查看用户管理页面，普通成员无权限查看该页面。

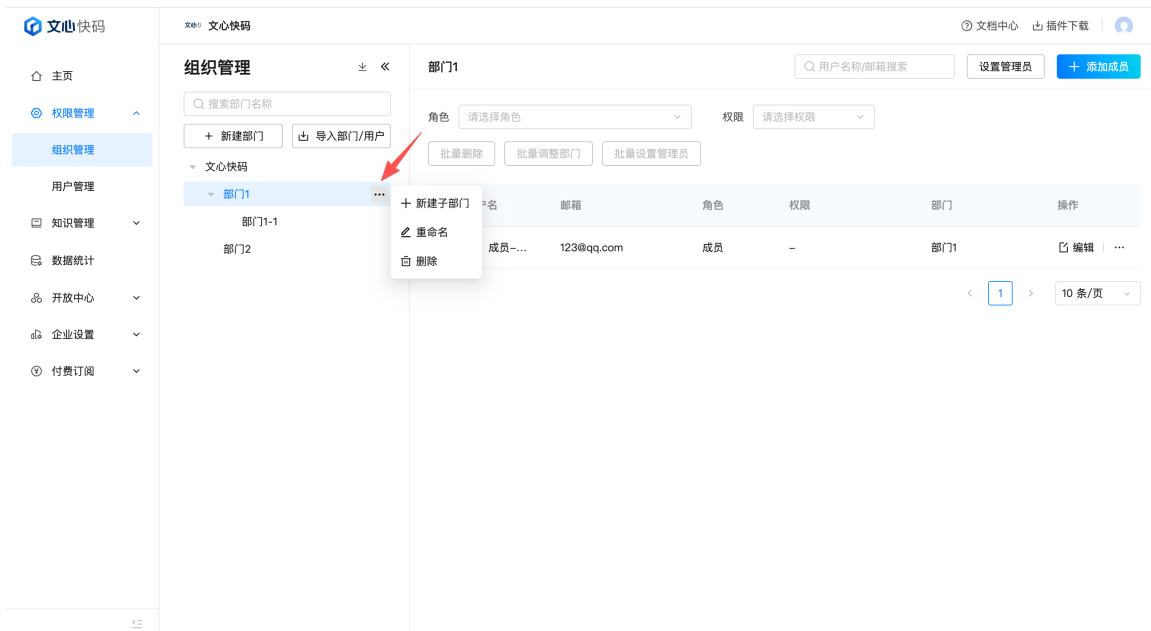


一、编辑组织架构 展示企业组织架构与用户。用户与组织数据同步采用接口对接，用户不可修改原有组织架构信息。

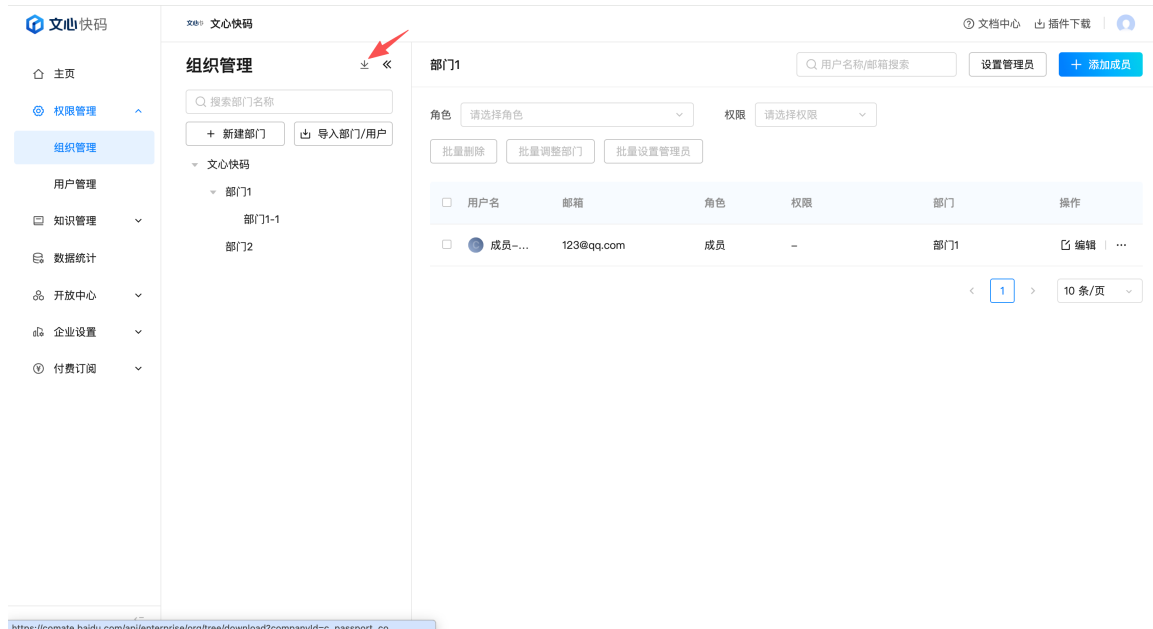
1.点击每个层级的添加子部门按钮，输入部门名称，可自动在该层级内添加一个下级部门。



重命名可修改当前部门名。

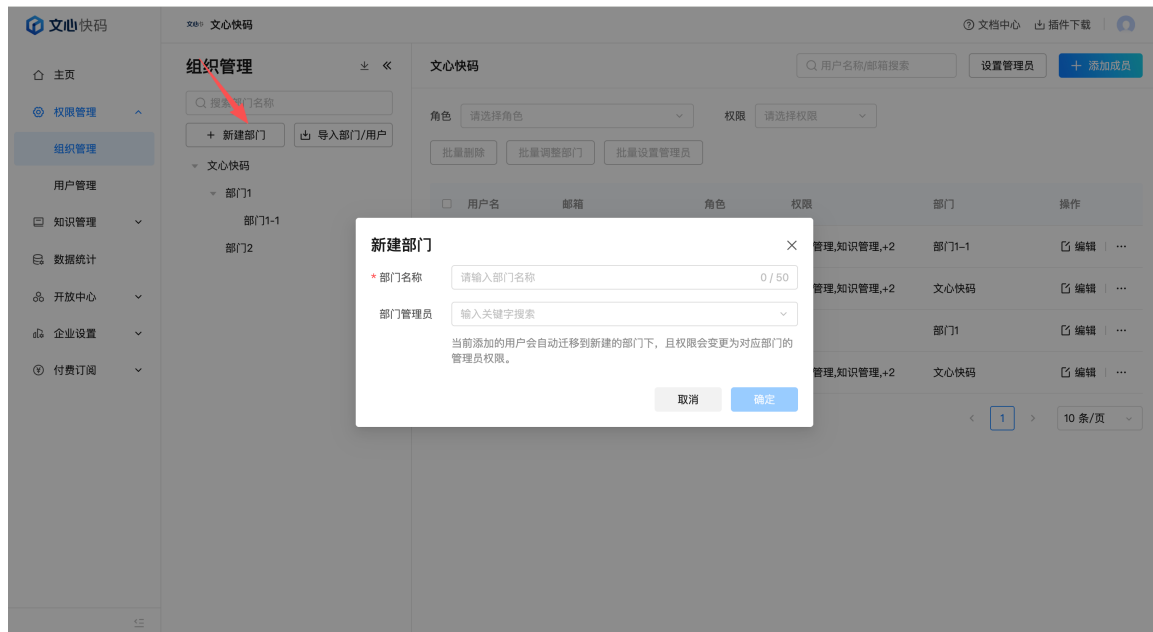


下载可将组织架构下载为excel文件。

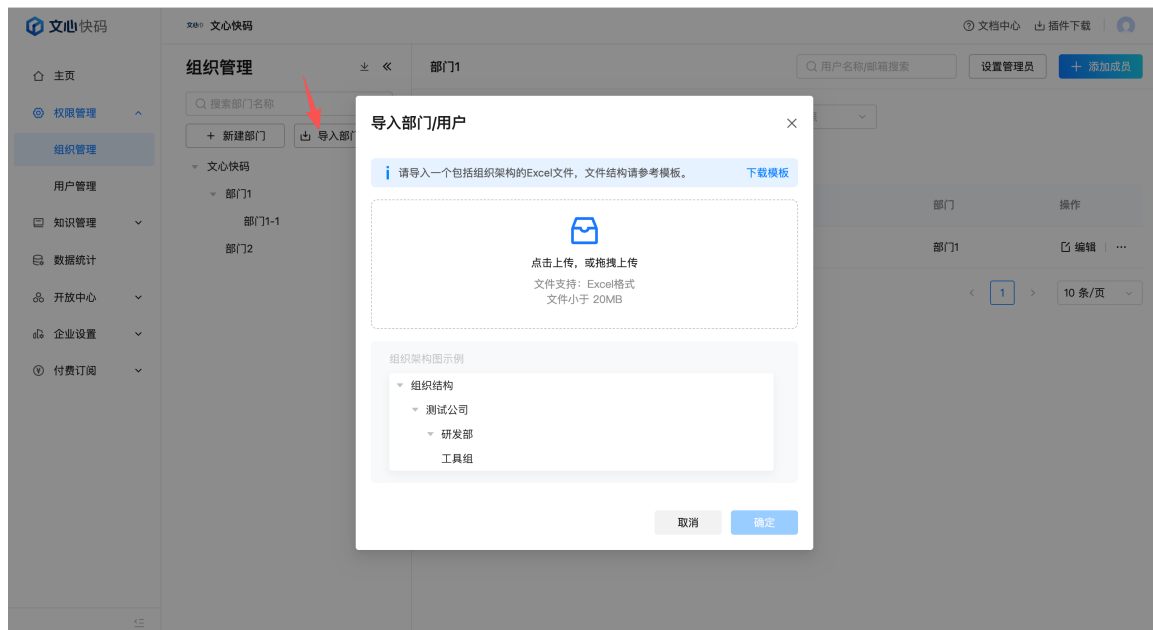


3. 点击

新建部门可以新建一个部门并添加管理员，添加的用户会自动迁移到新建的部门下，且权限会变更为对应部门的管理员权限。



4. 以从本地Excel文件导入组织架构，文件结构可以下载模板文件进行参考。

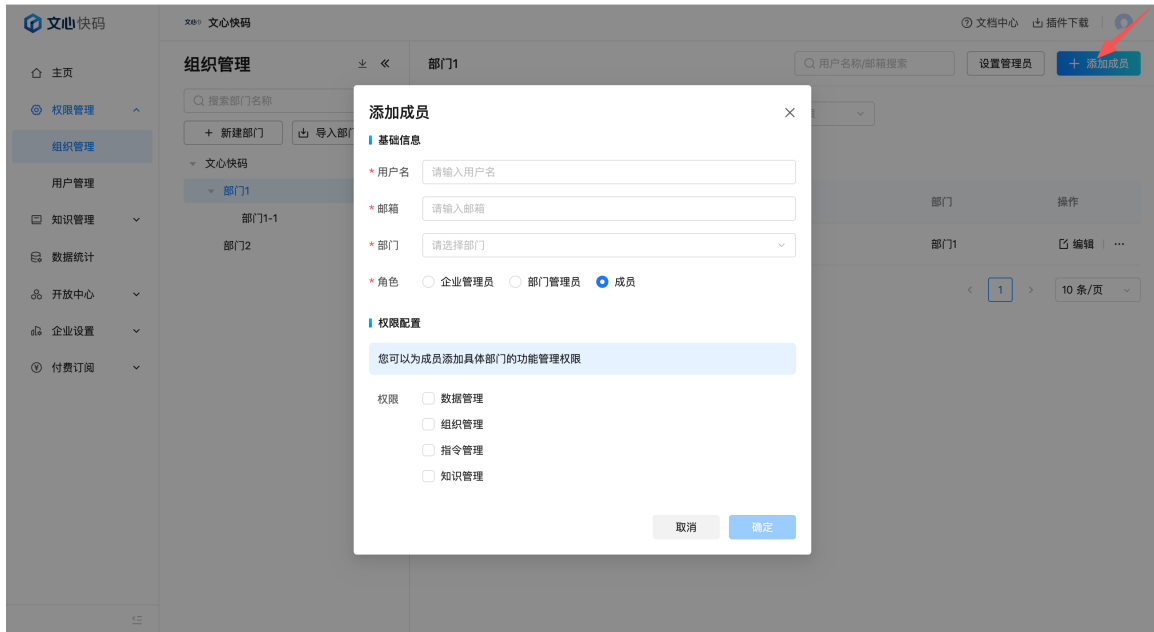


二、编

辑用户信息 从左侧选择组织架构中的某个部门后，可在右侧添加用户到该部门。管理员可以对单个用户进行权限管理、删除、调整部门、重置密码等操作。

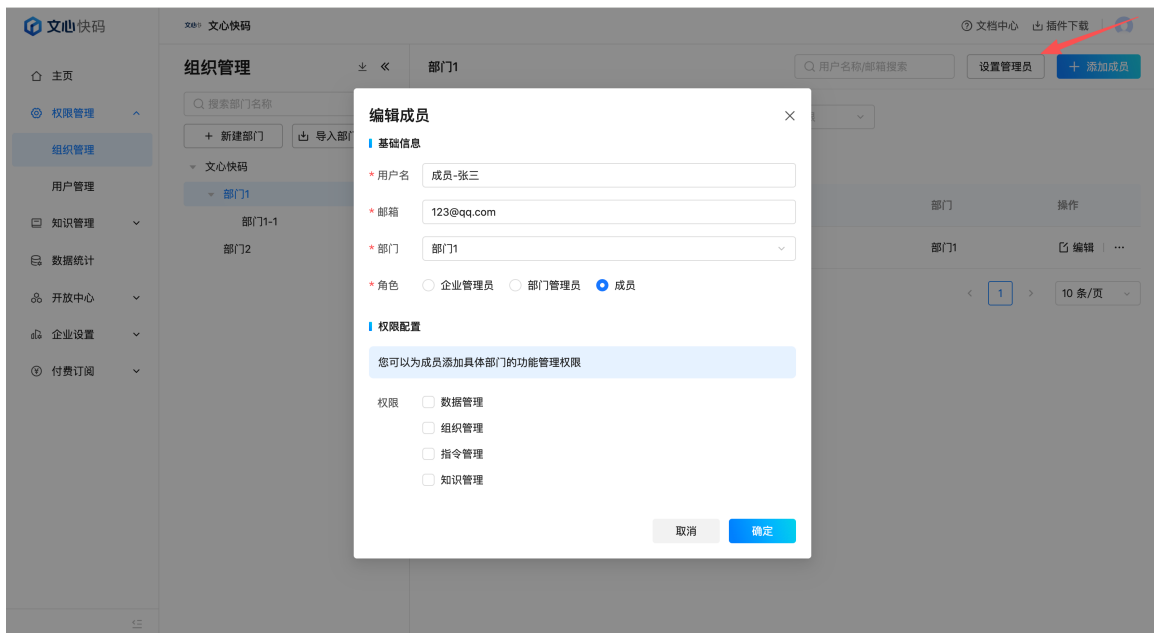
- 企业管理员可添加企业管理员、部门、部门管理员、普通用户
- 部门管理员可添加本部门管理员、部门成员（普通用户）

1. **添加成员**：点击添加成员按钮，出现添加成员的弹窗，需要手动输入用户信息并为成员添加具体部门的功能管理权限。



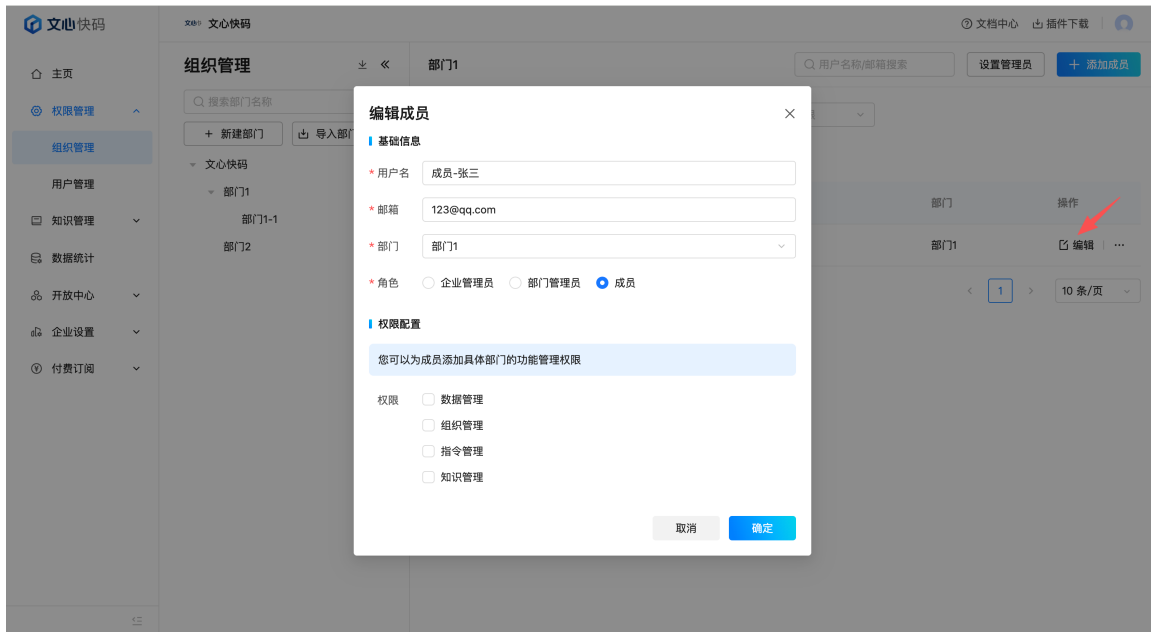
2. 设置

**管理员**：企业管理员可以设置企业管理员以及部门管理员。



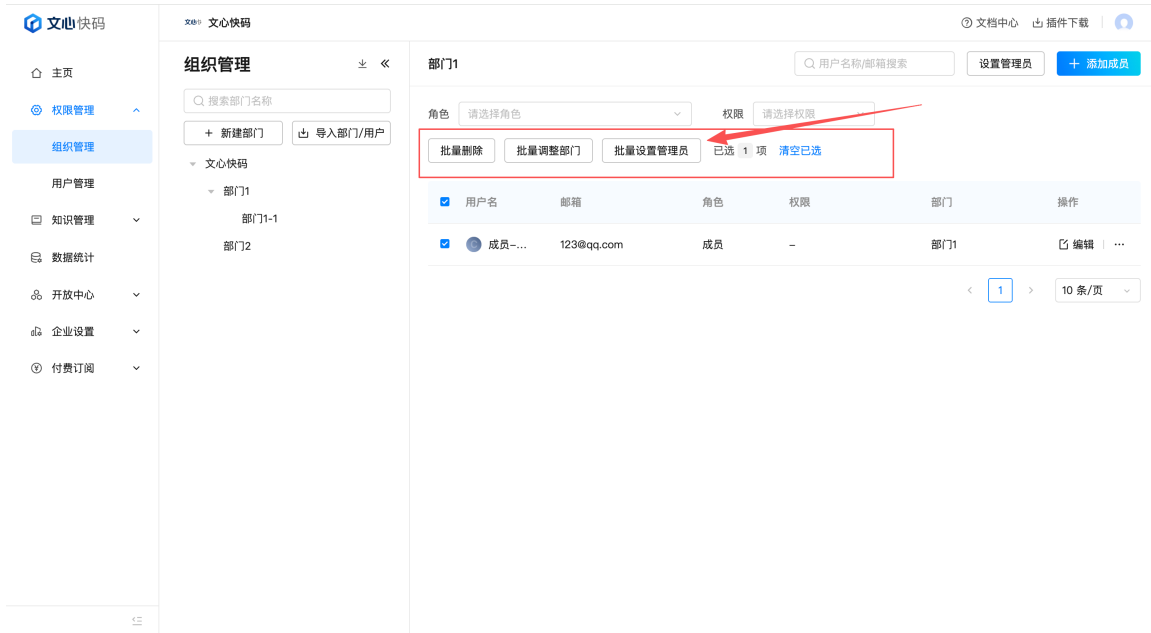
3. 编辑

**成员**：企业管理员和部门管理员可以修改成员角色、权限、部门等，点击编辑后在弹出页面中手动修改。



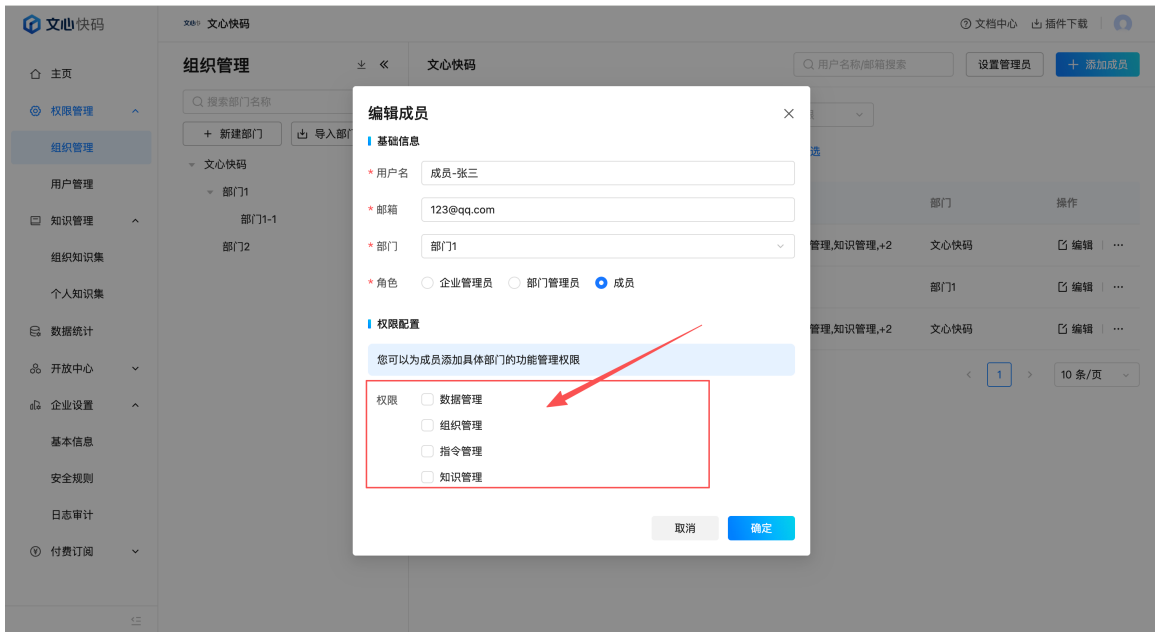
4. 批量

管理成员：选择多个成员后，可以批量删除、批量重置密码、批量调整部门、批量设置管理员等操作。



5. 权限管理：企业管理员和部门管理员可以对某个成员设置权限。具体可以针对某个部门设置对应的数据管理、组织管理、指令管理、知识管理的权限设置，支持添加多个权限。

企业管理员拥有企业的全部管理权限；部门管理员自动拥有所属部门的管理权限，可以额外增加其他部门的功能管理权限；管理员可以为成员添加具体部门的功能管理权限，成员的组织管理权限不能给别人授权任何业务权限，只能管理组织架构、人员基础信息修改（姓名、邮箱、所属组织）

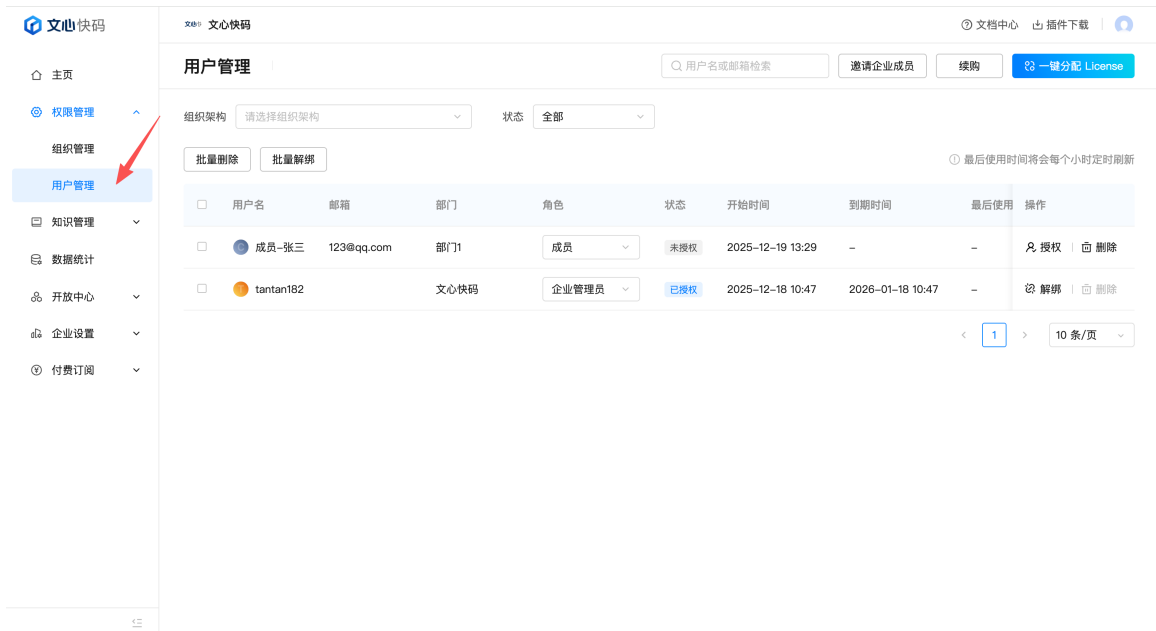


- **数据管理**：拥有数据管理权限的成员可以查看、下载成员有管理权限的部门的数据
- **组织管理**：拥有组织管理权限的成员可以新建/删除子部门、添加/删除组织成员、为组织成员修改角色、修改权限、调整部门、重置密码
- **指令管理**：拥有指令管理权限的成员可以新建、删除、查看、编辑个人或具有管理权限的组织指令集
- **知识管理**：拥有知识管理权限的成员可以新建、删除、查看、编辑个人或具有管理权限的组织知识集

### 用户管理

用户管理页面用于进行license授权和解绑，此页面展示和管理企业用户绑定License的基本情况（包括License状态、用户名、邮箱、开始时间、到期时间、最后使用时间）。表单展示的是公司所有license的授权信息，企业管理员可以对企业当前拥有的所有license进行解绑或授权操作。还可以筛选查看未授权、已授权和已过期的license。

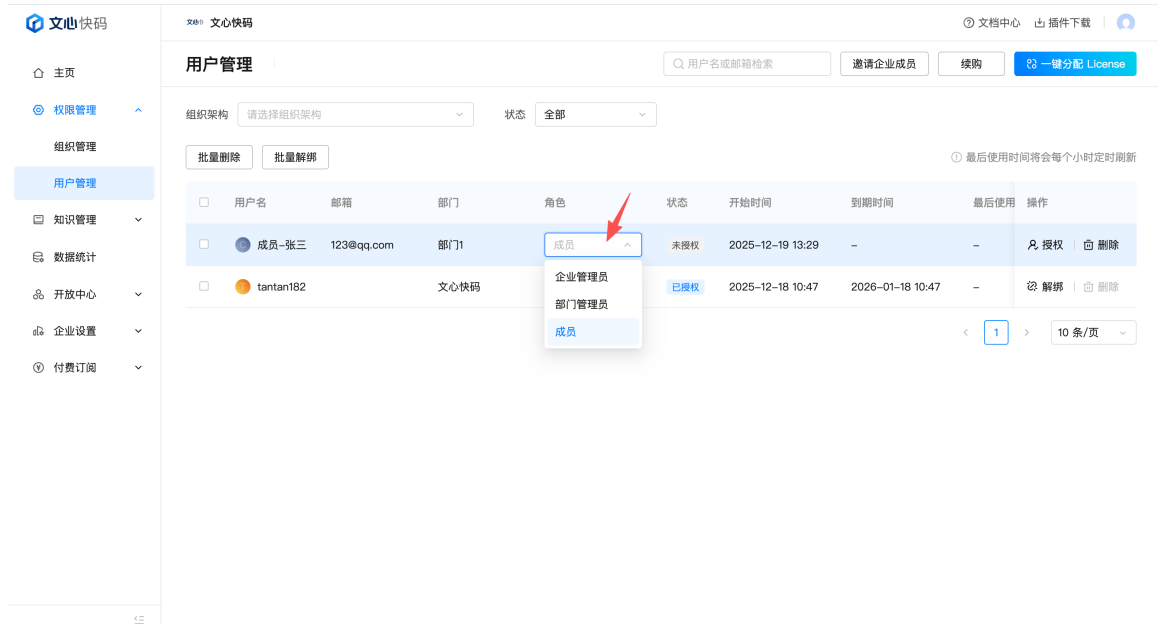
- 企业管理员：可查看全部的企业用户并进行管理。
- 部门管理员：可查看本部门的用户并进行管理。
- 针对只有权限管理的用户：只可以查看有权限部门的用户列表并进行license解绑和授权，但不可以改变成员角色。



一、更

### 改用户角色

企业管理员可以改变或将用户身份设置为企业管理员、部门管理员或成员

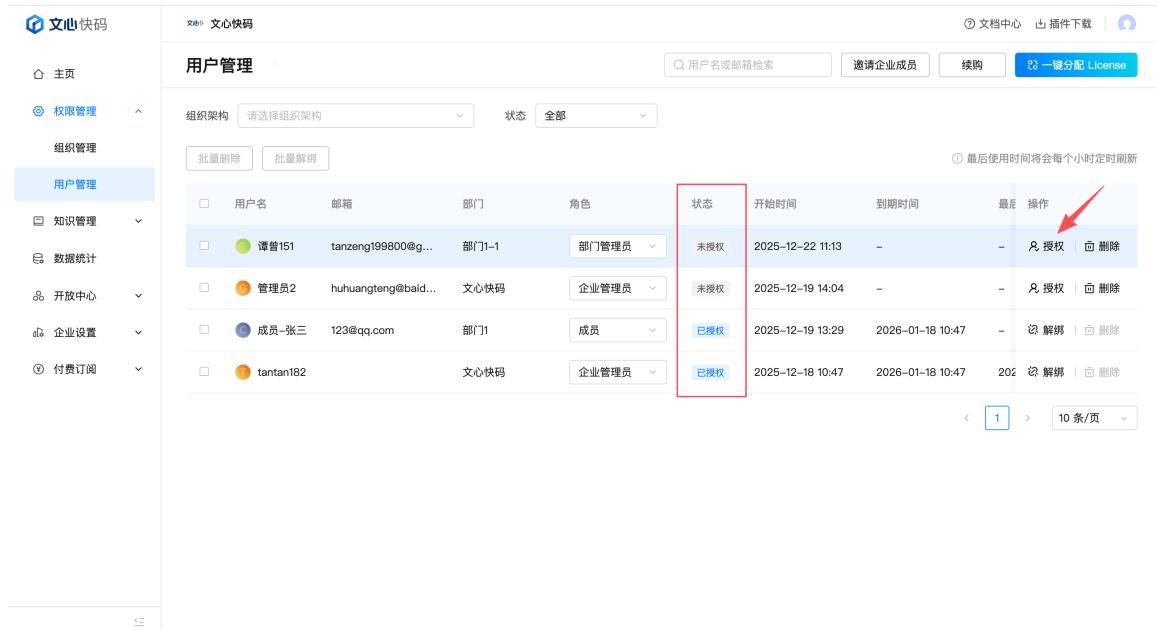


二、用

### 户授权、解绑、删除操作

在表单中的“操作”一栏，可对某一个license进行授权和解绑操作。

- 若用户未授权license：点击授权后，可以对其进行授权，点击后直接进行授权
- 若用户已授权license：点击解绑后，该license将被废弃，不可分配给其他企业成员
- 一键分配license：支持按照用户名单自动分配License，点击后自动分配license



## 知识管理

### 基本概念

#### 1. 知识集定义

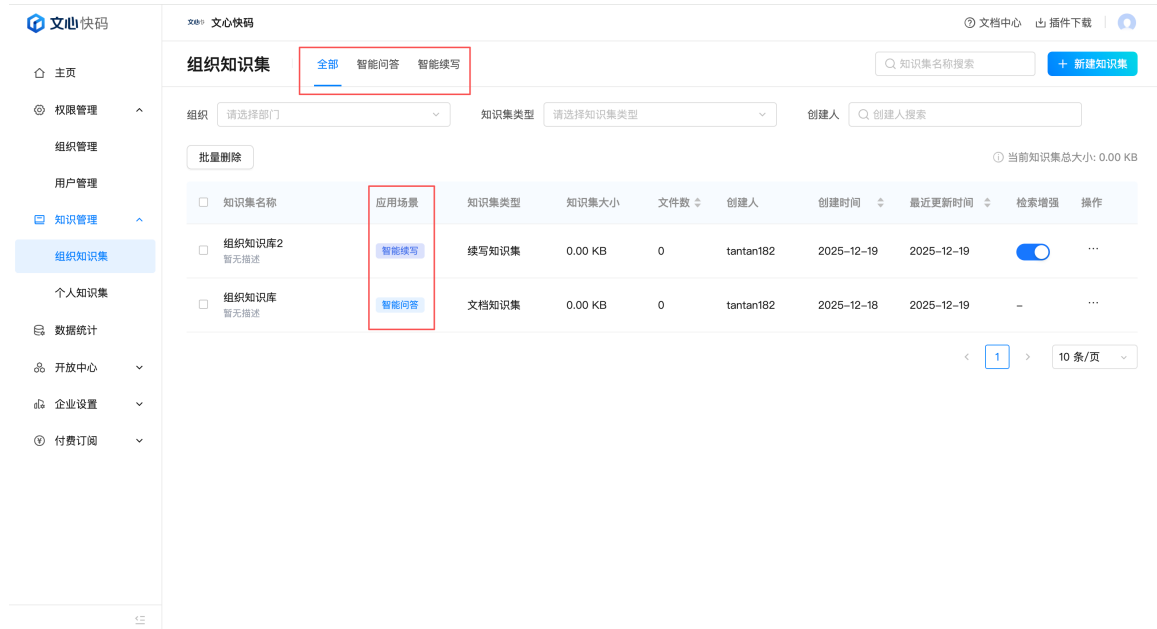
知识集：一系列具备同类别知识文件的集合，分别用于问答场景和续写场景。

- 智能问答：可以上传文档形式的知识内容，如Spring开发文档、文心一言API文档、LangChain开发文档等，也可以上传代码库、代码文件等代码形式的知识文件内容主要用于私域知识问答功能。
- 智能续写：可以上传代码库、代码文件等代码形式的知识文件内容。主要用于代码续写增强功能。

## 2. 知识集类型

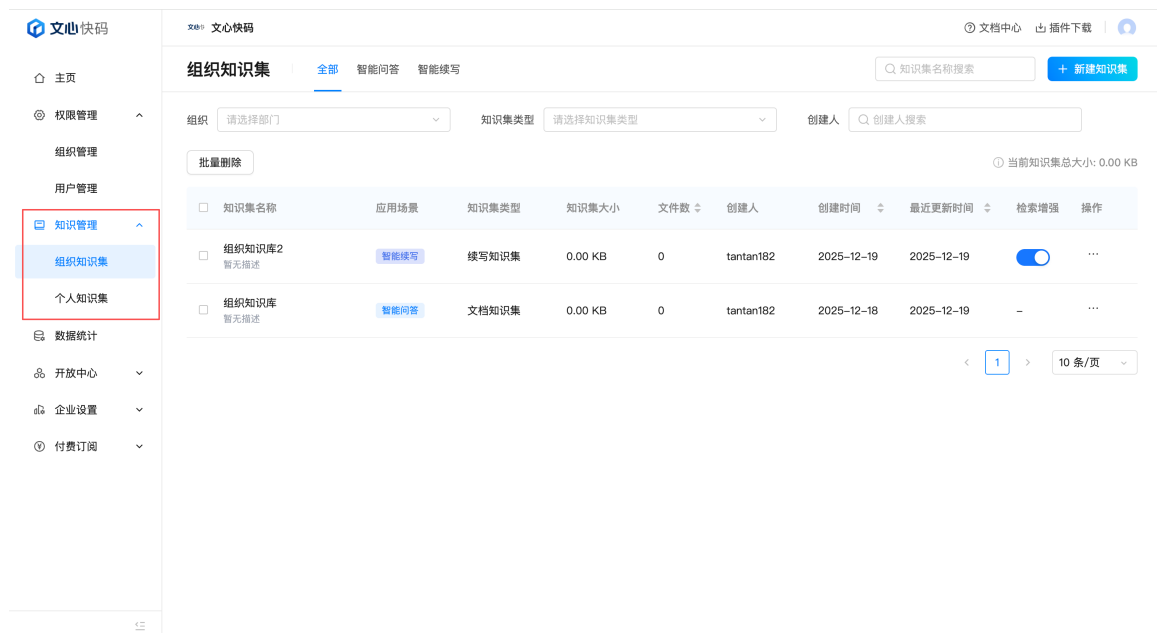
页面默认展示当前企业用户创建的全部知识集，排列方式根据创建时间先后顺序倒排。不同知识集类型通过标签区分，也可以通过筛选框单独查看，以下为知识集类型与对应的权限。

- 按插件应用场景划分，以保证插件侧问答、续写两个场景的检索生成效果：
  - 智能问答知识集，主要用于智能体私域知识问答功能
  - 智能续写知识集，主要用于代码续写增强功能



- 按具体使用权限划分，各业务单元可以做到知识隔离：

- 组织知识集（组织可见）：被管理员创建知识集时赋予权限的部门具有该组织知识集的管理、可读权限；创建知识集后，管理员依然可以为某个部门、或某个企业成员赋予管理、可读权限
- 个人知识集（仅我可见）：个人自动拥有个人知识集管理和可读权限



注：知识集可见范围=知识集使用权限

## 3. 知识定义

- 文档知识：知识是一篇文档（如一篇Markdown文档、一篇PDF文档、一篇txt文档等），一系列文档知识组装成文档知识集。

- 代码知识：知识是一个zip格式打包的代码库，或一个独立代码文件，一系列代码知识组装成代码知识集。

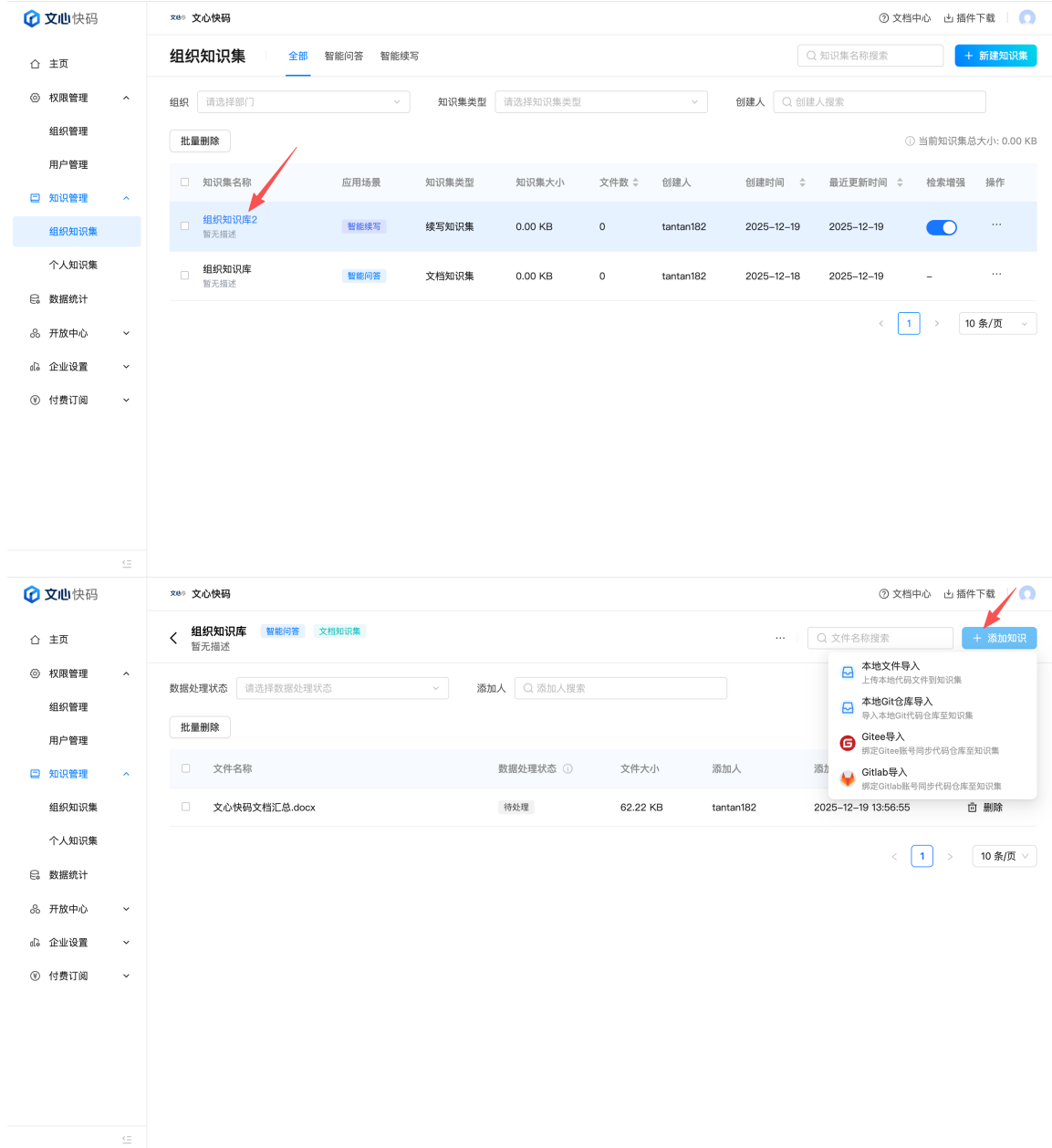
#### 4.知识准备规则

- 按照私域知识问答、代码续写增强两类应用场景分别准备对应的文档和代码文件。
- 推荐按照类别的不同拆分成不同的知识集（如A应用的API文档、B应用的API，拆成两个不同的知识集）
- 如果私域知识存在多个版本，建议拆分成不同的知识集，如 A应用API 1.0、A应用API 2.0、A应用API 3.0等

#### 知识管理

#### 添加知识

当您已完成知识集的创建后，可点击知识集名称进入知识集，完成知识文件的上传。



#### 内容要求

##### 1.问答场景

- 内容要求：支持上传需求文档、开发规范、API文档等内容。
- 格式要求：文件支持：pdf、md、txt、docx(doc)、xlsx、pptx(ppt)、csv、xml、sql、zip、gz格式。对Markdown格式文件进行的特殊调优，推荐上传Markdown格式文件，效果更佳！

## 2.续写场景

- **内容要求**：支持上传Java、Python、JS、TS、C++、Go等语言内容。
- **格式要求**：文件支持.zip、.gz、.tar、.gz格式，文件小于100M。

### \*\*文档类知识内容建议\*\*

- **文字描述丰富**：Comate背后的模型是语言（文本）模型，如果文档中存在较多的图片，会大大降低检索的效果，因此文档中尽可能减少图片、增加详实的文字描述，将会显著提高检索质量
- **文档逻辑清晰**：高质量的数据不单单在大模型的训练中极其重要，在通过RAG强化模型输出效果时也极其重要。因此，文档内容有层次、语言描述言简意赅无歧义、每个段落有标题等结构，会显著提高检索质量
- **目录结构清晰**：如果您有大量的文档分散在不同的目录中，您可以直接将根目录打成压缩包上传为知识，Comate将自动解压缩读取每篇文档。为了保证更好的检索效果，建议每个层级目录的命名言简意赅，能够充分表达这一层级目录的文档类别。
- Markdown格式的内容效果最好。

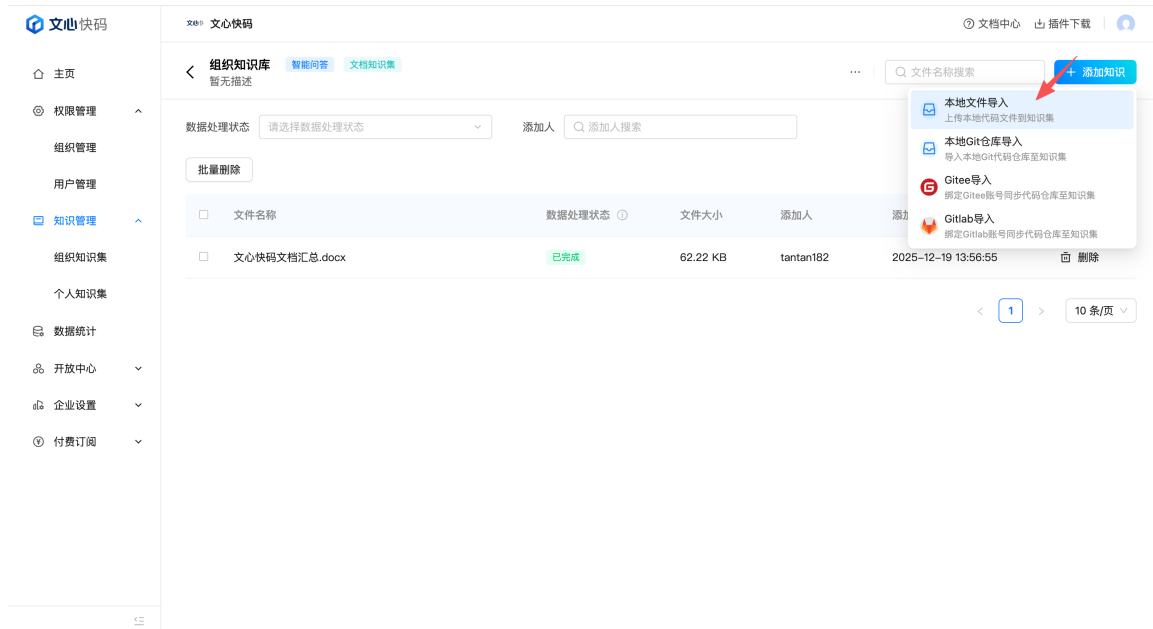
### \*\*代码类知识内容建议\*\*

- 上传高质量的代码
  - 高频次出现的、多次被引用的、符合规范的代码片段
  - 例如
    - 后台核心业务组件、可复用的工具
    - 前端公共模板库代码、通用组件代码
  - 包含函数注释，或者函数名规范（可清晰描述函数功能）
- 避免干扰性代码或大量质量不高的代码
  - 避免上传测试代码、MOCK 数据、构建产物等
  - 同一知识集内，若上传多个代码库或代码文件，请确保属于同一项目，以保证检索效果。
- 目前支持的文件类型：

'java', # Java '.py', # Python '.go', # Go '.js', # JavaScript '.ts', # TypeScript '.c', # C '.h', # C/C++ 头文件 '.cpp', # C++ '.cc', # C++ '.cxx', # C++ '.hpp', # C++ 头文件 '.hh', # C++ 头文件 '.hxx' # C++ 头文件

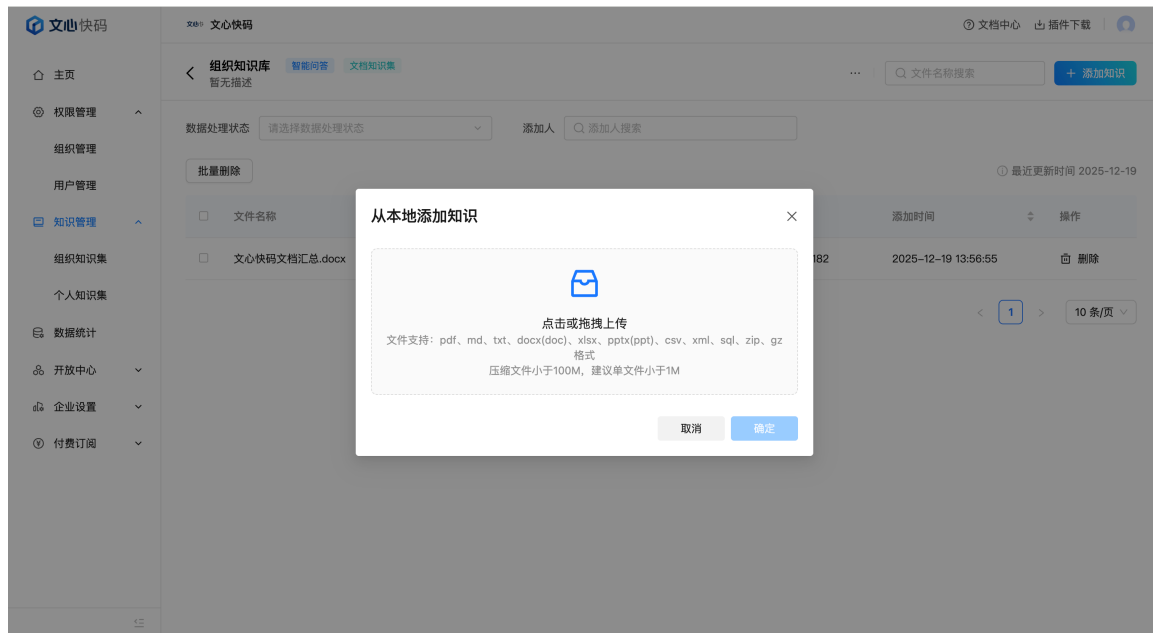
## 🔗 上传方式

知识文档支持多种上传方式，包括本地文件导入、本地Git仓库导入、Gitee导入和Gitlab导入。



1.本地

文件导入 文件支持：pdf、md、txt、docx(doc)、xlsx、pptx(ppt)、csv、xml、sql、zip、gz格式 压缩文件小于100M,建议单文件小于1M

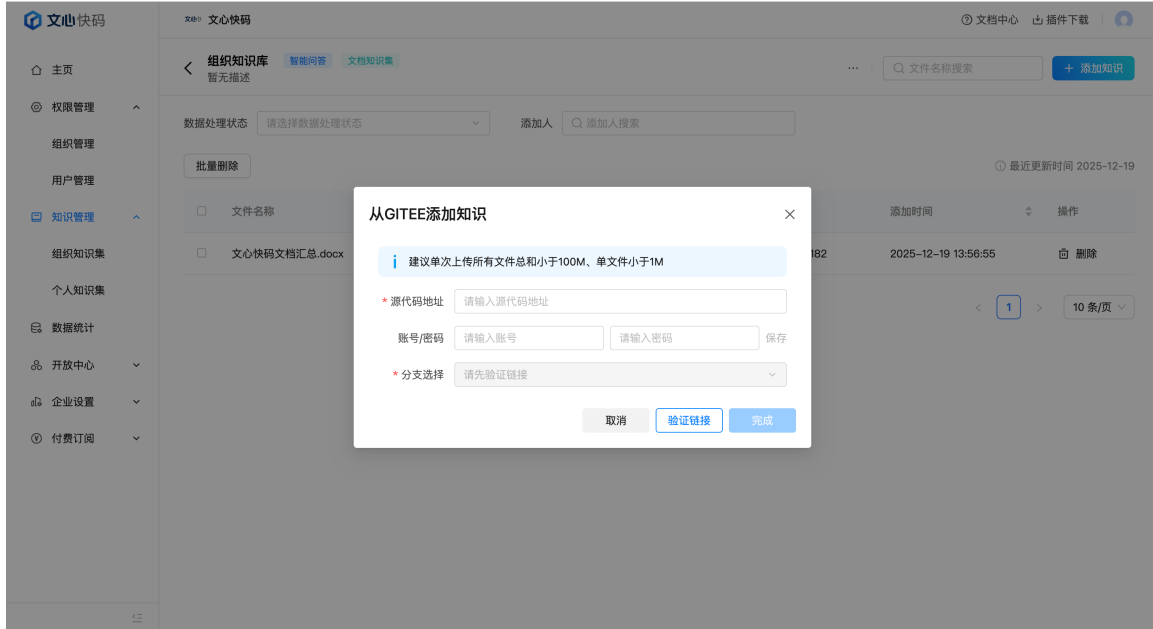


2.本地

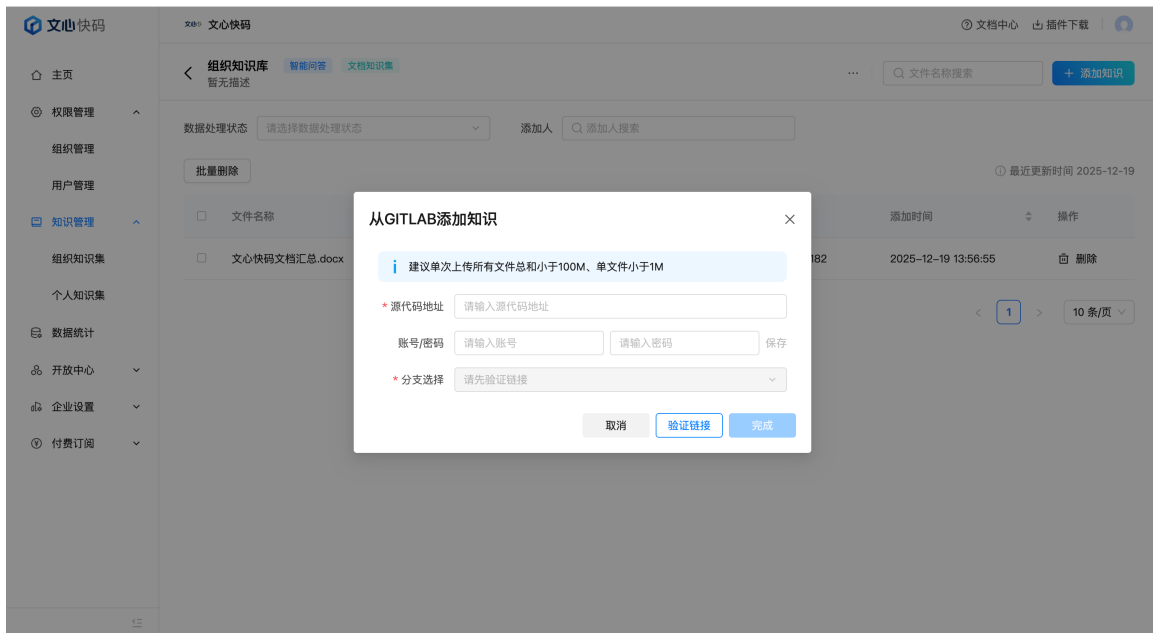
Git仓库导入 建议单次上传所有文件总和小于100M、单文件小于1M 需要输入源代码地址、账号/密码，填写后选择分支



### 3.Gitee导入 建议单次上传所有文件总和小于100M、单文件小于1M 需要输入源代码地址、账号/密码，填写后选择分支

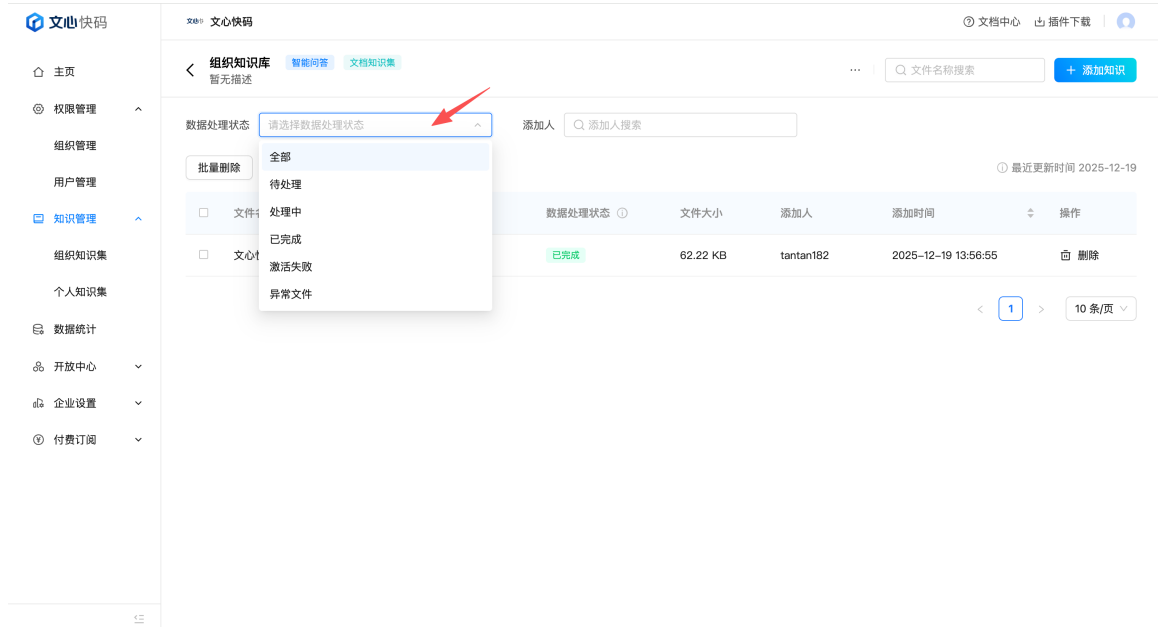


### 4.Gitlab导入 建议单次上传所有文件总和小于100M、单文件小于1M 需要输入源代码地址、账号/密码，填写后选择分支



#### 🔗 知识上传状态

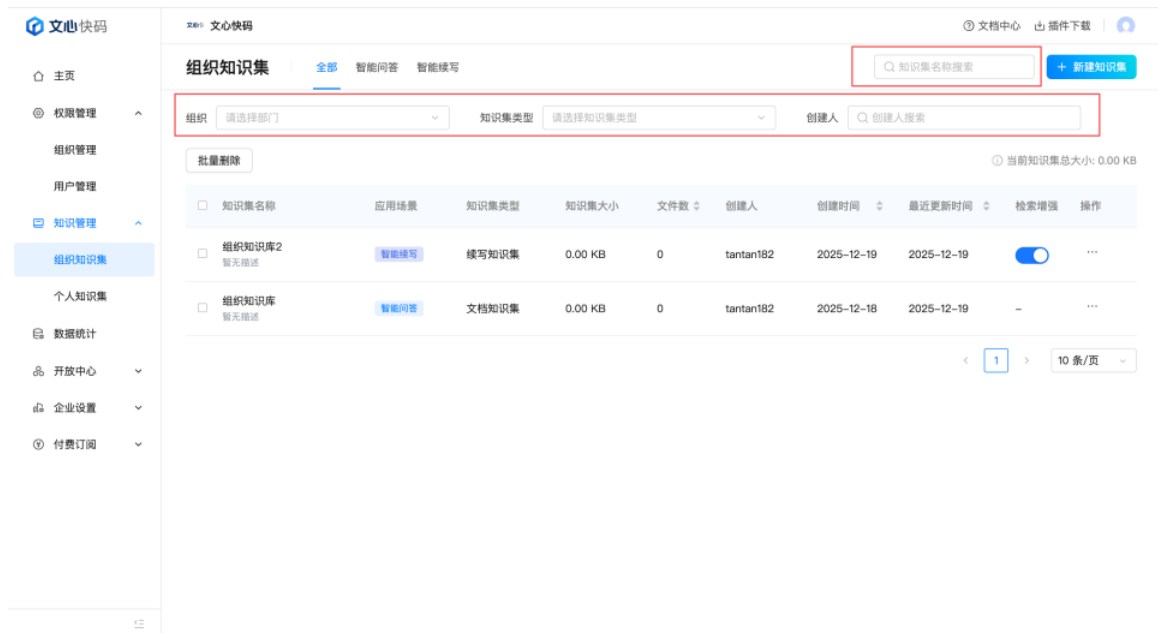
- **【待处理】**：代表系统将有序对上传的文件进行处理，未开始处理的进入排队状态。
- **【处理中】**：处理过程中，请等待
- **【已完成】**：代表上传成功。
- **【激活失败】**：在知识集上传阶段出现异常，可以进行重新上传重试。
- **【异常文件】**：上传状态为【异常】代表文件处理出现异常，聚焦在状态标记上可查看原因，此时可结合不同的情况进行重新处理或删除异常文件。上传状态为【重复文件】文件内容重复，不进行重复处理，可删除重复文件。



## 知识集管理

### 1. 查看知识集

组织知识集和个人知识集都支持多条件查询：组织结构、知识集类型、创建人，同时也可以点击右上角按照知识集名称进行查询



### 2. 新增知识集

点击新增知识集，用户根据所需使用场景（智能问答、智能续写）、所需使用范围（个人知识集、组织知识集）可以选择需要上传的知识集类型

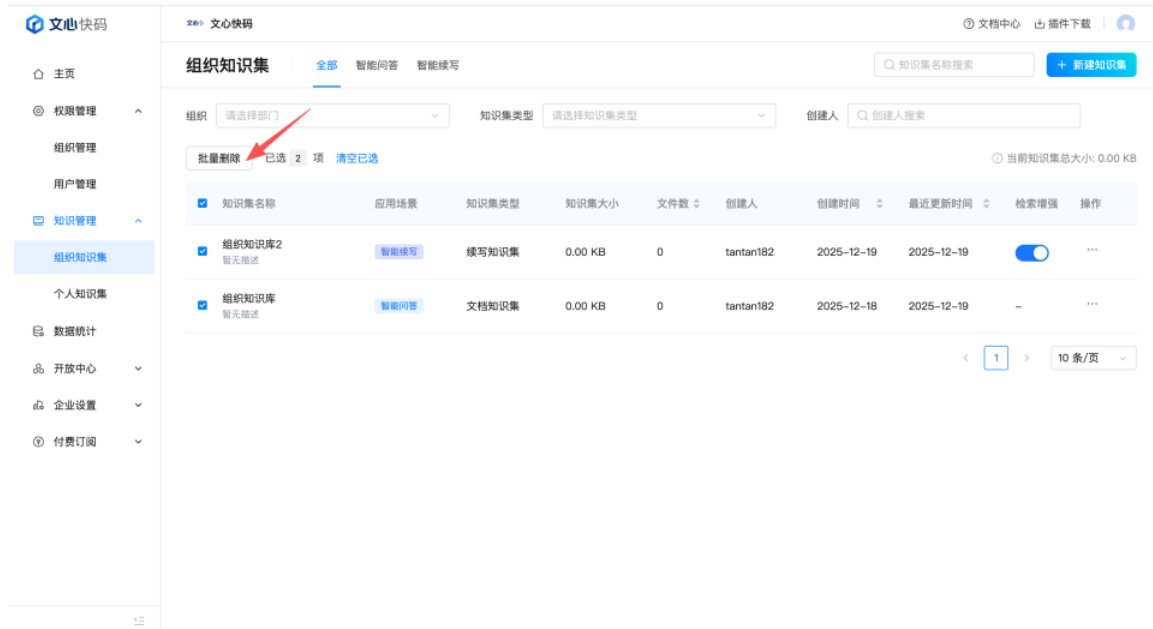


仅我可见：创建个人知识集，仅支持个人进行增删改查；其他管理员不可见  
 组织可见：创建组织知识集，并选择具体部门赋予管理权限



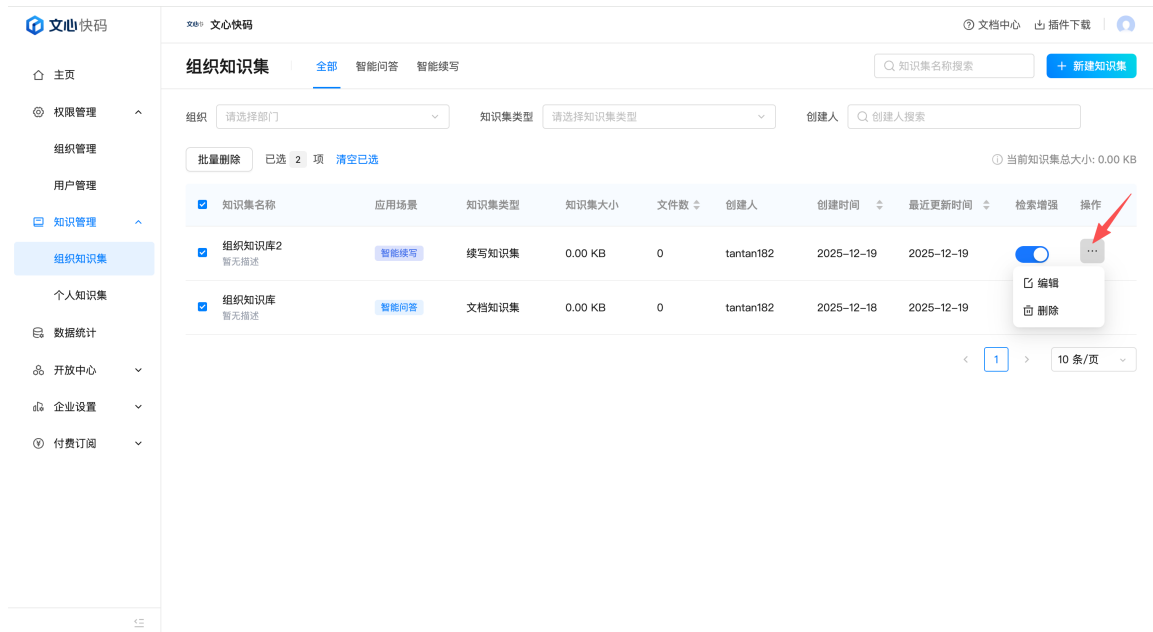
### 3. 批量删除知识集

勾选多个知识集后，可以点击批量删除



- 编辑知识库：点击表的操作栏目，可以对知识库进行编辑/删除

编辑知识库无法变更知识库应用场景以及类型（无法点击），可以修改知识库可见部门、名称、描述



## 数据统计

数据统计功能呈现了 Comate 在企业内部的使用情况。该模块汇总用户活跃趋势、采纳行数等关键指标，便于评估 Comate 推荐代码的接受度与质量；同时追踪生成代码占比、续写与问答次数等，全面评估 Comate 在编码流程中的贡献，助力企业量化 Comate 价值，优化开发流程，提升代码编写质量与效率。

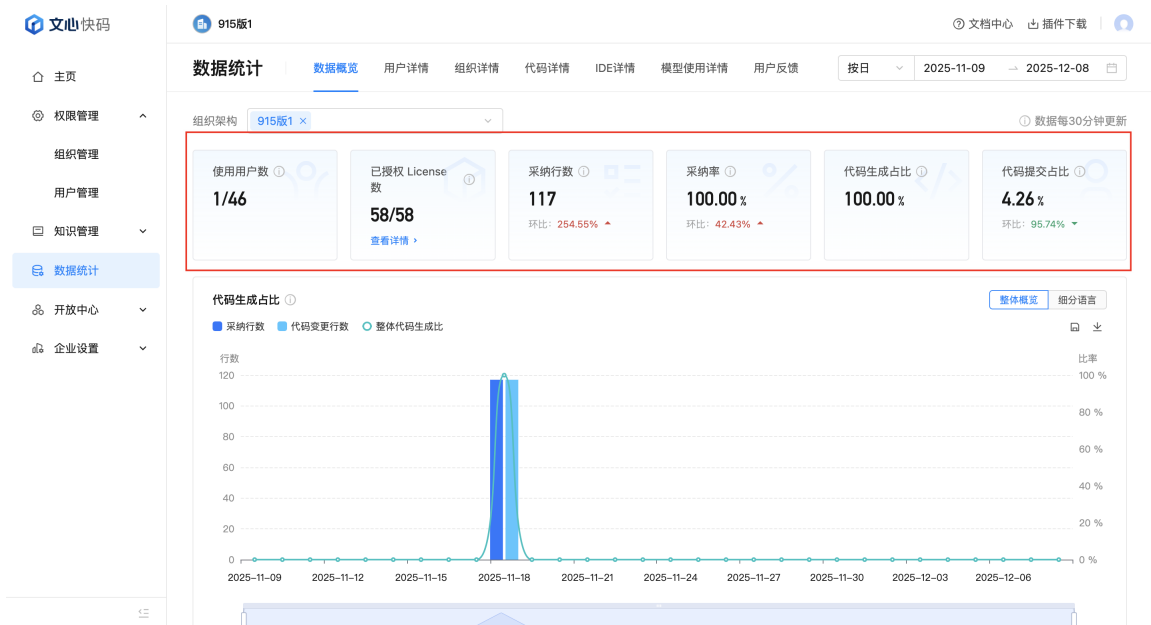
### 一、数据维度

针对企业内部各部门进行筛选，筛选后仅展示所选部门的 Comate 使用数据。



展示所选时间和组织架构内，企业使用 Comate 的核心数据和图表，具体数据说明如下：

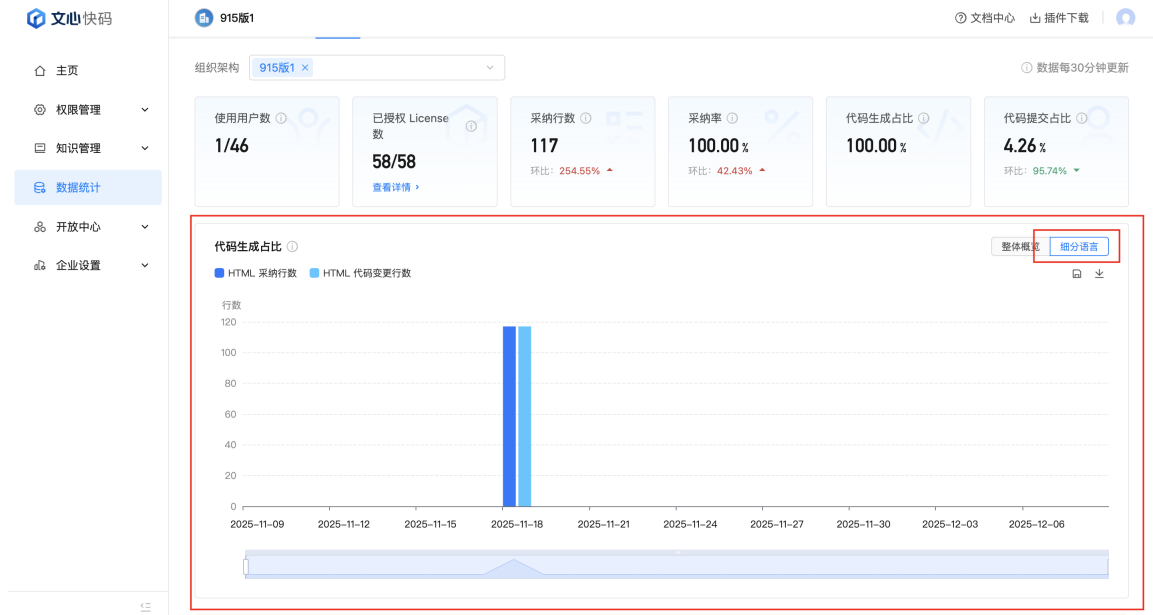
- 使用用户数：所选时间范围内使用过的用户数/全部加入过的用户数
- 已授权 License 数：授权过的License数量/全部有效License，数据计算包括解绑的License
- 采纳行数：用户采纳的行数(包括代码补全场景中点击 Tab 和 Agent 场景中采纳/部分采纳/复制/插入/新建)
- 采纳率：采纳行/推荐行数（包括代码补全场景、Agent场景）
- 代码生成占比：按照实时变更计算，代码生成占比=采纳后未删除的代码行/代码变更行数
- 代码提交占比：按Git提交计算，代码提交占比=采纳后未删除的代码行/代码变更行数



## 二、数据概览

### 1. 代码生成占比趋势图

展示所选时间和组织架构内，整体代码生成占比、采纳行数、变更行数的变化趋势图。横坐标为时间，纵坐标为比率、行数（可下载图表图片或数据表文件）；点击细分语言可以查看细分代码语言的统计数据。



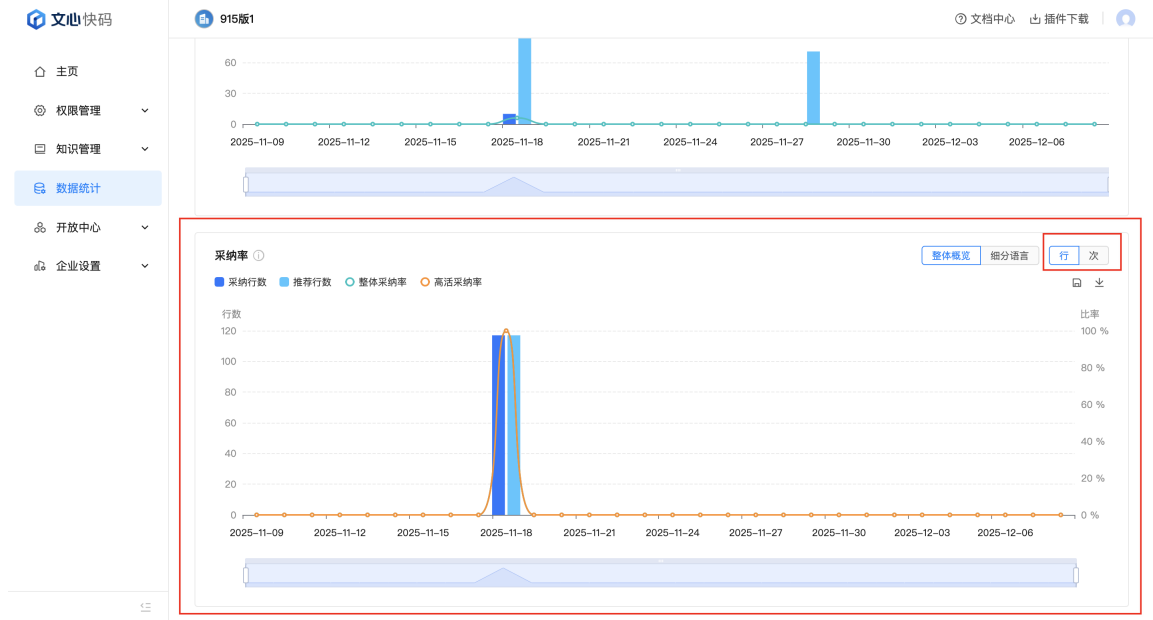
### 2. 代码提交占比趋势图

展示所选时间和组织架构内，整体代码提交占比、采纳行数、变更行数的变化趋势图。横坐标为时间，纵坐标为比率、行数（可下载图表图片或数据表文件）。



### 3. 采纳率趋势图

展示所选时间和组织架构内，整体采纳率、高活采纳率、采纳行数、推荐行数的变化趋势图。横坐标为时间，纵坐标为比率、行数/次数（可下载图表图片或数据表文件）；点击右上角可以切换数据统计方式为行数/次数；点击细分语言可以查看细分代码语言的统计数据。



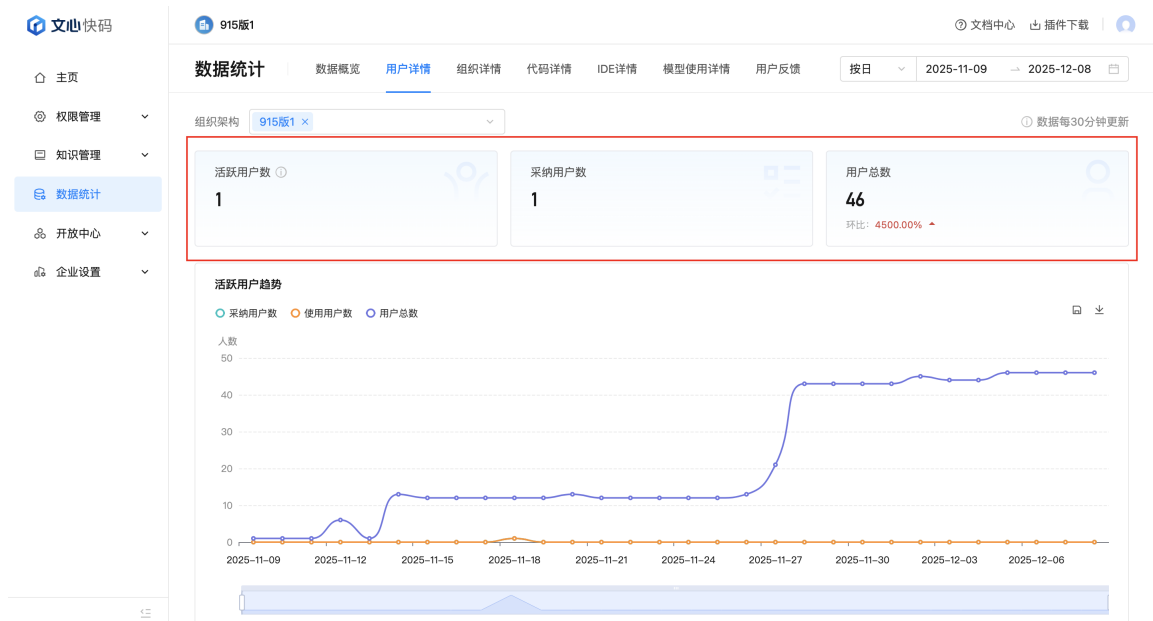
### 三、数据详情

数据详情页面旨在通过多维度的数据统计和可视化，帮助企业管理者全面了解 Comate 在企业内部的渗透度、使用效率和用户行为模式。

#### 1. 用户详情

展示所选时间和组织架构内，企业内用户使用 Comate 的使用数据、图表和具体明细，具体数据说明如下：

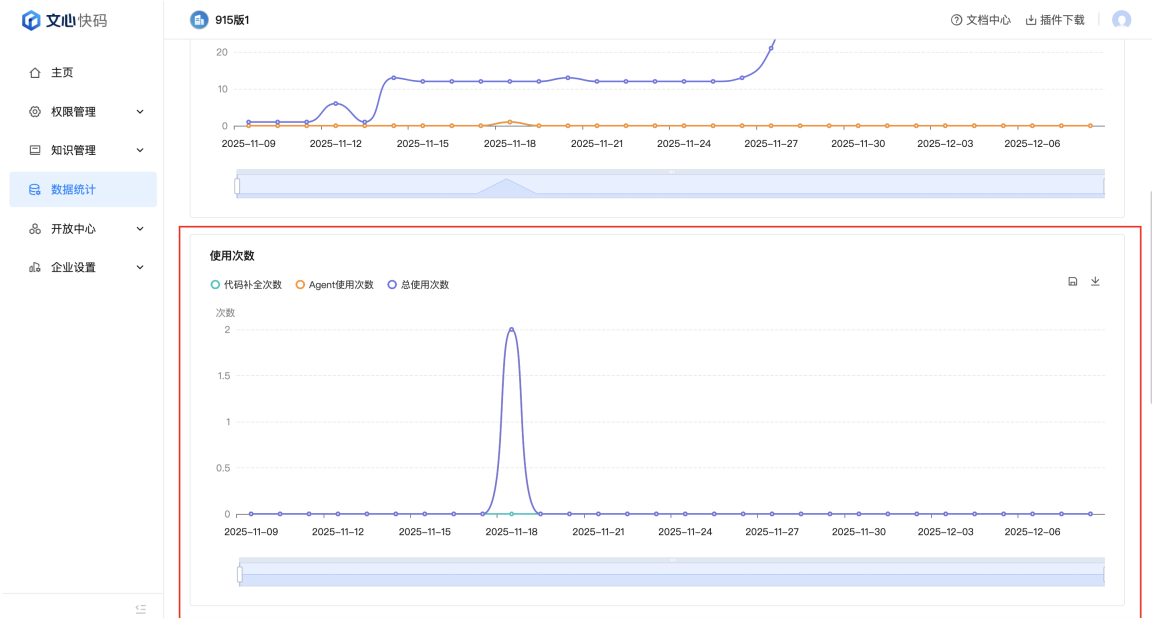
- 活跃用户数：已使用的用户数量
- 采纳用户数：采纳 Comate 生成代码的用户数量
- 用户总数：当前组织架构中全部成员数量



- 活跃用户趋势图：展示所选时间和组织架构内，采纳用户数、使用用户数、用户总数的变化趋势图。横坐标为时间，纵坐标为人数（可下载图表图片或数据表文件）。



- **使用次数趋势图**，展示所选时间和组织架构内，代码补全次数、Agent使用次数、总使用次数的变化趋势图。横坐标为时间，纵坐标为次数（可下载图表图片或数据表文件）。



## 2. 用户明细

展示企业至今为止 Comate 使用情况的详细累计数据，可以下载为表格文件查看，包含：

- 用户名
- 邮箱
- 部门
- 采纳行：用户采纳Comate所有功能推荐的代码总行数（续写、问答场景相加）
- 推荐行：Comate所有功能推荐的代码总行数（续写、问答场景相加）
- 采纳率：用户采纳 Comate 推荐代码的行数占 Comate 推荐代码总行数的比例
- 代码变更行数：用户在git仓库提交的代码行数，包含删除+新增
- 代码生成占比：用户编写代码的场景中，用户采纳 Comate 生成的代码行数占用户变更代码行数的比例 此外，用户明细页面的字段选取，可以记录每个用户上一次选择的字段记录，支持多选，底部恢复默认选项，点击后可以回到默认的字段选取。

用户名	邮箱	部门	用户状态	是否使用	采纳率(行)	采纳率(次)	代
研发中心-张三	zhangsan@test.com	LDAP研发中心	已授权	否			
运维部-部七	zouqi@test.com	LDAP运维部	已授权	否			
指令test	zitest@123.com	LDAP研发中心	已授权	否			
测试部-赵六	zhaoliu@test.com	LDAP测试部	已授权	否			
张知识2	zhangzhishi2@123....	LDAP测试部	已授权	否			
张知识1	zhangzhishi1@123....	运营部运营部运营部运营部运营...	已授权	否			
运成	yuncheng@123.com	LDAP运维部	已授权	否	0%	0%	
yn-test	yn@baidu.com	LDAP研发中心	已授权	否	0%	0%	
营管	yingguan@123.com	营销组	已授权	否	0%	0%	
营成	yingcheng@123.com	营销组	已授权	否	0%	0%	

### 3.组织详情

展示所选时间和组织架构内，企业部门的整体使用数和具体明细，具体数据说明如下：

- 采纳率：采纳行/推荐行数（包括代码补全场景、Agent场景）
- 采纳行数：用户采纳的行数（包括代码补全场景中点击 Tab 和 Chat 区点击采纳或插入）
- 推荐行数：Comate所有功能推荐的代码总行数（续写、问答场景相加）
- 代码生成占比：按照实时变更计算，代码生成占比=采纳后未删除的代码行/代码变更行数
- 变更行数：用户在git仓库提交的代码行数，包含删除+新增
- 使用用户数：在周期和组织范围内，使用插件的用户数量

子部门名称	采纳率(行)	采纳率(次)	代码补全采纳率(行)	Agent采纳率(行)	Agent留存采纳率(行)	代码生成占比
研发部	0%	0%	0%	0%	0%	0%
LDAP研发中心	0%	0%	0%	0%	0%	0%
运营部运营部运营部运营...	0%	0%	0%	0%	0%	0%
LDAP总裁办公室	0%	0%	0%	0%	0%	0%
测试2	0%	0%	0%	0%	0%	0%
商业部	0%	0%	0%	0%	0%	0%

### 4.部门明细

展示企业部门至今为止 Comate 使用情况的详细汇总数据，可以下载为表格文件查看，包含：

- 子部门名称
- 采纳率（行）：用户采纳 Comate 推荐代码的行数占 Comate 推荐代码总行数的比例
- 采纳率（次）：用户采纳 Comate 推荐代码的次数占 Comate 推荐代码总次数的比例

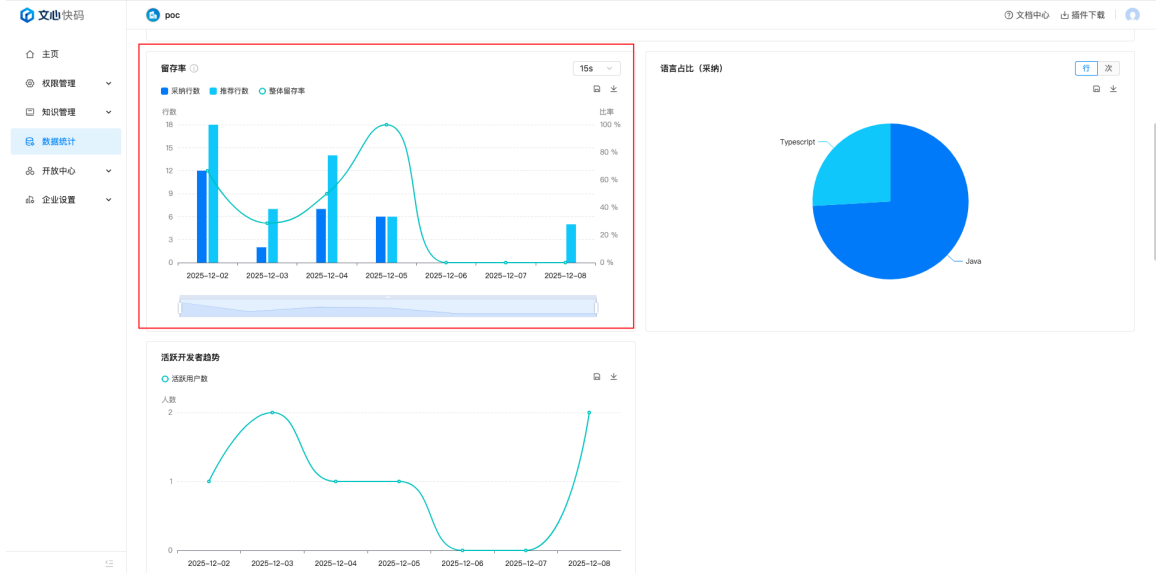
- 代码补全采纳率（行）：代码补全场景下，用户采纳 Comate 推荐代码的行数占 Comate 推荐代码总行数的比例
- Agent采纳率（行）：智能问答场景下，用户采纳 Comate 推荐代码的行数占 Comate 推荐代码总行数的比例
- Agent留存采纳率（行）：智能问答场景下，用户采纳并留存 Comate 推荐代码的行数占 Comate 留存推荐代码总行数的比例
- 代码生成占比：用户编写代码的场景中，用户采纳 Comate 生成的代码行数占用户变更代码行数的比例
- 使用用户数：当前组织架构中已使用的用户数量
- 采纳用户数：当前组织架构中采纳 Comate 生成代码的用户数量
- 总使用次数：当前组织架构中用户使用Comate的总次数
- Agent使用次数：当前组织架构中用户使用智能问答Agent的总次数
- Agent采纳次数：当前组织架构中用户采纳智能问答Agent推荐代码的总次数
- 自定义智能体使用次数：当前组织架构中用户使用自定义智能体的总次数



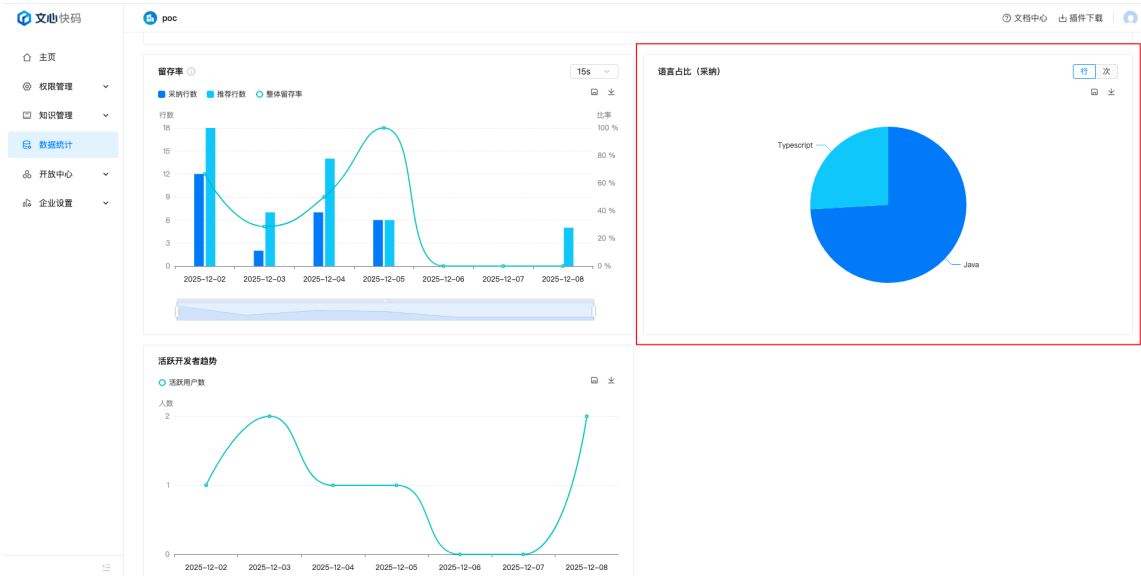
## 5. 代码详情

展示所选时间和组织架构内，在代码补全和智能问答两个功能场景分别的用户使用数据图表，具体数据说明如下：代码补全功能

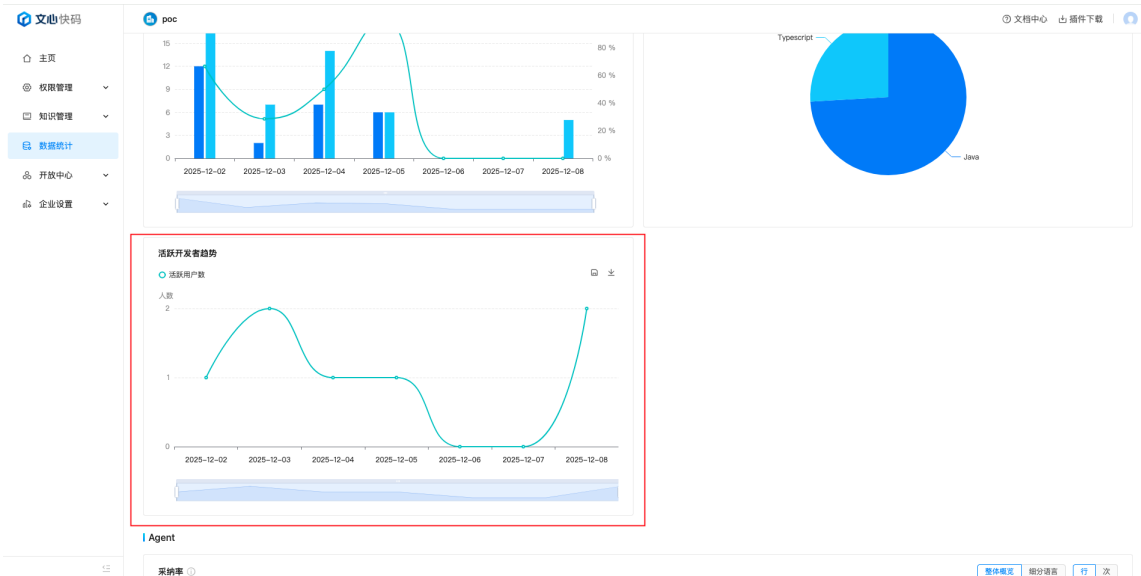
- 采纳率趋势图：代码补全场景下，采纳行数、推荐行数、整体采纳率的变化趋势图。右上角可以切换统计口径以及细分语言
- 留存率：代码补全场景下，计算采纳后未删除的代码行数，推荐行数和整体留存率的比例变化趋势图，右上角可以切换统计时间口径



● 语言占比：代码补全场景下，使用comate生成代码的企业用户使用的编码语言占比分布图



● 活跃开发者趋势图：代码补全场景下，使用comate的活跃开发者数的变化趋势图

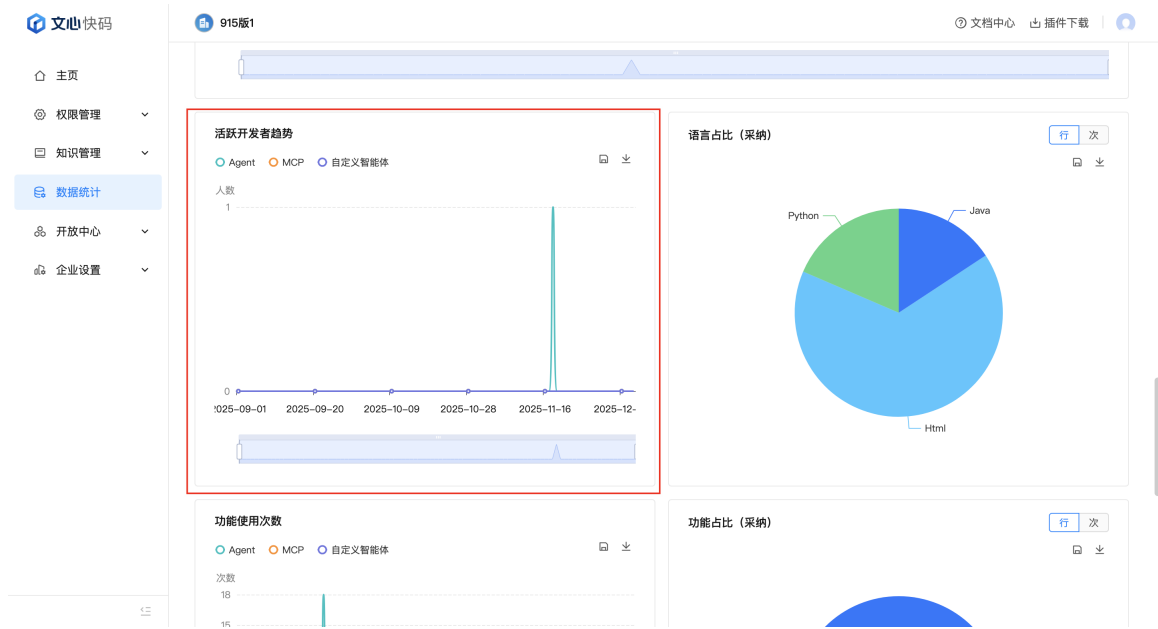


### 6. 智能问答详情

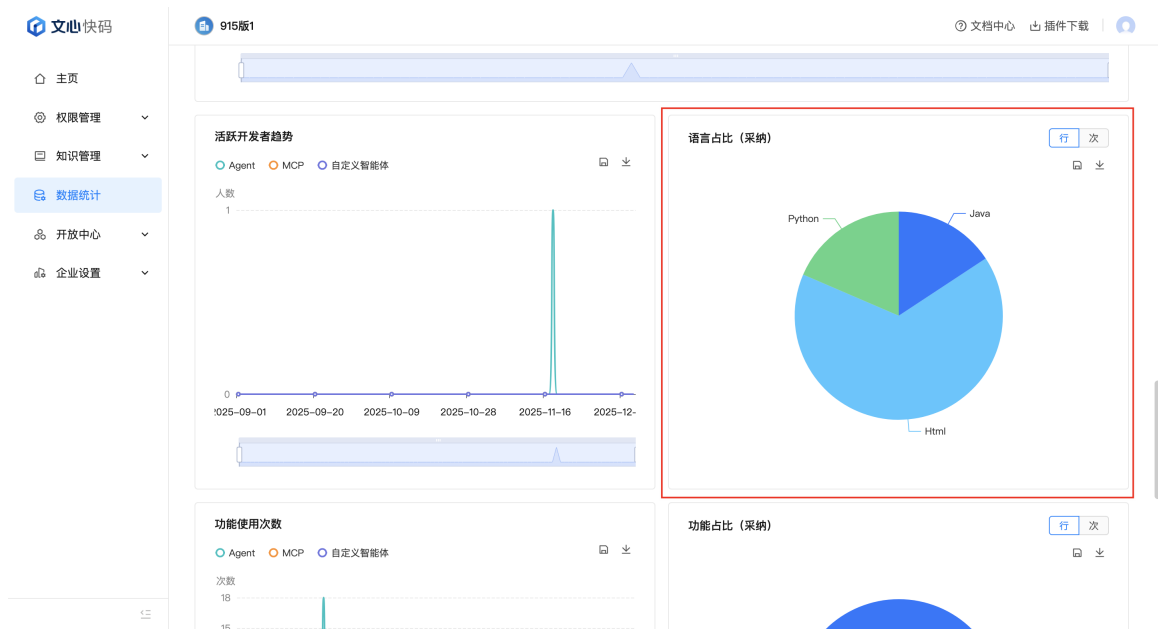
● 采纳率趋势图：智能问答场景下，采纳率、推荐行数、整体采纳率、留存采纳率、留存推荐行数、留存采纳率的变化趋势图；点击右上角可以切换为细分语言的数据统计柱状图



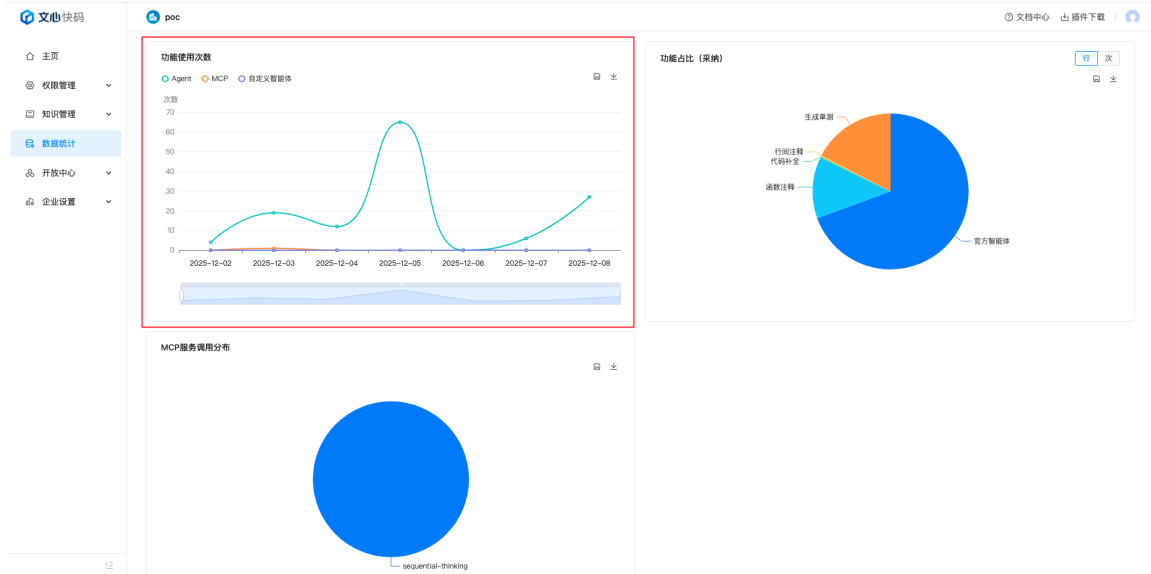
● **活跃开发者趋势**：智能问答场景下，使用Agent、MCP、自定义智能体的活跃开发者数的变化趋势



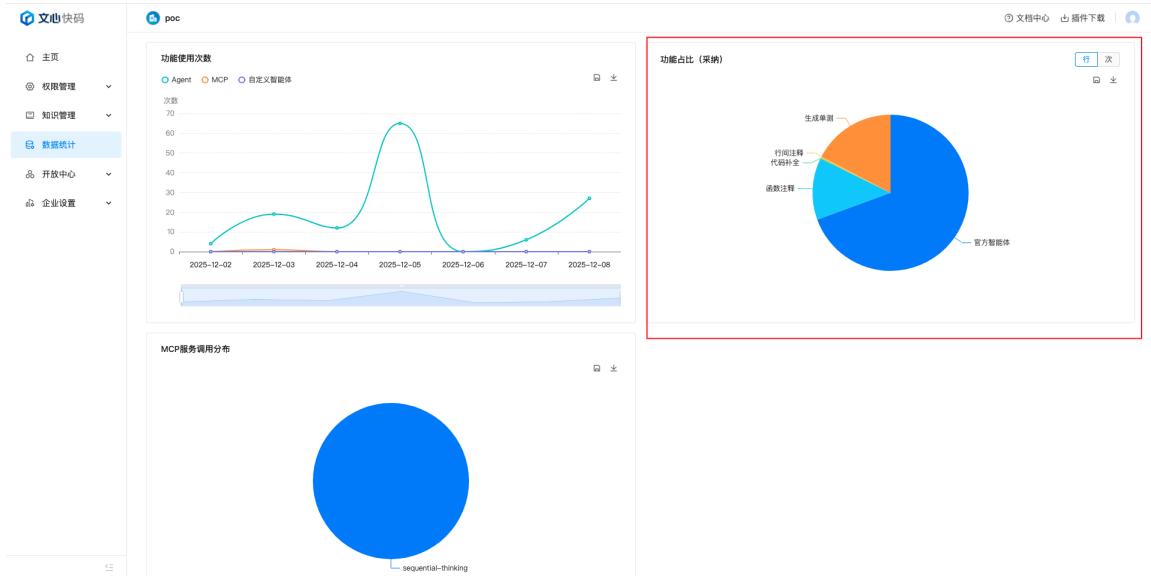
● **语言占比 (采纳)**：智能问答场景下，企业用户分别使用Java、Html、Python等不同代码语言生成代码的采纳次数/采纳率占比图



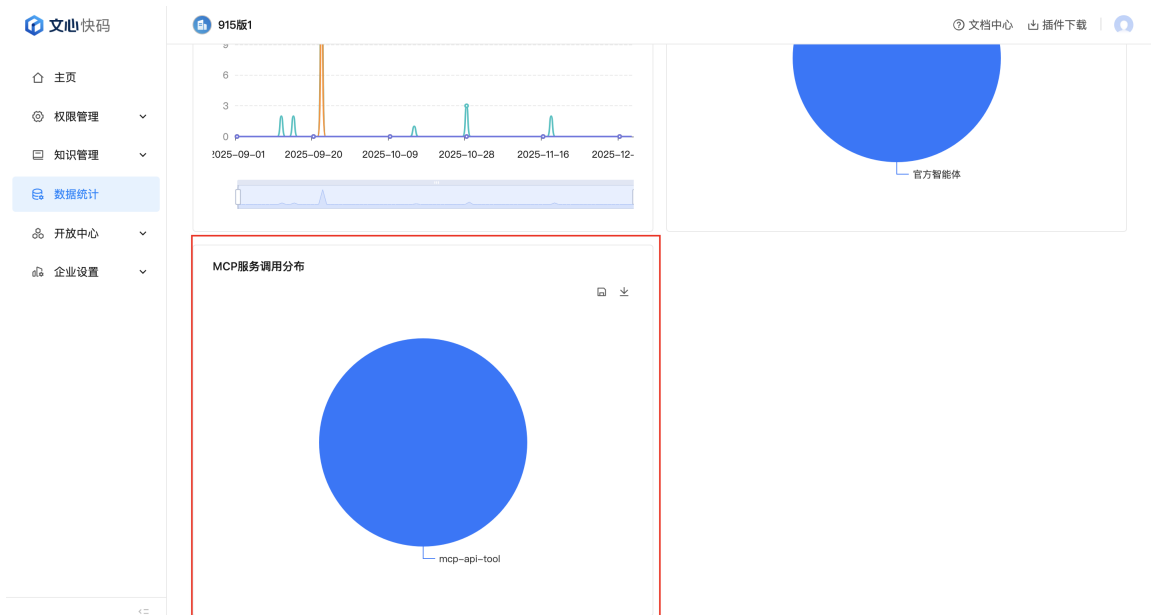
● 功能使用次数：智能问答场景下，Agent、MCP、自定义智能体的使用次数折线图



● 功能占比 (采纳次数/采纳行数)：智能问答场景下，企业用户分别使用行间注释、生成单测、函数注释等问答类功能生成代码的采纳次数/采纳行数占比图



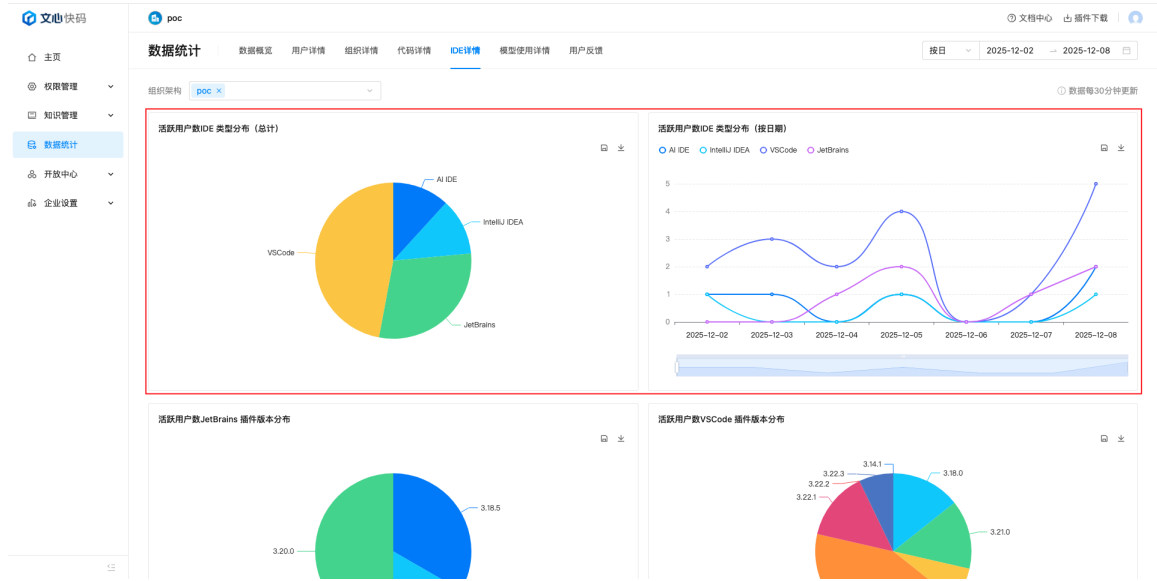
● MCP服务调用分布：智能问答场景下各个MCP调用次数



### 7.IDE详情

展示所选时间和组织架构内，企业用户使用的IDE类型和插件版本，具体数据说明如下：

- IDE类型分布图：展示所选时间和组织架构内，用户使用不同IDE的人数占比图，以及不同日期下使用不同IDE的人数分布图，横坐标为时间，纵坐标为人数。



- 不同IDE使用的插件版本分布图：展示所选时间和组织架构内，使用 JetBrains 和 VSCode 以及使用AI IDE的用户正在使用Comate 的插件/IDE版本分布图。



细 展示企业用户至今为止 Comate 插件版本使用情况的详细数据，可以下载为表格文件查看，包含：

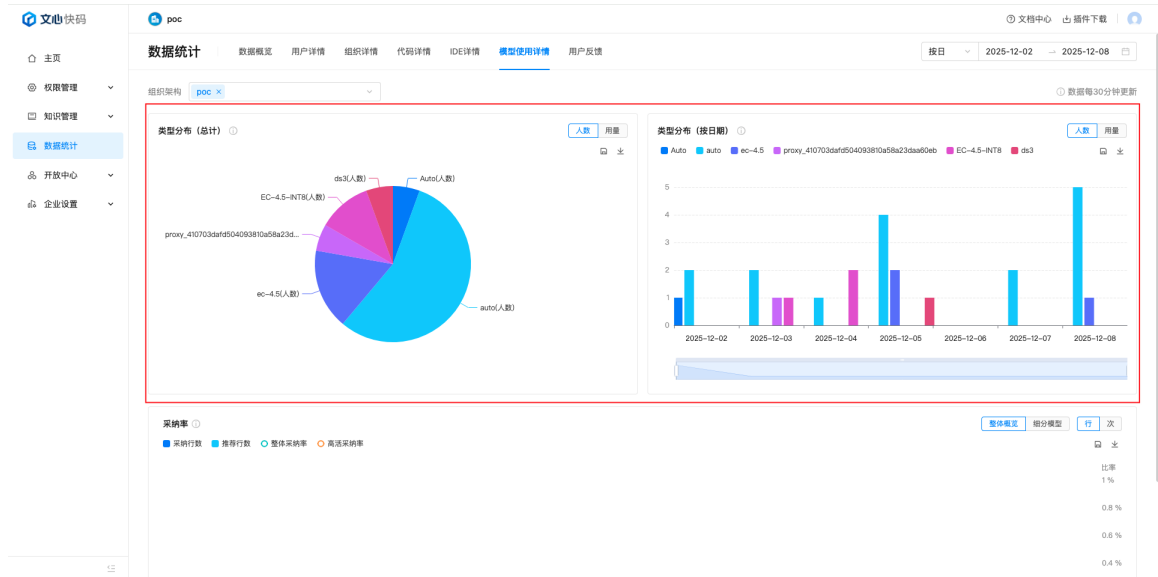
- 用户名
- 部门
- IDE 类型
- 插件版本
- 采纳率
- 代码生成占比



### 8.模型使用情况

展示所选时间和组织架构内，企业用户使用的模型类型，具体数据说明如下：

- **模型类型分布图**：展示所选时间和组织架构内，用户使用不同模型的人数/用量占比图，以及不同日期下使用不同模型的人数/用量分布图，横坐标为时间，纵坐标为人数/用量。
  - 人数：调用不同模型的去重用户数量
  - 用量：调用不同模型的内容生成次数



- **采纳率**：展示所选时间和组织架构内，用户采纳函数、推荐行数、整体采纳率、高活采纳率的变化趋势图。点击细分模型可以查看不同模型的详细采纳数据

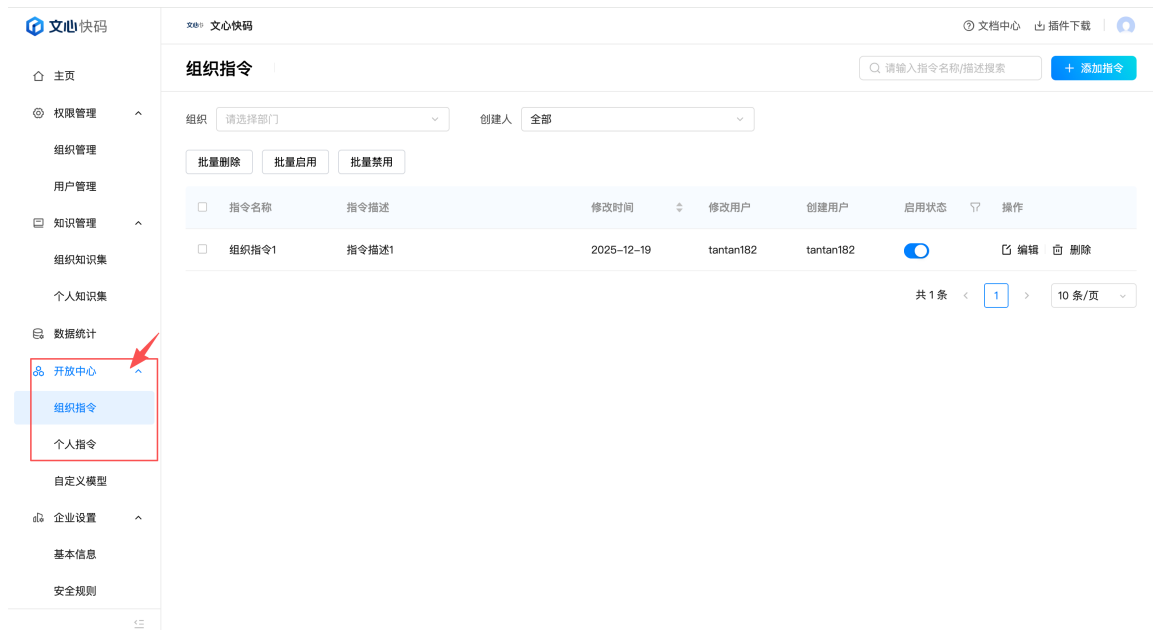


## 开放中心

### 指令管理

指令管理页面聚焦于组织 / 个人自定义指令的全生命周期管理，为企业内指令配置与使用提供一站式解决方案。核心价值体现在“管理便捷化 + 使用高效化”。

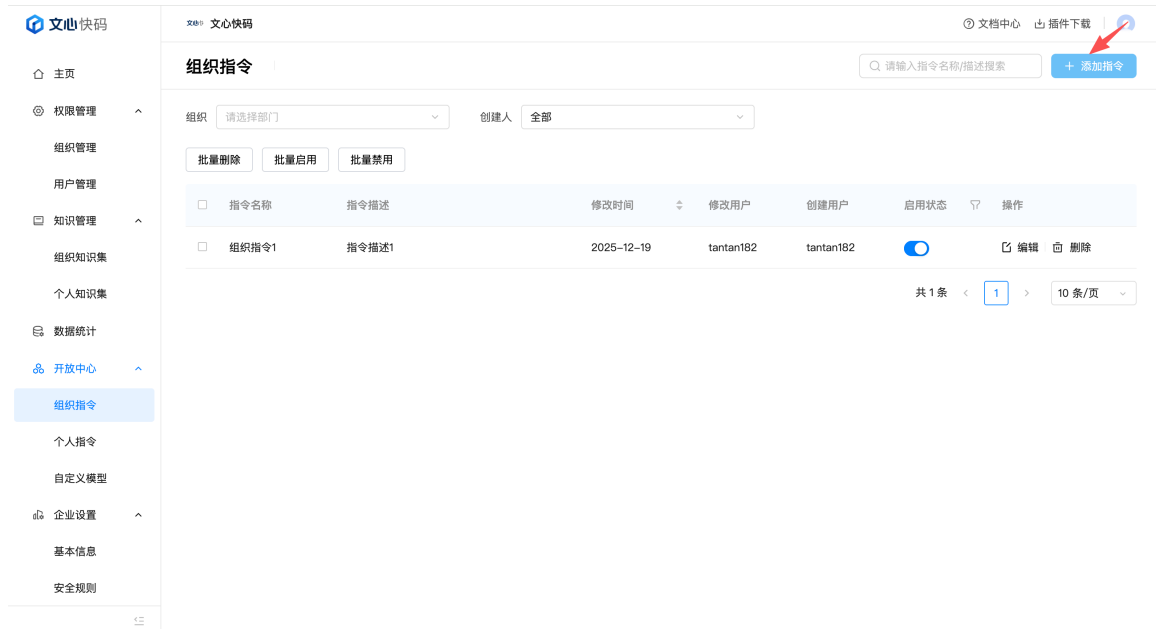
**1. 查看指令** 用户可以在企业管理后台>>开放中心>>组织/个人指令页面，查看并管理指令。用户可以在文心快码Comate插件中，通过输入“/”唤起自定义指令，并选择要使用的指令。



2. 添加

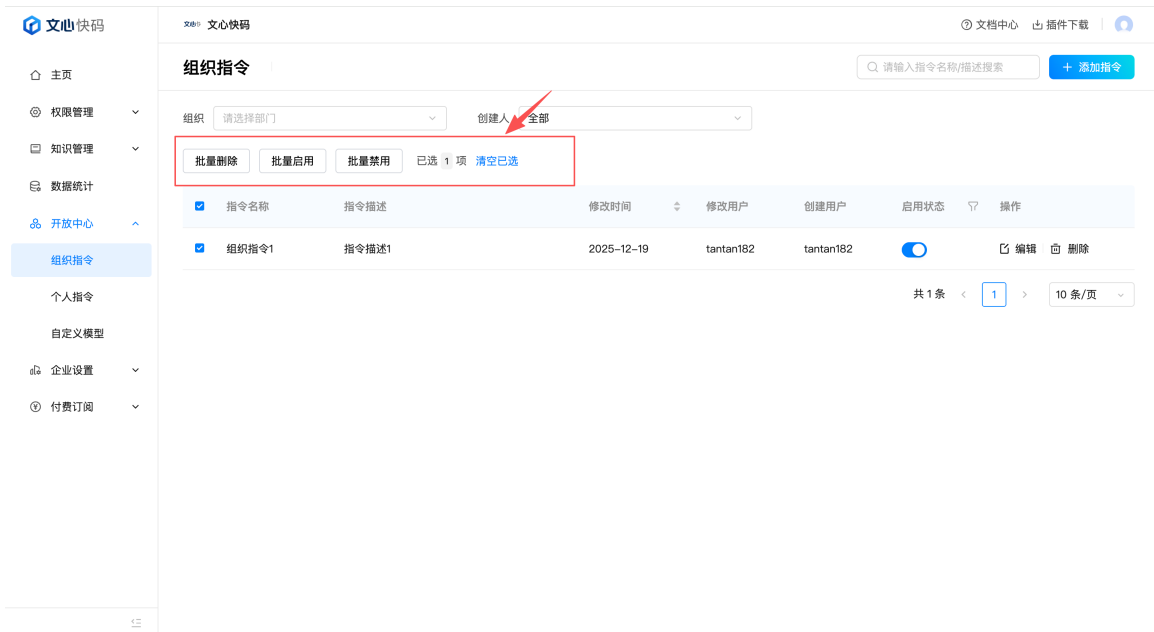
**指令 第一步：**点击「组织/个人指令」子页面右上角「添加指令」按钮，新建自定义指令 **第二步：**填写指令名称、指令描述、可见部门（新建个人指令无此项）、指令内容，即可保存自定义指令

- 指令名称：限10字符
- 指令描述：限40字符
- 可见部门：可在企业范围内进行选择（新建个人指令无此项）
- 企业内用户在文心快码Comate插件中，可以看到指令的「指令名称、指令描述」
- 指令内容：限5000字符，描述指令的具体提示词内容，在填写「指令内容」时，可以点击「插入案例」按钮，并在此基础上对内容进行编辑



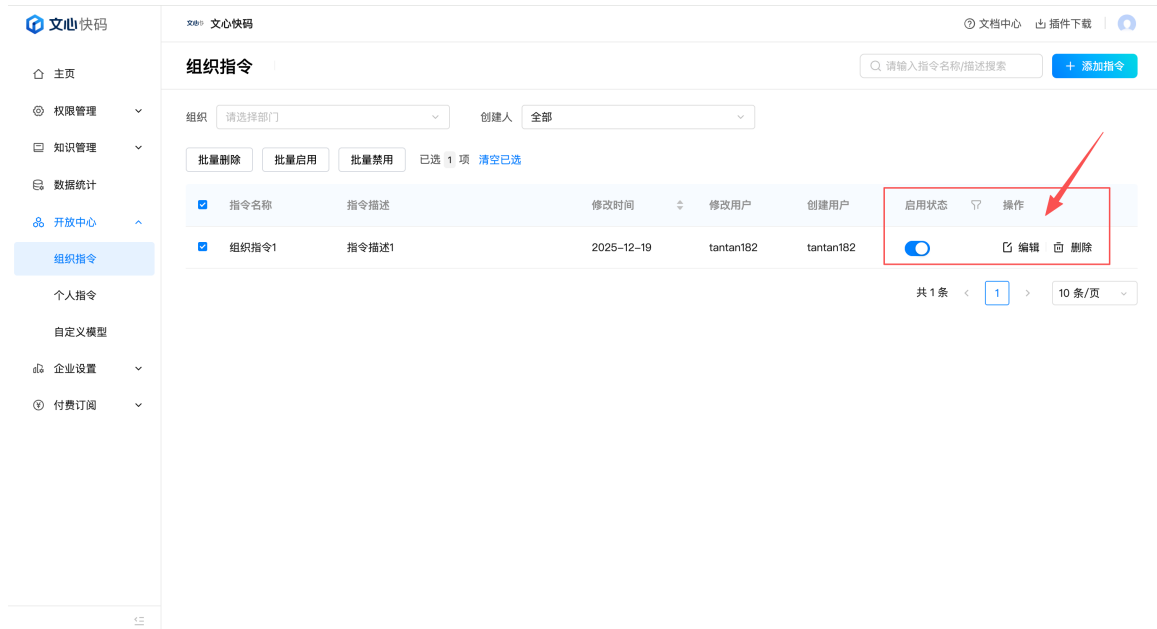
### 3. 批量操作

- 多选指令，可以对指令进行批量操作，包括「批量删除、批量启用、批量禁用」。



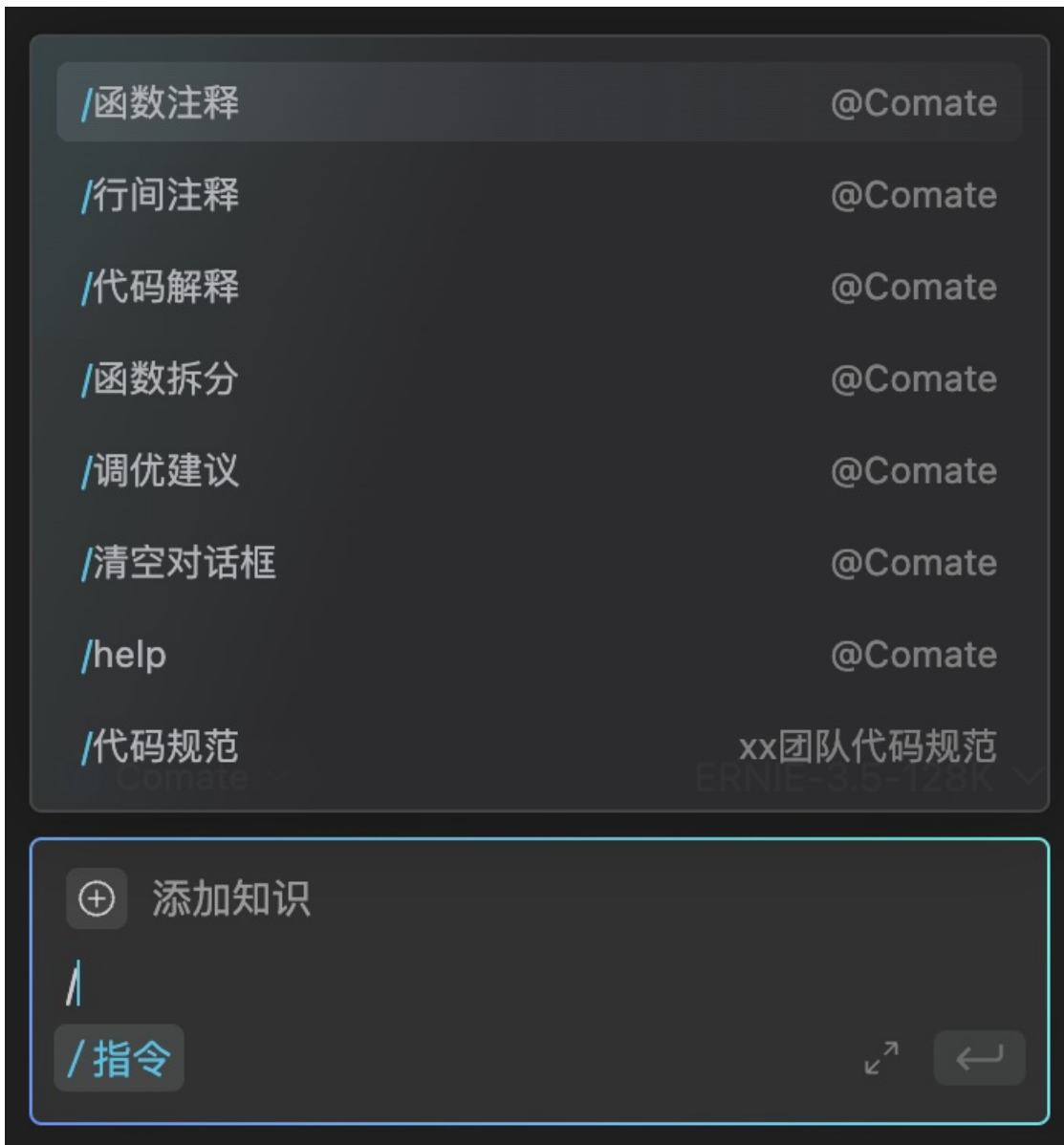
### 4. 指令操作

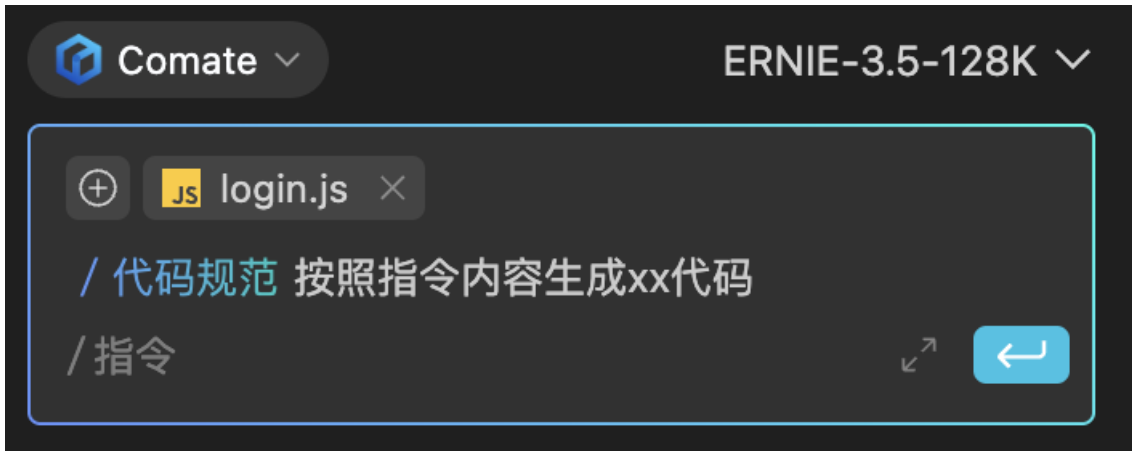
- 可以对任意指令进行「启用、禁用、编辑、删除」等操作。删除后不可恢复，请谨慎操作。



### 5.指令使用

- 指令启用或修改后，预计约5分钟内容在文心快码Comate插件生效，企业内用户可以通过输入“/”使用自定义指令。
- 用户可以添加对话上下文，或是在自定义指令后面输入提问内容，模型会自动参考相关内容并进行回答。





### 自定义模型

自定义模型页面旨在帮助企业管理当前已添加的所有模型资源。1.查看模型 支持查看当前企业已经添加的模型。

模型名称	类型	权限开通部门	模式	开关	操作
gpt-4.1-test	自定义	LDAP总裁办公室、测试2、+3	NORMAL	<input checked="" type="checkbox"/>	编辑   删除
qwen3-coder-flash	自定义	LDAP总裁办公室、测试2、+3	NORMAL	<input checked="" type="checkbox"/>	编辑   删除

2.添加模型 支持添加模型相关基本信息和参数信息，添加企业所需模型。支持按照部门进行可见性配置。

**添加模型**

**基础信息**

- \* 供应商: OpenAI
- \* 域名: https://api.openai.com/v1
- \* 路径: /chat/completions
- \* 模型 ID: gpt-4.1
- \* 名称: gpt-4.1 (7 / 50)
- \* API Key: 请输入 API Key
- \* 最大输入 Tokens: 模型可接受的最大输入 Tokens, 如120000
- \* 最大输出 Tokens: 模型可输出的最大 Tokens, 如8192
- \* 可见部门: 请选择可见部门

< 添加模型

**基础信息**

\* 供应商

\* 域名

\* 路径

\* 模型 ID

\* 名称  7 / 50

\* API Key

\* 最大输入 Tokens

\* 最大输出 Tokens

\* 可见部门

### 3.模型

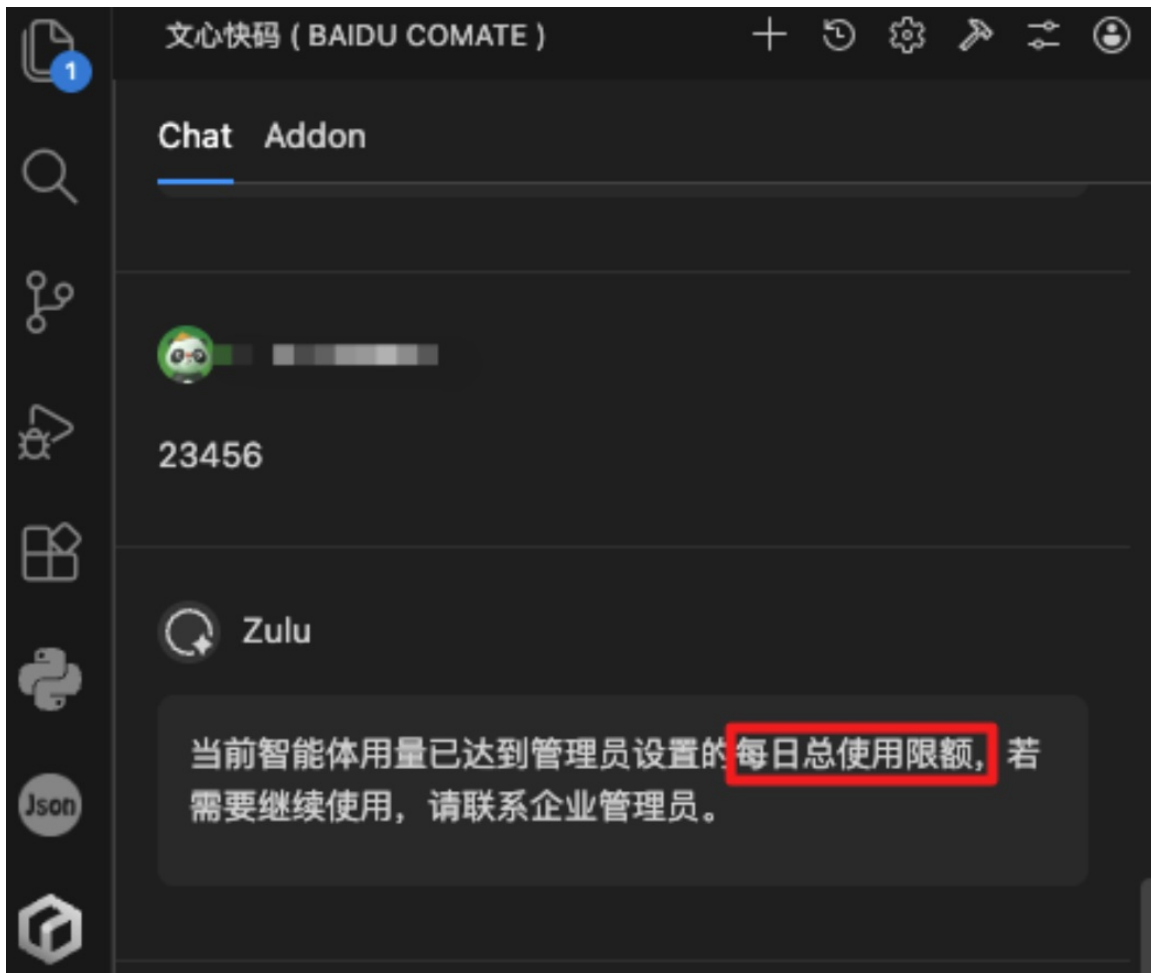
**操作** 支持管理员对模型进行开关，支持编辑历史信息，删除已有模型。

**4.模型使用** 支持在插件端选择模型进行使用。

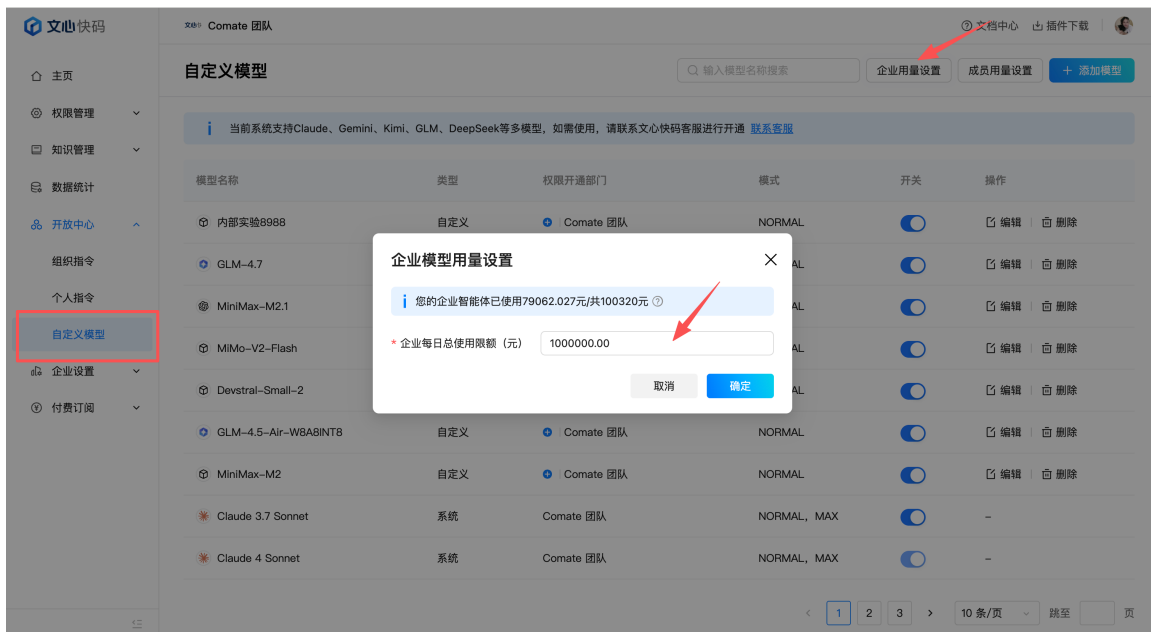
### 🔗 用量限额

**1.企业用量设置** 在开放中心的自定义模型管理页面，管理员可为所属企业配置每日消耗总额度上限。该额度为企业内所有成员账号共享，当企业当日总用量超出设定上限，全体成员账号将自动暂停产品功能的使用权限，直至管理员提高限额/次日额度重置后恢复使用。

**超限提示：**当用户消耗额度达到账号限制时，系统将自动弹出提示并暂停账号服务，直至管理员提高限额/次日额度重置后恢复使用。



**配置方式：**开放中心 管理员可在开放中心的自定义模型管理页面，点击【企业用量设置】按钮，为整个企业配置每日消耗总限额。输入具体额度数值后，点击确认即可生效。



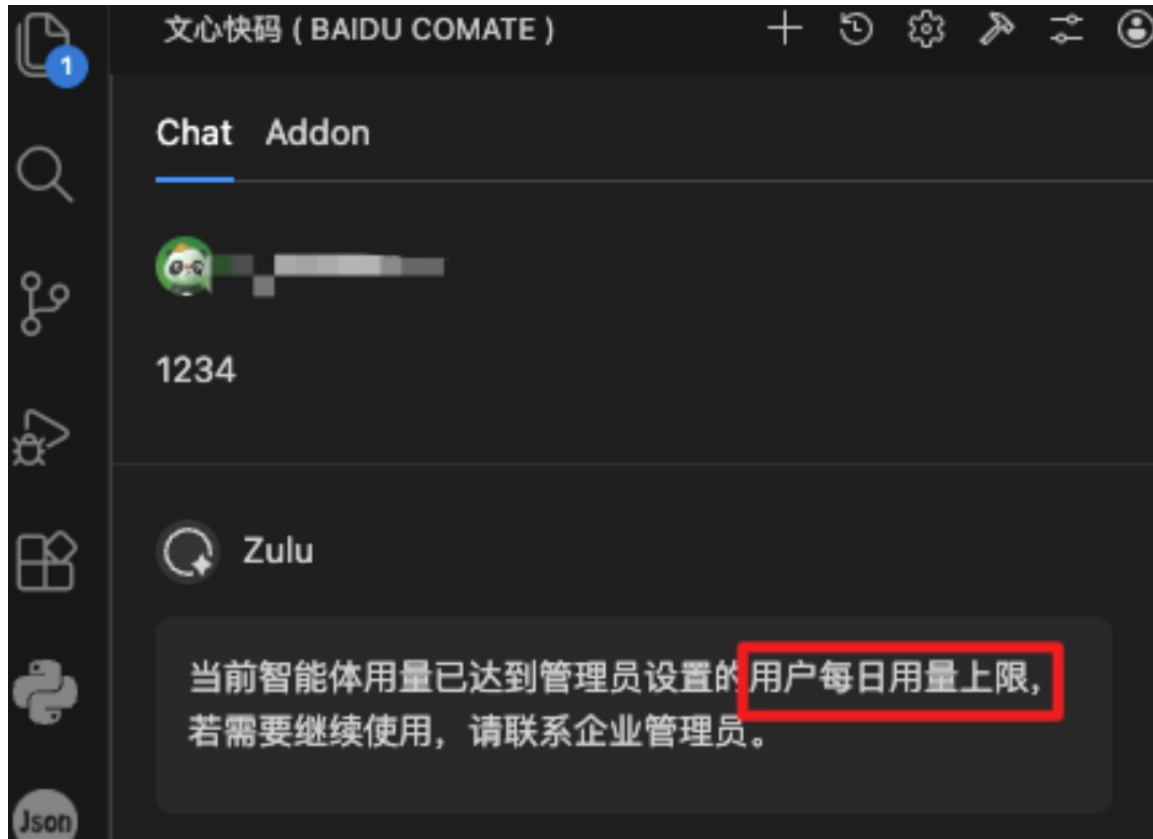
**2.成员用量设置** 系统支持按【成员账号】设置每日模型用量额度的方式进行组织预算的管理。每日额度指单账号每日可消耗的请求金额上限，超出限额后将无法继续使用。

**状态说明：**

- 已限额（具体数值）：成员每日使用金额不超过设定值
- 未限额：成员每日使用不受金额限制

**超限提示：**当用户消耗额度达到账号限制时，系统将自动弹出提示并暂停账号服务，直至管理员提高限额/次日额度重置后恢

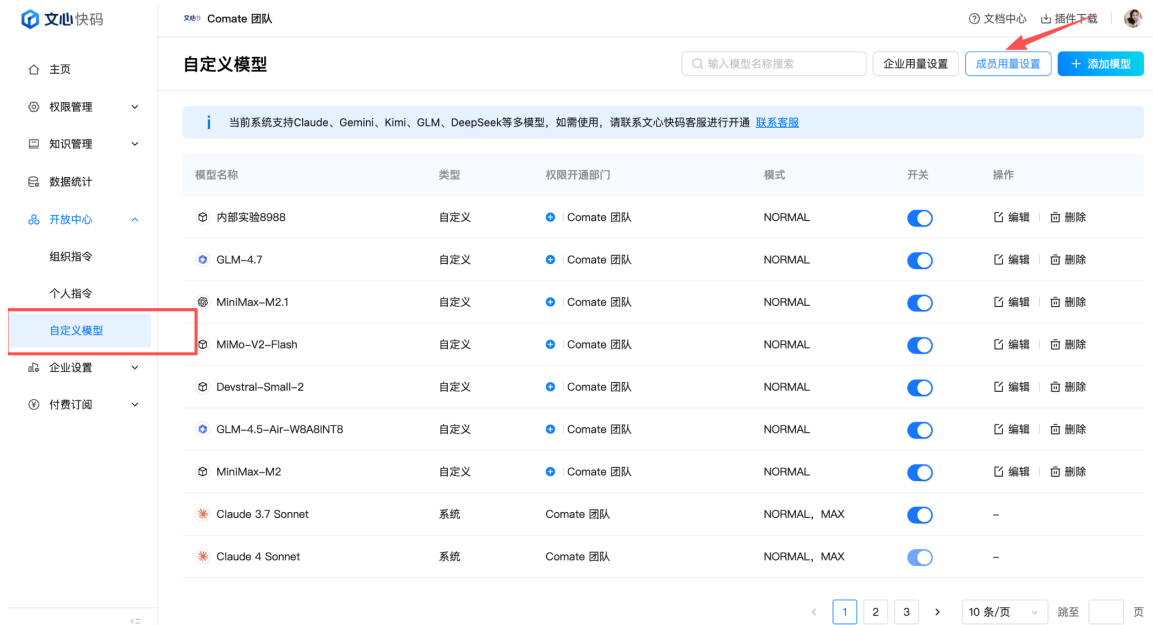
复使用。

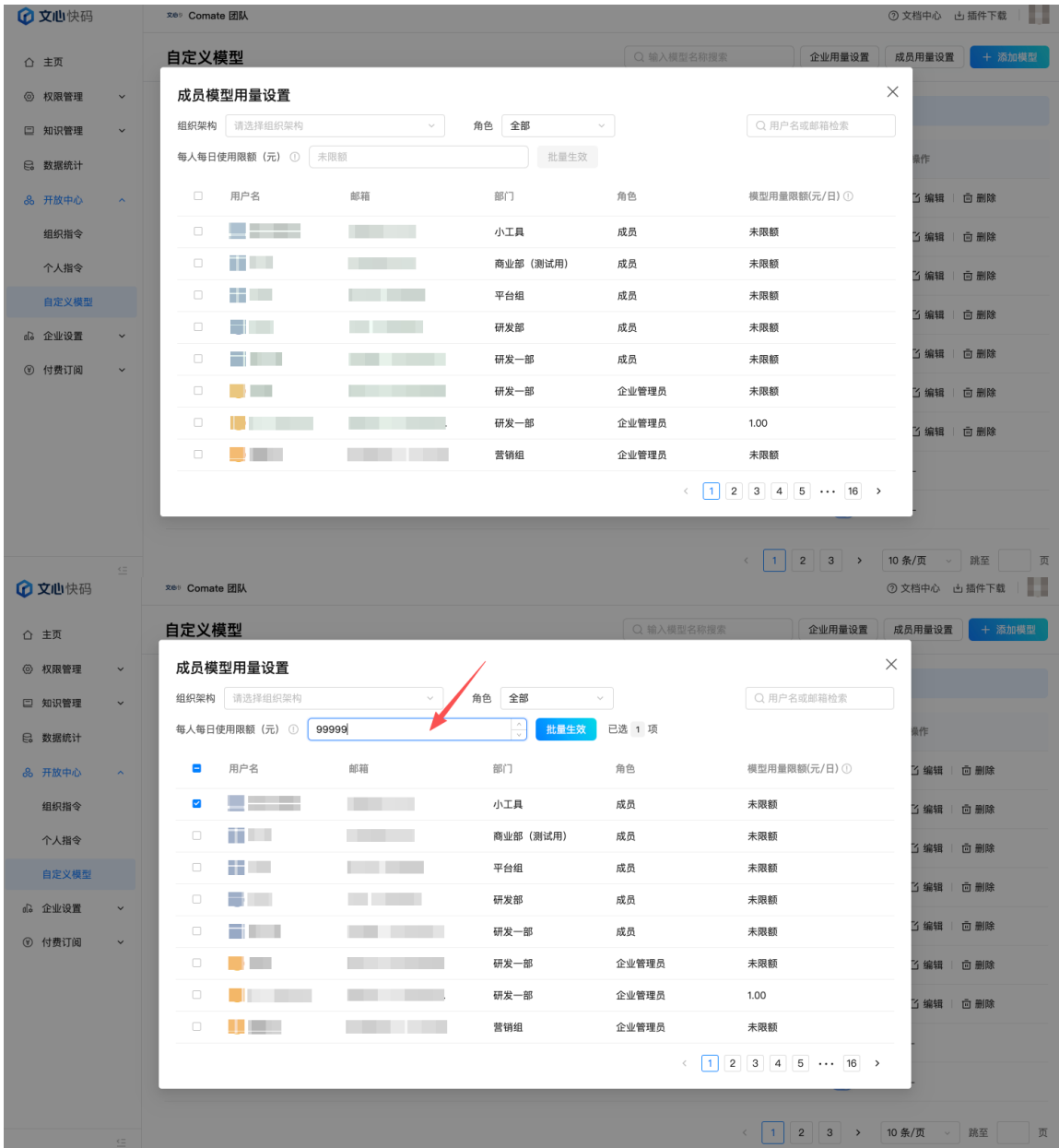


配置方式1：开放中心 登录开放中心，进入【自定义模型】页面，点击【成员用量设置】按钮进行配置。

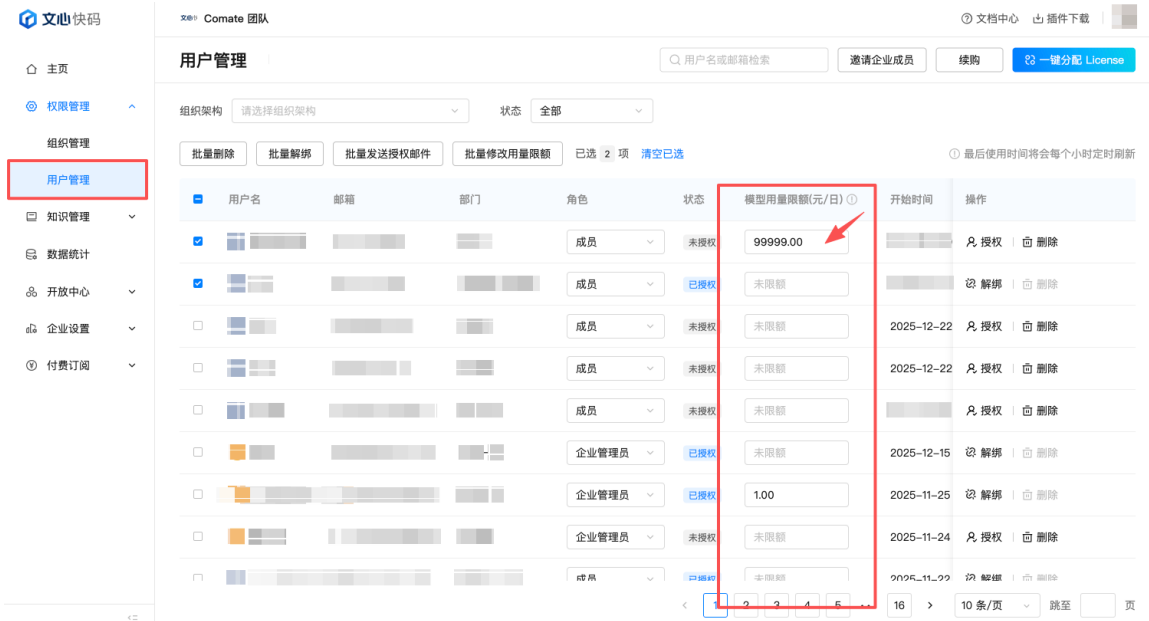
- 在成员用量设置页面，可按组织架构、账号角色或用户名筛选需设置的成员。
- 选定成员后，在“限额额度”栏输入具体金额，点击【批量生效】即可完成设置。

注意：如不填写额度直接生效，该成员将标记为【未限额】，可无限制使用模型。





**配置方式2: 用户管理-单用户修改** 进入【用户管理】页面，在用户列表中找到对应账号，用户双击“模型用量限额”字段输入框直接修改数值后退出后将自动保存。

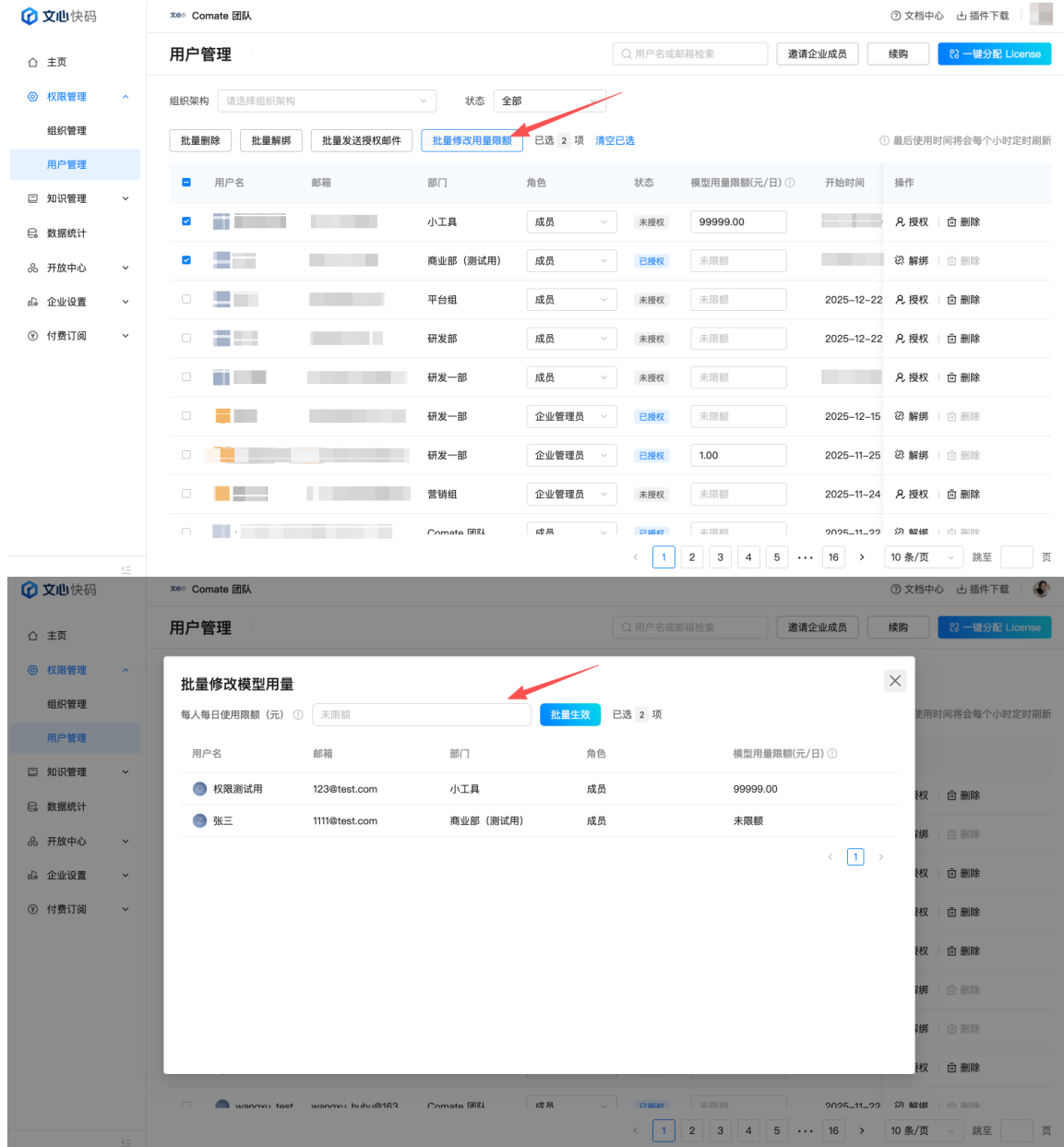


**配置方式3: 用户管理-批量修改**

- 在【用户管理】页面勾选需修改的账号，点击【批量修改用量限额】。

- 在弹出窗口中输入统一限额值，确认后生效。

注意：批量修改时若不输入额度直接生效，所选账号均将设为【无限额】状态。

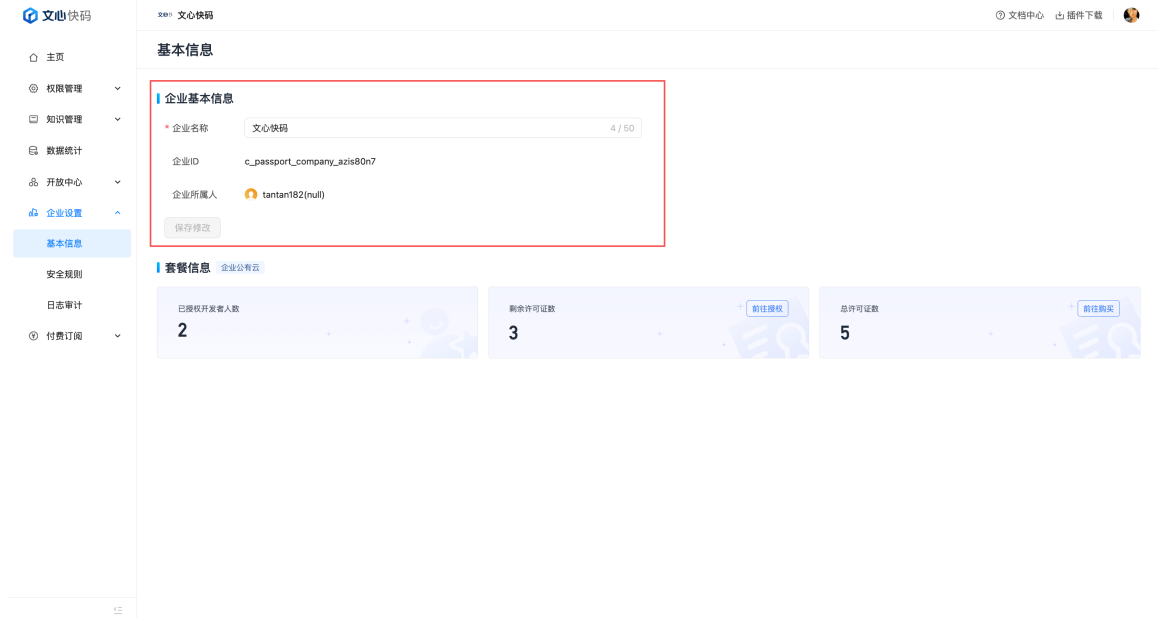


## 企业设置

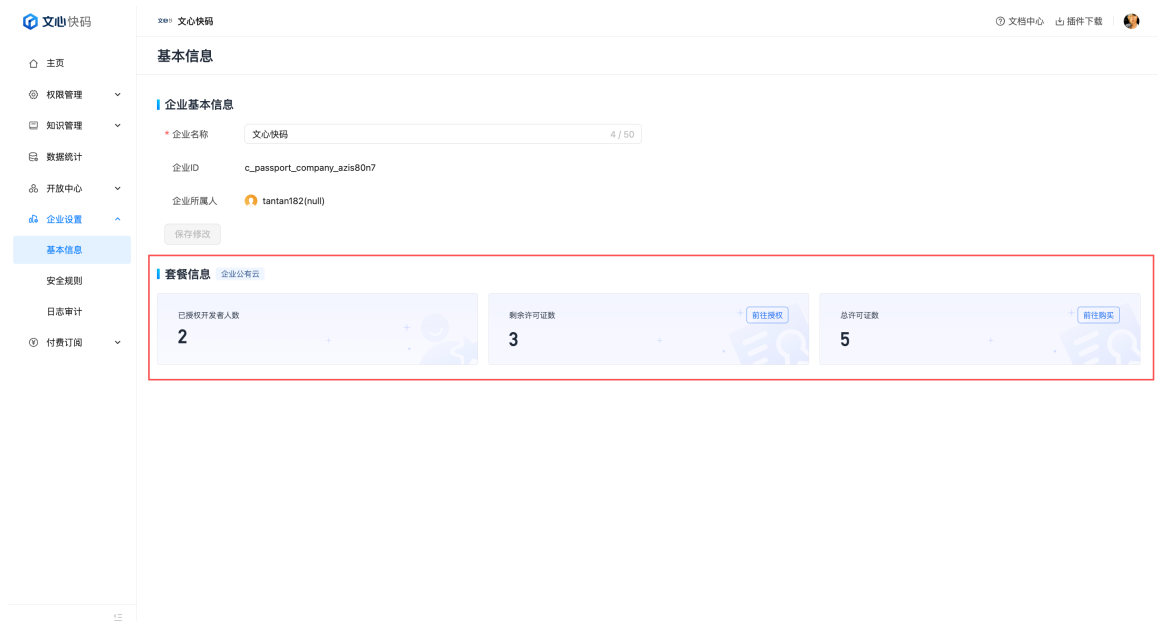
### 🔗 基本信息

企业基本信息页面主要用于展示和管理企业的核心信息与资源配额。

- 1.企业基本信息 展示了企业名称、企业ID、企业所属人等基本信息，企业管理员可以对企业名称进行修改并保存

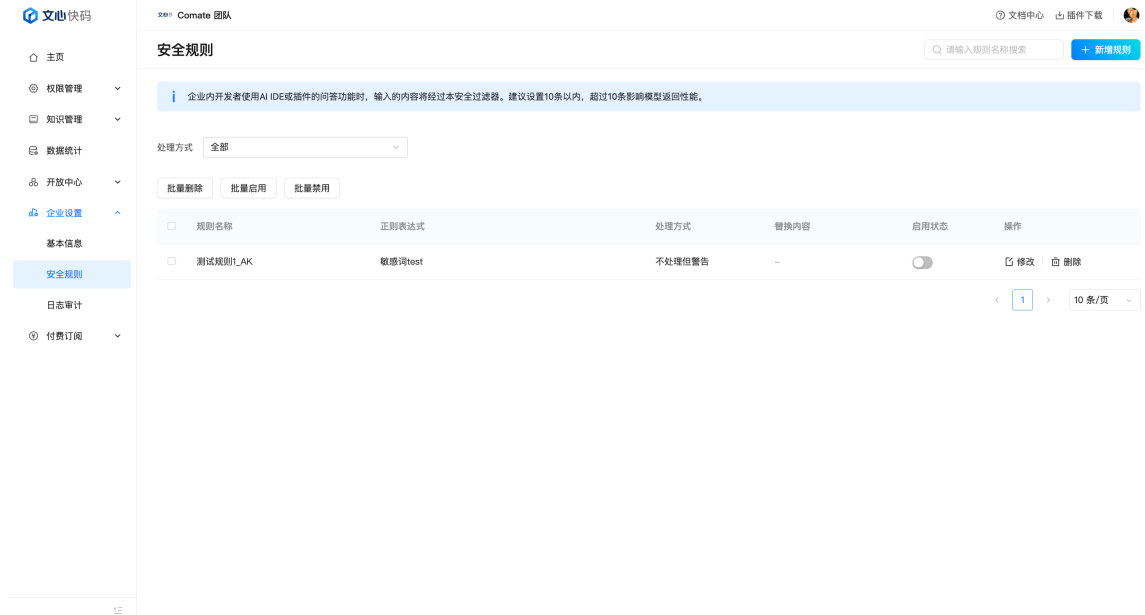


信息 展示了企业总license数、剩余license数以及已授权人数等信息



### 安全规则

企业管理员可在安全规则页面查看及管理全部安全规则。企业内开发者使用 AI IDE 或插件的问答功能时，输入内容会自动经过本页面的安全过滤器检测，以此保障内容的合规性与安全性。

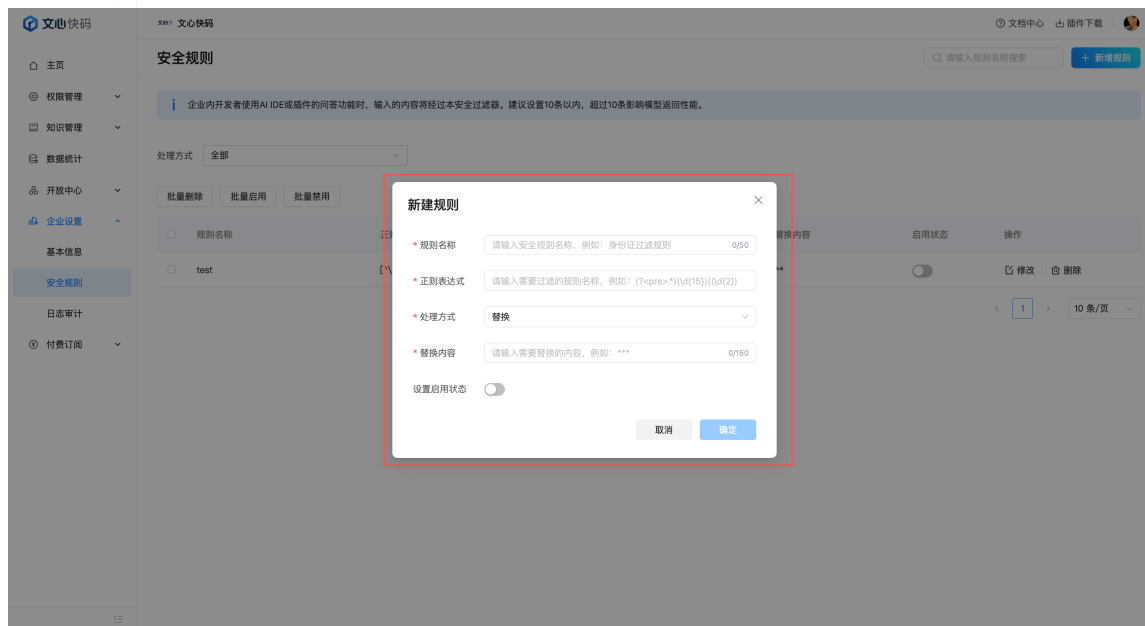


1.新增

安全规则 点击新增规则可以进入详情页面，填入规则名称、正则表达式、处理方式（拦截、替换、不处理但警告）、替换内容，并设置启用状态



2.批量

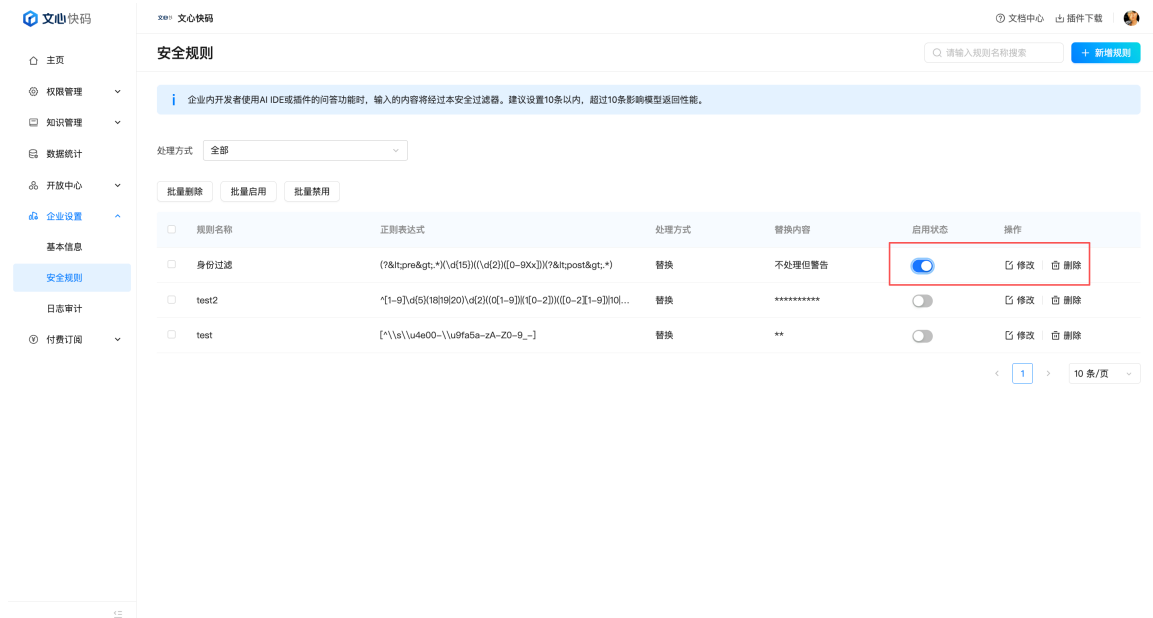


操作 多选安全规则，可以对规则进行批量操作，包括「批量删除、批量启用、批量禁用」。



### 3.规则

操作 可以对任意规则进行「启用、禁用、编辑、删除」操作。删除后不可恢复，请谨慎操作。



### 🔗 日志审计

日志审计页面旨在帮助企业排查信息安全问题、追溯操作源头，日志审计表单内展示了企业用户在管理后台中的全部操作记录。企业管理员可以查看企业用户的操作时间、邮箱、操作类型以及具体操作内容。

- 可以通过筛选时间段和操作类型，快捷筛选出所需的日志内容（数据默认保留半年，仅可筛选近半年内的时间范围）。
- 选择日期点击下载日志后，即可下载当前时间段的操作日志。

文心快码 2025-12-19 文档中心 插件下载

日志审计

操作时间: 2025-12-19 操作类型: 全部操作类型

操作时间	操作人	操作人权限	操作类型	操作内容
2025-12-19 15:15	管理员2 (huhuangteng@baidu.com)	企业管理员	安全规则管理	启用以下安全规则: 身份过滤
2025-12-19 15:12	管理员2 (huhuangteng@baidu.com)	企业管理员	安全规则管理	新增安全规则: 身份过滤
2025-12-19 15:11	管理员2 (huhuangteng@baidu.com)	企业管理员	安全规则管理	新增安全规则: test2
2025-12-19 15:08	管理员2 (huhuangteng@baidu.com)	企业管理员	安全规则管理	修改安全规则: test
2025-12-19 15:07	管理员2 (huhuangteng@baidu.com)	企业管理员	安全规则管理	新增安全规则: test
2025-12-19 14:20	tantan182 (null)	企业管理员	指令管理	新增个人指令: 个人指令1
2025-12-19 14:20	tantan182 (null)	企业管理员	指令管理	修改组织指令: 组织指令1
2025-12-19 14:20	tantan182 (null)	企业管理员	指令管理	修改组织指令: 组织指令1
2025-12-19 14:04	管理员2 (huhuangteng@baidu.com)	企业管理员	用户管理	管理员2(huhuangteng@baidu.com) 通过邀请链接增加...
2025-12-19 13:48	tantan182 (null)	企业管理员	用户管理	一键授权用户License

10条/页 跳至 页

## 付费订阅

### 付费服务使用说明

付费服务页面是企业管理员管理企业资源的核心入口。在这里，您可以购买企业专业版 License（席位）、补充智能体请求代金券，并对已有的 License 进行续费和授权

**1.新购：企业专业版 License** 此页面用于采购企业成员的使用许可（License）。购买 License 是企业成员正常使用平台功能的基础。

文心快码 2025-12-19 文档中心 插件下载

付费服务 新购 续购

企业专业版License 智能体请求包

专业版·1年 ¥1500 专业版·3个月 ¥450 专业版·1个月 ¥150

每人每月赠送智能体消耗代金券，企业范围内容共用，智能体按照不同模型消耗的Tokens数进行金额扣减。

有效期至 2026-12-19 | 购买数量 1

加购权益

150元智能体请求代金券 ¥150 33元智能体请求代金券 ¥33 18元智能体请求代金券 ¥18

同意并接受 用户服务协议、隐私协议

订单金额 ¥1500 立即购买

### 操作步骤：

**步骤1：选择套餐时长：** 在页面顶部选择所需的 License 有效期。我们提供灵活的方案：

专业版·1年（¥1500）：推荐方案，适合长期使用的核心团队。

专业版·3个月（¥450）：适合季度项目组。

专业版·1个月（¥150）：适合短期试用或临时人员。

注：所有套餐均包含每月赠送的智能体消耗代金券。

**步骤2：设置购买数量：** 在右侧调整“购买数量”，系统会自动计算有效期的截止时间（例如：有效期至 2026-12-19）。

**步骤3：加购权益（可选）：** 如果企业预计大模型调用量较大，可在下方的“加购权益”区，直接勾选额外需要的“智能体请求代金券”。

金券” (¥150/¥33/¥18)。

**步骤4：支付订单：** 确认右下角总金额，勾选协议后点击“立即购买”。

**2.新购：智能体请求包** 当企业 License 充足，但大模型调用额度 (Token) 不足时，可单独在此页面购买“智能体请求包”。



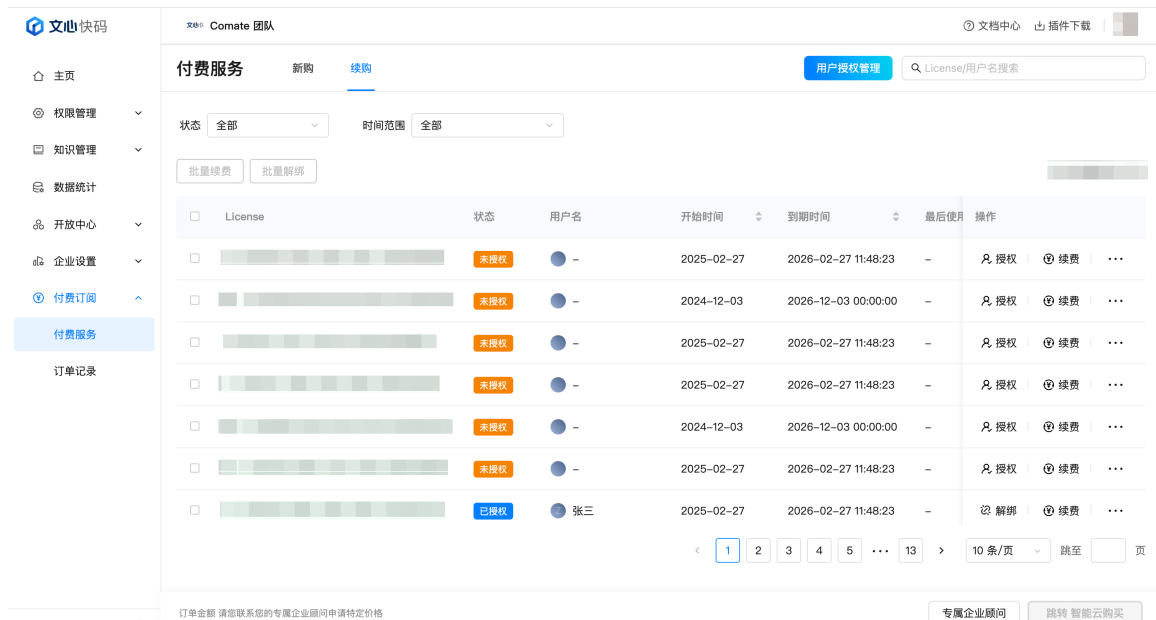
**功能特点：** 全企业共用：购买的代金券进入企业资金池，所有企业成员在调用智能体时共享使用。

**永久有效：** 与 License 每月赠送的额度不同，单独购买的请求包没有时间限制，用完为止，不会过期。

**购买选项：** 提供 ¥150、¥33、¥18 三种面额的代金券。

选择所需数量后，点击右下角“立即购买”即可实时充值到企业账户。

**3.续购与 License 管理** 此页面用于管理企业已购买的 License 资产，支持查看状态、分配人员以及进行续费操作。



**状态监控：**

列表展示了每一条 License 的详细信息：License ID：唯一识别码。

**状态：** 未授权：该席位闲置，未分配给具体人员。 已授权：该席位正在被成员（如“成员-张三”）使用。

**时间信息：** 清晰展示“开始时间”与“到期时间”，方便管理员提前规划续费。

**管理操作**

在列表右侧的“操作”栏，您可以进行以下管理：

- 授权/解绑：

点击“授权”：将空闲 License 分配给指定企业成员。

点击“解绑”：收回成员的 License，收回后变为“未授权”状态，可重新分配。

- 续费：

单条续费：点击对应 License 后的“续费”按钮。

批量续费：勾选多条 License，点击列表上方的“批量续费”按钮，统一延长有效期。

## 🔗 订单记录

订单记录页面专为企业的财务对账与订单记录追溯，集中呈现所有历史消费明细，确保支出透明、可审计。支持通过组合条件快速定位目标订单，提升查询效率。每笔交易均清晰展示以下信息，确保全程可追溯。

### 功能详述：

- 多维筛选查询：支持通过以下条件快速定位订单：
- 订单类型：筛选是“新购”还是“续费”。
- 订单状态：查看“支付成功”或“待支付”的订单。
- 关键词搜索：直接输入订单号或操作人姓名（如“admin”）进行精确查找。
- 时间范围：筛选特定月份或季度的支出记录。
- 记录明细：列表详细展示了每笔交易的购买内容、金额、创建/付费时间以及操作人，确保每一笔企业支出都有据可查。
- 快捷入口：页面右上角提供【用户授权管理】入口，可一键跳转至权限/用户管理界面，便于在查看订单后直接进行席位分配或权限调整。

# 企业版

## 开通企业版

### 🔗 概述

Baidu Comate 现已推出企业版免费试用服务，诚邀您体验。

通过企业版，您可以深入了解并应用企业管理、数据看板等核心功能。这些功能为管理者提供高效的权限管理解决方案。

此外，企业版还引了高效的私域知识管理功能，旨在助力企业内部知识的有效利用，从而大幅提升开发效率。

### 🔗 访问入口

您可通过访问Baidu Comate官网首页，点击企业免费试用来发起申请。



## 如何注册企业

1. 点击【企业免费试用】后，请您登录您的百度账号。



2. 登录后进入企业申请表单填写页，您需要填写您的姓名、企业名称、您的企业研发组织规模、您的手机号和企业邮箱，点击【立即开通】即可。



3. 开通后您将进入企业主页，您可以前往用户管理为您的企业用户授权账号，并查看数据看板。

## 授权管理

### 概述

当您创建完企业后，您可到用户管理页进行发起授权、删除成员、更改成员角色等操作，并可查看企业内部成员的姓名、邮箱、部门、角色、使用时间等信息。

### 授权企业成员

1. 您可以通过点击【邀请企业成员】来邀请企业成员，您可以修改符合您的分享场景的文案模板，进行复制。

2. 分享给企业成员后，企业成员可直接点击链接加入您的企业。

## 批量管理

您可以在操作中授权/删除对应成员、更改成员角色，也可通过多选企业成员批量修改。

The screenshot shows the '用户管理' (User Management) page in the Baidu Comate interface. The page title is '文心快码 Baidu Comate'. The navigation bar includes '首页', '案例', '价格', '资讯', '文档', and '在线问答'. The user management section has tabs for '全部用户', '未授权', and '已授权'. A search bar is present with the placeholder '请输入用户名或邮箱搜索'. There are buttons for '邀请企业成员' and '批量导入用户'. A dropdown menu is open over the table, showing options: '批量管理', '批量更改角色', '批量删除用户', '批量授权 License', and '批量解绑 License'. The table has columns: '邮箱', '部门', '角色', '状态', '最后使用时间', and '操作'. The data rows show users with names like 'yue 3', 'yue 2', 'yue 1', and 'yue', all with the status '未授权'. The bottom right shows '共 6 个' and '10 条/页'.

## 企业成员使用

### 激活流程

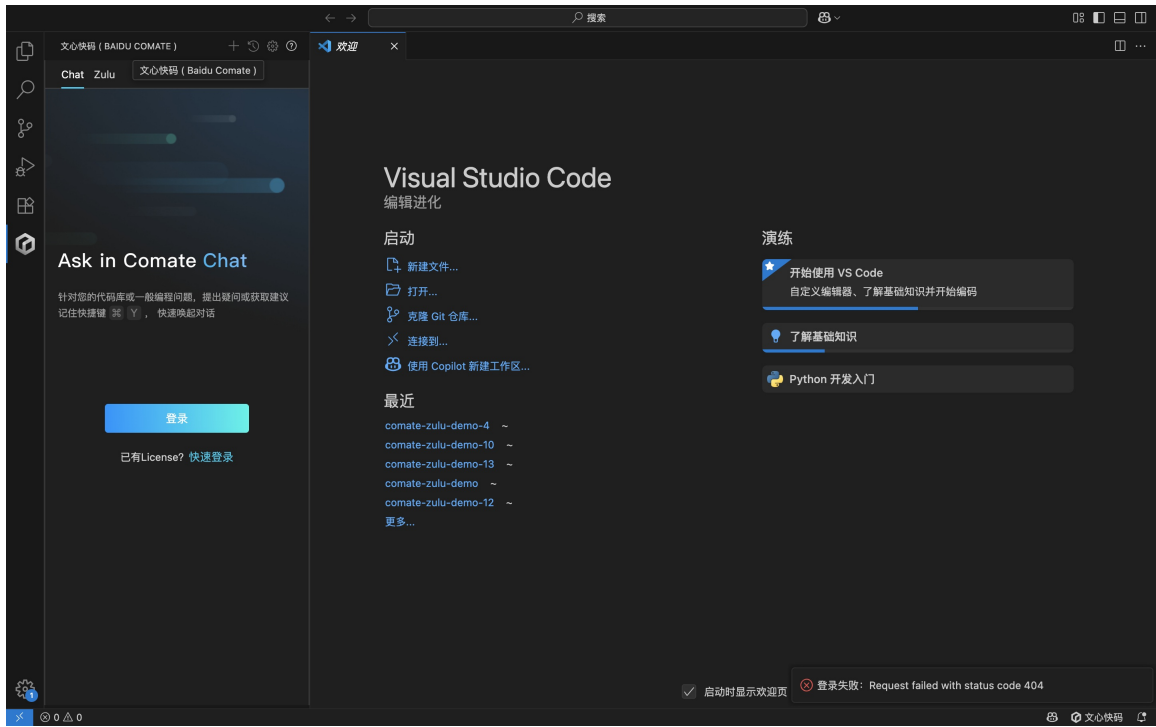
当您作为企业员工时，您收到链接后请点击链接，并登录百度账号。

为了保障您的企业管理员能更好的识别到您的身份，请您认真填写您的姓名和邮箱，并点击【确认激活】。

The screenshot shows the activation form for enterprise members. The title is '文心快码 (Baidu Comate) 企业版智能代码助手'. The text says: '您的企业文心快码(百度Comate)邀请您使用文心快码 (Baidu Comate) 您的信息将用来帮助企业管理者完成您的身份识别，请认真填写'. The form has fields for: '企业名称' (Baidu Comate), '部门名称' (Baidu Comate), '您的角色' (成员), '用户名' (请输入您的用户名), and '您的邮箱' (请填写您的邮箱). A '确认激活' button is at the bottom.

### 使用流程

当您在IDE中下载Baidu Comate插件后，请您点击【登录】，并跳转到浏览器选择企业账号进行登录，登录完成后跳转回IDE内即可使用。



### 请选择您的账号登录

- 个人账户
- 文心快码(百度Comate)

确认

登录账号: 悦悦错了嘛

登录设备: liyue liyuedeMacBook-Pro.local (Mac OS x64 23.5.0)

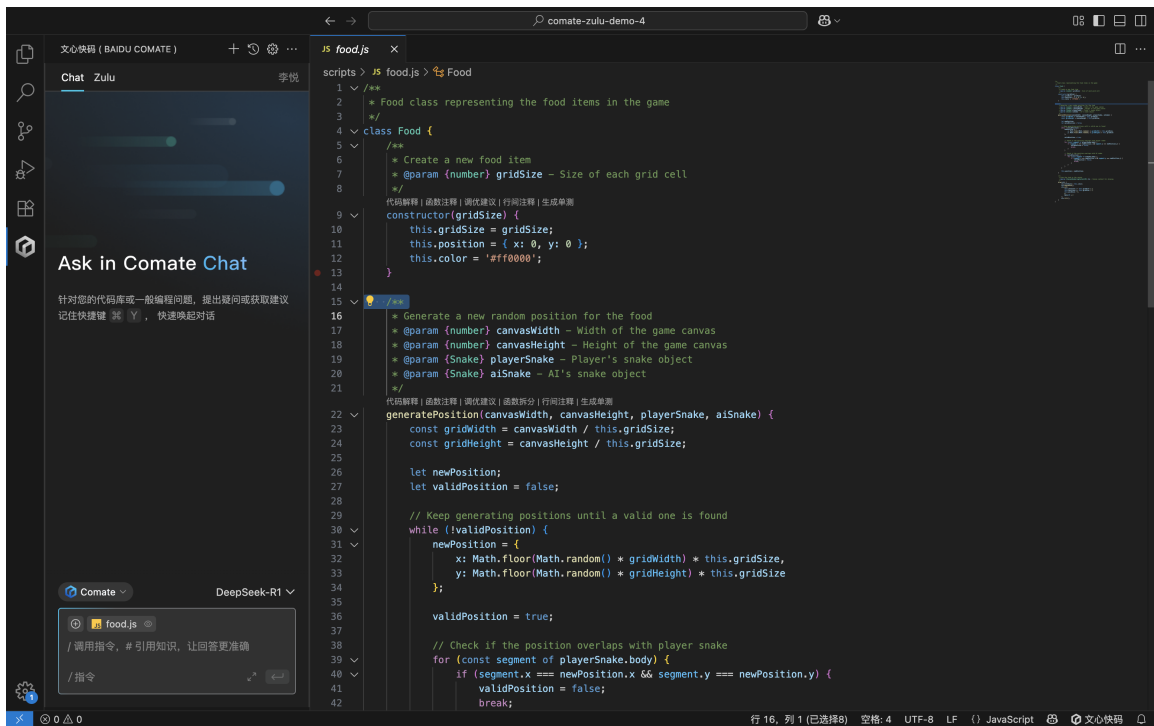
## Baidu Comate



### 登录成功

可返回 IDE 客户端立即体验文心快码 (Baidu Comate)

返回 VS Code



## 混合云帮助手册

### 什么是混合云？

混合云是一种云计算环境，它结合了私有云（私有化部署或本地数据中心）和公共云（如百度智能云）的优势。通过这种组合，企业可以在本地数据中心和公共云服务之间自由转移数据和应用程序，从而在安全性、灵活性、成本效益等方面实现最佳平衡。

在Baidu Comate中使用混合云方案，可以带来以下几方面的优势：

1. **数据安全和隐私：**代码助手可以在私有云中处理和存储代码、知识库等敏感数据，确保数据隐私和安全，同时利用公共云的计算能力进行复杂的代码分析和机器学习模型训练。
2. **高效资源利用：**混合云架构允许智能代码助手在需要时动态扩展计算资源，利用公共云资源进行计算密集型推理任务，而日常的小规模操作则在本地完成。
3. **实时更新和模型优化：**公共云可以提供最新的机器学习模型和算法更新，智能代码助手可以通过混合云架构及时获取这些更新并应用于本地，从而保持工具的先进性和高效性。
4. **开发者协作：**通过混合云，开发团队可以在不同地理位置和网络环境下协同工作，共享代码和分析结果，提高开发效率和协作水平。

### 混合云适用哪些企业？

混合云架构的智能代码助手适合以下类型的客户：

1. **大型企业：**具有复杂的IT基础设施和严格的数据安全需求，混合云可以帮助他们平衡安全性和成本效益。
2. **金融机构：**对数据隐私和合规性要求极高，混合云能够提供安全的本地存储和公共云的计算能力。
3. **科技公司：**需要频繁进行机器学习模型训练和代码分析，混合云架构可以提供灵活的计算资源和最新的技术支持。

### 如何使用企业混合云版？

#### 1. 注册企业

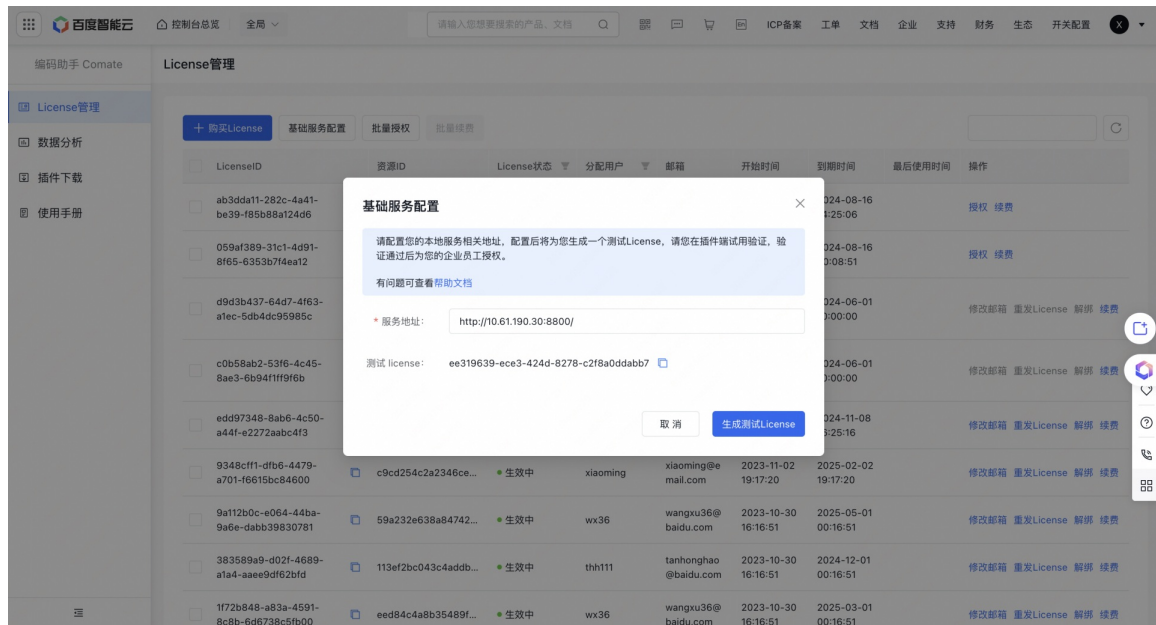
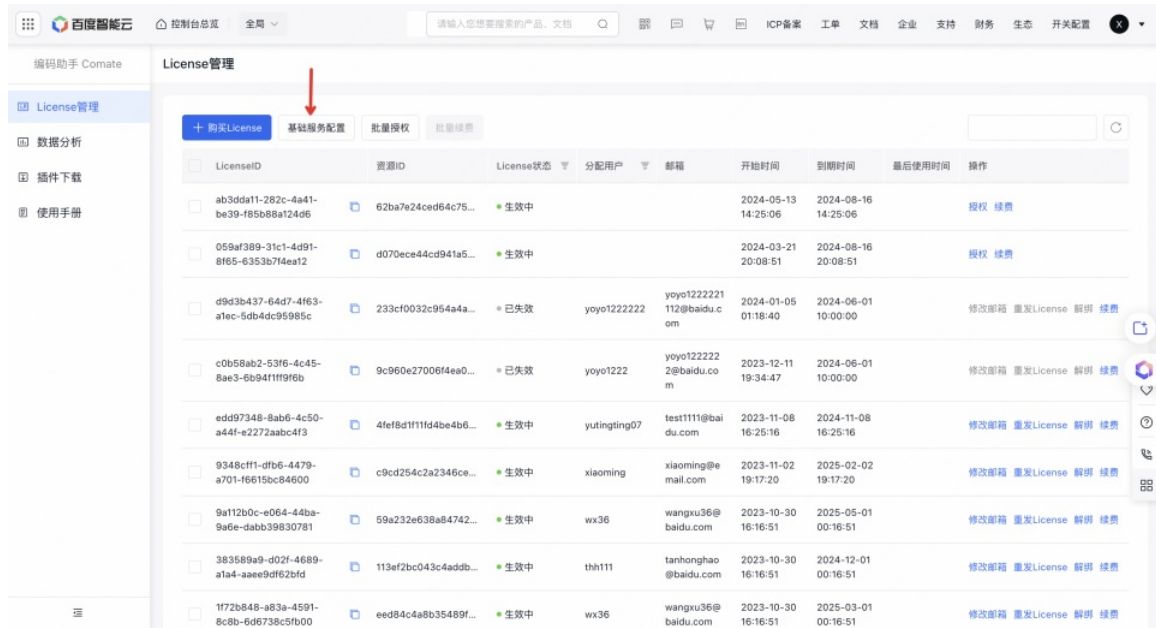
您的专属销售顾问将协助您开通百度智能云企业版账号，请您按照步骤完成实名认证。

#### 2. 开通混合云账号

当您完成企业注册，并决定以混合云的方式使用Baidu Comate时，请您联系您的专属技术支持顾问，将您的账号类型变更为【混合云版】。

### 3. 配置服务地址

当您进入百度智能云控制台后，请您先配置基础服务，请点击License管理中的【基础服务配置】，填写您的服务地址后，点击【生成测试License】后，在页面中将为您生成一个测试License，请您将测试License复制粘贴到IDE中进行测试。

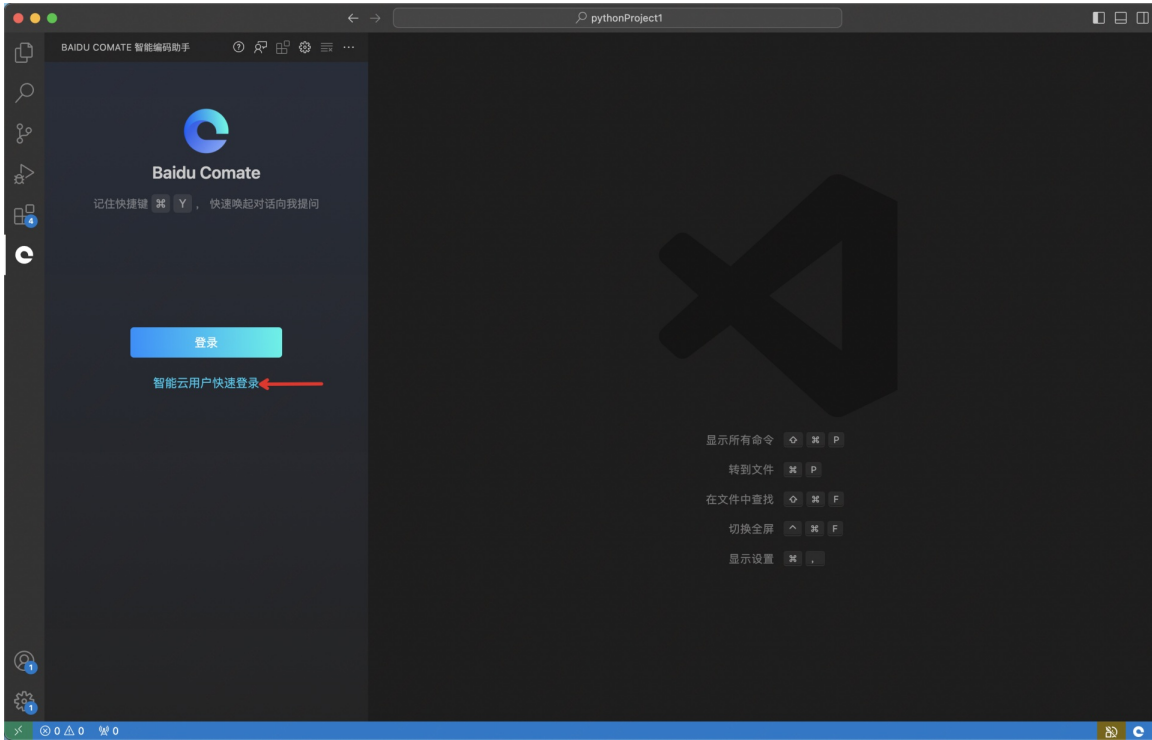


### 4. 测试服务地址

请您在IDE中选择智能云用户登录，并粘贴您的测试License，保存后，我们将为您默认配置您的服务地址。

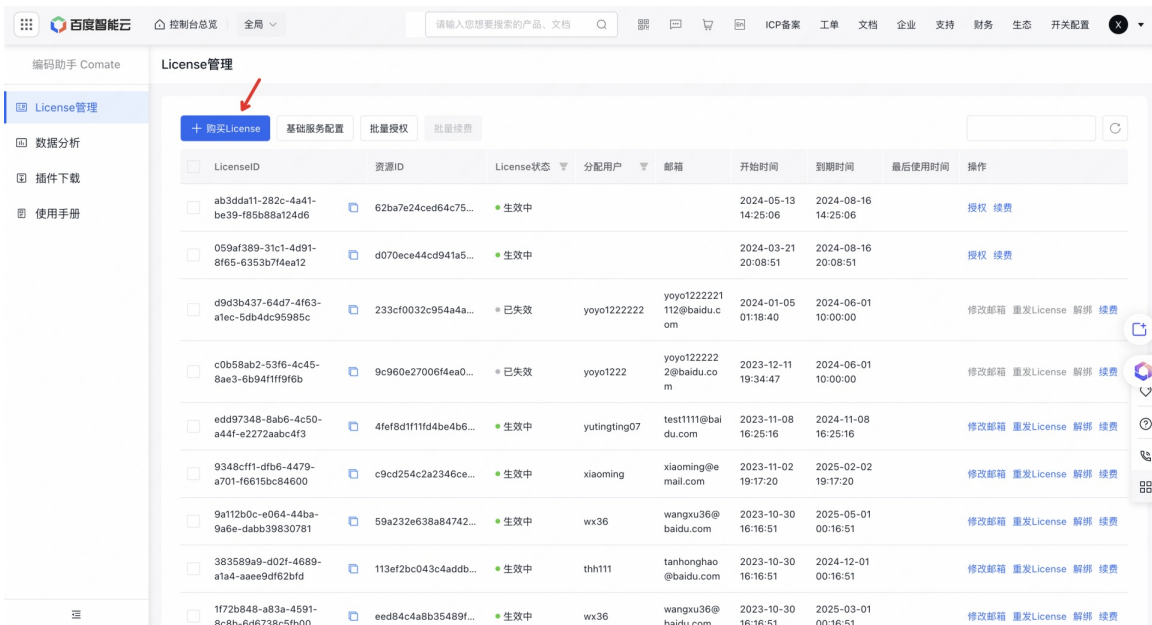
请您通过@AutoWork、/代码库、/知识库等方式测试Autowork、知识增强是否可以正常使用，如不能正常使用，请重新配置您的服务地址；如可以正常使用，请您返回使用。

您的测试License有效期为3天，请您在3天内完成服务地址配置。



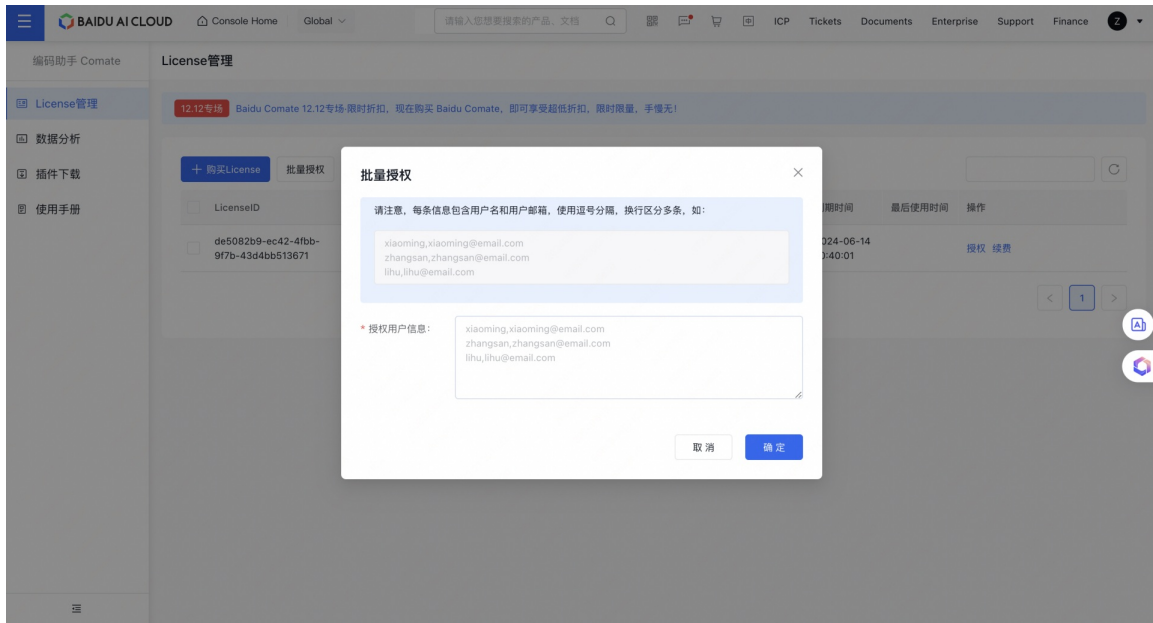
## 5. 购买License

您可通过点击【购买License】来批量购买License，也可联系您的专属销售顾问进行购买。



## 6. 配置License

当您购买License后，您可通过点击【批量授权】，填写您的企业员工邮箱进行批量授权，我们将会向对应的邮箱发放您购买的License，请注意查收。



## 7. 企业成员使用

企业成员在登录时，需要选择【智能云用户快速登录】，并填写收到的License。我们将为您的用户默认配置服务地址，如果用户发现服务地址不可使用或不符合服务环境，可在插件设置-服务配置中，填写正确的服务地址，保存使用即可。

## 代码安全

### 功能介绍

代码安全是 Comate 建设的核心的IDE安全能力，将百度多年累积的安全经验与大模型相结合，实现了代码编写过程中的漏洞自动检测和修复，可帮助用户在编码阶段快速发现并修复安全风险。

- **漏洞扫描**：「Comate 代码安全」支持一键检测代码中的安全漏洞，包括供应链漏洞（如 Log4j2 远程命令执行、Fastjson 远程命令执行等）和通用漏洞（如 SQL 注入、SSRF、RCE等），并给出详细的漏洞说明和修复方案。
- **漏洞修复**：对于扫描出的供应链漏洞「Comate 代码安全」支持进行一键修复，将组件更新到安全版本，帮助工程师低成本解决安全风险；对于 SQL注入、SSRF、RCE 等通用安全漏洞，通过漏洞修复大模型帮助用户一键生成修复代码，用户在采纳后即可替换修复。

### 支持语言

当前支持 Java 语言，对其他语言的支持也在快速迭代中，尽情期待。

### 前置准备

#### 1. 注册企业账号

代码安全功能需要通过企业账号进行试用，您可通过访问Baidu Comate官网首页，点击企业免费开通来发起申请，具体流程请参考[企业开通说明](#)

#### 2. 开启安全配置

在企业管理页面的【配置中心】，启用 Baidu Comate 代码安全并保存更改。



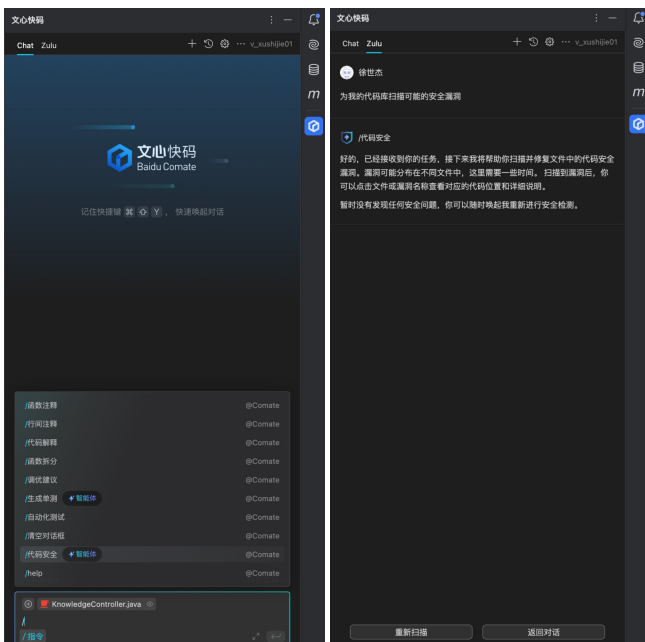
### 3. 登录企业账号

打开IDE，点击登录后，选择企业账号登录。

### 使用方式

【代码安全】在 Comate 可通过 / 调用指令，选择 /代码安全 回车或点击发送即可对当前项目文件发起安全检测（如果无法看到/代码安全 指令请重启 IDE）。

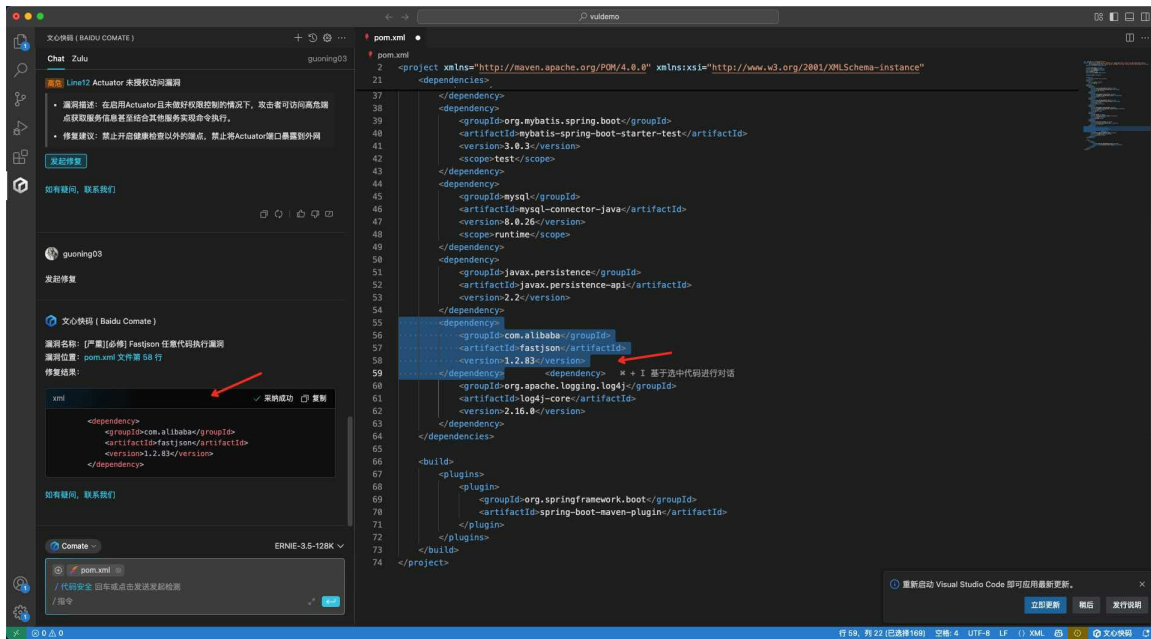
同时您可以通过 /代码安全 + 特定目录的参数形式扫描指定文件夹，实现快速扫描。



扫描完成后可以在对话框看到扫描结果，包括漏洞统计、漏洞详情、修复建议等信息，点击漏洞行数可以直接跳转到漏洞所在行。



点击【发起修复】可以尝试对特定漏洞进行自动修复并返回修复后的代码（无修复按钮的漏洞表示暂不支持自动修复），点击【采纳】按钮可直接将修复后的代码粘贴到原文件所在位置，替换原漏洞代码，实现自动修复。



## 解决方案

文心快码具备代码补全、技术问答、知识增强、智能体协作等多种核心能力，覆盖整编码个环节。为了满足企业数据安全，文心快码支持多种部署形态。

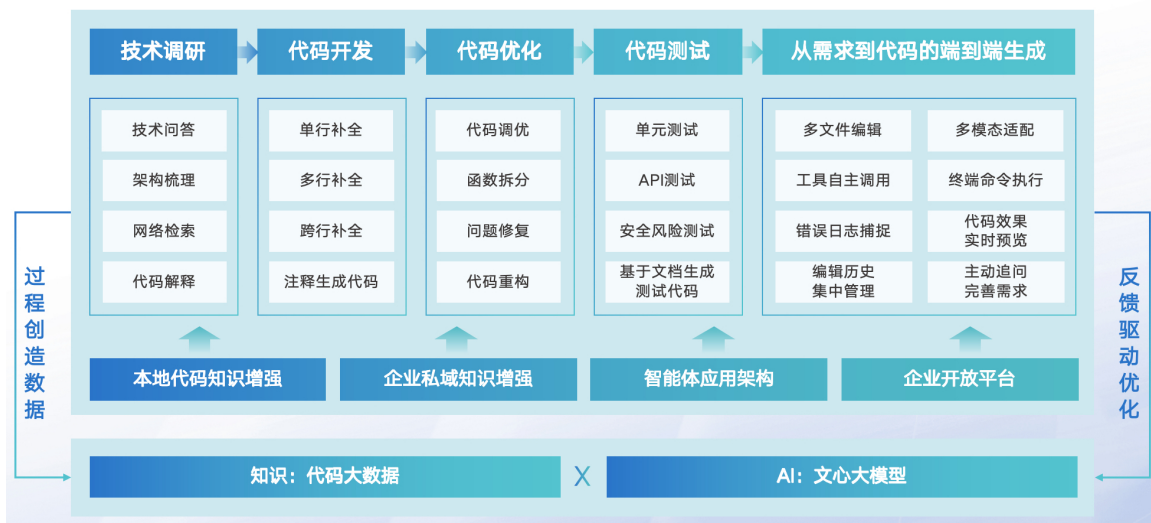
### 1、产品架构

文心快码以插件的形态集成到各类IDE中，为开发者提供全流程的编码辅助，支持代码补全、技术问答，通过私域知识增强来强化代码生成效果，同时提供智能体模式与开发者高效协作。

- 支持的IDE：JetBrains（涵盖IntelliJ IDEA、PyCharm、GoLand、WebStorm、Android Studio等）、VSCode、Visual Studio、

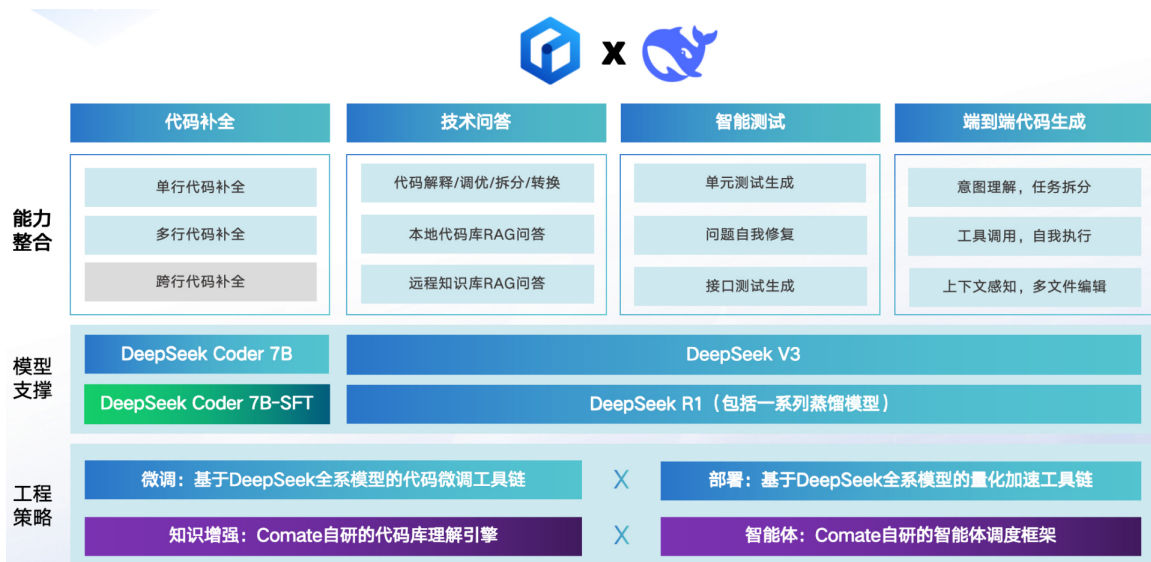
Xcode、Eclipse、DevEco Studio

- 支持的语言：Java、Python、Golang、C++、C、C#、TypeScript、JavaScript、Rust、Ruby、PHP等全部主流语言
- 支持的操作系统：Mac、Windows 7及以上



### 2、整合DeepSeek

文心快码对DeepSeek各类模型进行了全面整合，支持在问答、测试、智能体等各类场景调用DeepSeek能力。

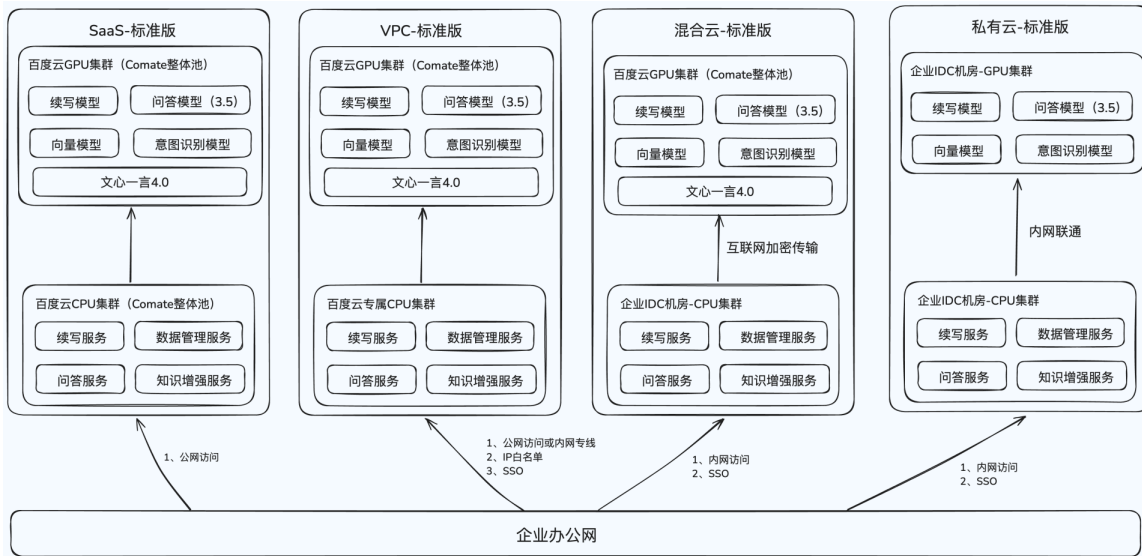


### 3、产品能力清单

分类	功能	描述
代码补全	单行补全	基于上下文依赖分析，在开发过程中换行后自动代码补全单行代码，以及在行间补全部分代码。
	多行补全	基于上下文依赖分析，在开发过程中换行后自动代码补全多行。
	预测编辑	自动修正拼写错误并补全代码，以及预测下一处要编辑的光标位。
	注释生成代码	基于上下文依赖分析，并结合上文注释生成代码。
代码问答	技术问答	对各类技术通用问题进行解答，支持多模型切换，产品内置了文心一言、DeepSeek等模型。
	代码解释	对选定的函数、代码块等进行详细解释，并展示业务逻辑流程图。
	代码注释	对选定的函数、代码块生成注释，包含行间注释、函数头注释
	代码调优	对复杂函数进行逻辑调优，如整合多个if-else语句、增加异常捕获、合法参数校验等。
	函数拆分	针对复杂函数进行细粒度拆分，拆分为多个更简洁的函数，使得各个逻辑解耦。
	生成单元测试	对选中的代码块生成单元测试，支持函数、文件两个级别的生成。
	生成提交信息	基于当前代码变更生成提交信息，用于Commit Message。
	代码问题分析与修复	自动捕捉IDE在调试过程中的报错后分析并修复。
	本地代码库知识增强	检索本地代码库知识，并在技术问答时引用提高代码生成质量。
	网络检索知识增强	检索互联网最新知识，并在技术问答时引用提高代码生成质量。
企业文档知识增强	企业上传本地文档（需求/接口/技术等各类文档），在技术问答时引用。	
Inline Chat	支持在编辑区进行问答，并直接更改代码（仅支持JetBrains和VSCode）。	
智能体 (Zulu)	多文件编辑	基于对上下文的依赖分析和用户需求的理解，同时编辑多个文件。
	多模式适配	支持上传图片，解析图片内容并生成代码。
	工具自主调用	基于用户需求自主调用代码检索、文件读取、目录读取、命令执行、文件编辑等多个工具。
	终端命令执行	自主执行命令并捕捉命令执行结果，智能体更具执行结果判断下一步动作。
	错误日志捕捉	执行命令后报错将捕捉并修复。
	语义检索代码库	通过对用户需求的理解语义检索相关代码作为上下文。
	Diff视图审核代码	多文件编辑后Diff视图查看代码。
	代码效果实时预览	针对HTML等文件在线预览，查看代码执行效果。
编辑历史集中管理	多文件编辑后统一查看编辑历史。	
企业管理	用户管理	支持对接企业SSO进行统一用户管理（注：仅在VPC、混合云、私有化场景下可对接）。
	组织管理	支持企业按照部门、组织进行管理，可多级下钻查看各个粒度的组织成员。
	数据看板	支持查看采纳率、AI代码生成占比，并通过IDE、语言、组织等维度查看趋势和明细。
	内容审计	审计在代码补全、代码问答中Prompt是否有敏感信息，并支持企业自主配置过滤规则。
	知识管理	支持企业上传本地知识在技术问答中进行检索知识增强使用，提高代码生成效果。
	多模型接入	支持企业模型自主接入，需要满足OpenAI的接口格式，可以按照组织粒度配置可见范围（注：不支持续写模型接入，仅限于技术问答场景）。

4、部署形态

文心快码支持SaaS、VPC、混合云、私有云四种形态，来满足企业各类数据安全诉求（文心快码对数据无论何种形态，**始终将保障您的数据安全为第一要务**）。四种模式的区别主要在部署方式上，具体如下：

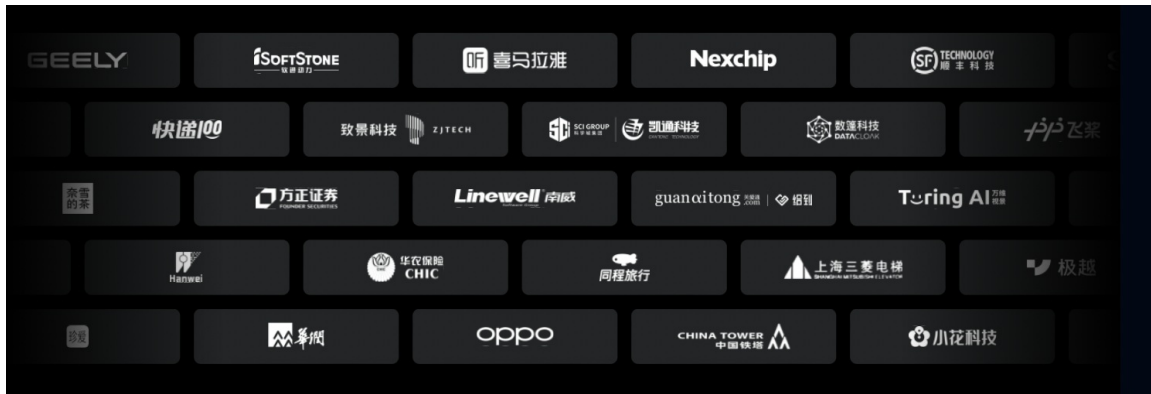


为了帮您更清晰的了解四种模式的异同，可参考下表：

分类	SaaS	VPC	混合云	私有云
企业管理	* 采纳率（次、行）、AI代码生成占比（行）等各类效率观测报表，面向个人、组织、企业等各维度的对比报表，涵盖IDE分布、插件分布、功能分布、语言分布等维度报表			
支持IDE	* JetBrains（涵盖IntelliJ IDEA、PyCharm、GoLand、WebStorm、Android Studio等）、VSCode、Visual Studio、Xcode、Eclipse、DevEco Studio			
支持的语言	* Java、Python、Golang、C++、C、C#、TypeScript、JavaScript、Rust、Ruby、PHP等全部主流语言			
部署形态	* 工程服务、模型均在百度云统一部署	* 在百度云提供企业专属服务集群（数据服务全部VPC，模型在SaaS） * 专线打通 * 定期升级	* 在客户侧部署企业专属服务集群（数据服务全部客户侧，需要提供CPU机器，模型在SaaS） * 定期升级（人工定期更新，无法实时使用最先进的产品能力）	* 在客户侧部署企业专属服务集群（数据服务、模型全部客户侧，需要提供CPU、GPU机器） * 定期升级（人工定期更新，无法实时使用最先进的产品能力）
安全保障	* Comate不存储用户任何代码，不会用于模型训练 * 传输链路加密	* 在百度云提供企业专属服务集群（数据服务全部VPC，模型在SaaS） * 专线打通 * 定期升级	* 在客户侧部署企业专属服务集群（数据服务全部客户侧，需要提供CPU机器，模型在SaaS） * 定期升级（人工定期更新，无法实时使用最先进的产品能力）	* 在客户侧部署企业专属服务集群（数据服务、模型全部客户侧，需要提供CPU、GPU机器） * 定期升级（人工定期更新，无法实时使用最先进的产品能力）
客户支持	* 提供丰富的产品介绍、企业落地经验、AI原生课程 * 7*24小时客户支持			
方案优势	* 以最低的成本，享受最先进的产品能力，定期更新模型	* 更强的数据防护，更快速的服务响应且兼顾服务定期更新，享受最先进的产品能力	* 更强的数据防护	* 最强的数据防护

5、各行各业都在使用文心快码

文心快码在金融、保险、医疗、能源、制造、汽车、物流、半导体、影视娱乐等各行各业均落地并产生显著提效，您可以在这里看到更多企业案例（点击[查看企业案例](#)）。如有需要，我们期待您的联系，免费为您提供专属解决方案：[专属解决方案咨询](#)



## 数据安全

保障您的源代码安全和本地开发环境安全对我们来说至关重要，本节详细描述了文心快码如何保障您的信息安全。如果您在使用文心快码的过程中发现任何与安全相关的问题，请随时通过 [comate@baidu.com](mailto:comate@baidu.com) 与我们联系。

虽然我们已经有几家大型企业信任文心快码，如顺丰、同程、喜马拉雅等。但请注意，我们仍在发展产品和改善安全状况的过程中。如果您在高度敏感的环境中工作，建议您使用文心快码私有化版（[点击进入专属解决方案通道](#)）。我们希望此页面能让您了解我们在安全保障方面的进展，并帮助您进行适当的风险评估。

### 1、认证和第三方评估

文心快码已通过信通院的信息安全性认证，同时也通过了SOC 3、SOC 2 Type 2、SOC 1 Type 2的认证。不会存储您的任何源码信息，更不会用您的数据进行训练。您可以向 [comate@baidu.com](mailto:comate@baidu.com) 发送邮件获取认证报告。

### 2、基础设施安全

我们依赖于百度智能云构建了完善的基础设施，以下详细说明用到了哪些组件以及如何保障数据安全。请注意，您的本地代码数据会碎片化的发送到我们的服务器以便支持文心快码的所有能力，但仅作为量化使用，不会存储您的任何源码信息。

- **百度智能云**：我们的基础设施全部托管在百度智能云，都在中国境内部署，并且根据开发者地域的不同请求可在不同的区域（Region）进行自动分发，以提供更快速的响应。
- **基础模型**：文心快码所使用的模型全部托管在 [百度智能云千帆平台](#)，包含文心一言4.0、文心一言3.5、DeepSeek等一系列模型，提供了完备的权限校验机制和传输链路防护机制。
- **向量存储**：我们使用PGVector存储代码向量数据，该数据经过文件名混淆和路径混淆，并且存储的数据只有向量数据（一个1024维度的矩阵）。只有开发者本地才可以对此数据进行解密，即将向量数据映射为代码数据。
- **百度搜索**：当您通过「#」选择网络搜索时，我们将调用百度搜索的API获取互联网上的实时信息。
- **MongoDB**：针对企业用户，我们在MongoDB中记录您的企业使用文心快码时的采纳数据，用于计算采纳率、AI代码生成占比等，该数据仅是一个「数字」，不会存储任何代码。如果企业不需要统计该信息，可以选择隐私模式并进行关闭。
- **K8S**：我们基于百度智能云的K8S（CCE）部署整个服务，并且基于Prometheus来监控集群状态，基于ELK体系记录日志，日志中不包含任何代码数据。
- **ModelBuilder**：我们基于千帆大模型服务与开发平台（ModelBuilder）用于训练一些自定义模型，训练的数据全部来自合成数据和互联网公开数据。
- **百度统计**：我们使用百度统计在文心快码的官网收集用户点击数据，不存储任何用户身份信息，仅用于用户行为分析以帮助改善产品体验。
- **如流**：如流是我们的内部通信工具，我们可能会在内部聊天中发送一些错误日志用于代码调试和问题排查。
- **如流知识库**：如流知识库是我们内部的文档协作工具，我们可能会在内部协作中通过知识库记录一些错误日志用于代码调试

和问题排查。

我们以最低权限为基础为团队成员分配基础设施访问权限。我们在百度智能云开启了多因素认证，以保证团队成员账号密码不会存在泄漏风险。

### 3、客户端安全

文心快码是插件形态，可以在VSCode、JetBrains、Visual Studio、Xcode、Eclipse等多个IDE使用，完全遵守各个IDE的隐私协议。在使用中，文心快码将向以下域名发出请求，以便与我们的后端服务进行通信。如果您的公司支持代理，请将这些域列入白名单，以确保文心快码正常工作。

- **comate.baidu.com**：针对用户认证、代码索引构建、代码问答、代码补全等所有请求均通过此域名。同时，为了让开发者有更好的产品体验，我们在客户端采用WebView的模式，同时在开发者本地环境启动一个进程（Comate Kernel），该进程负责IDE与服务端的请求中转。如果该进程占用您较多的内存或CPU，请联系我们 [comate@baidu.com](mailto:comate@baidu.com)，我们将提供问题排查服务以确保您的优秀体验。

### 4、远程请求安全

当您在聊天框提问时会发出请求、当您在编码区触发Tab时会发出请求、当您手动触发构建代码索引时会发出请求，所有的请求都被加密，完全基于HTTPS。当发起请求时，请求的内容通常包括您最近查看的文件、对话历史记录、光标附近上下文，最终会组装到一起发给模型。在整个过程，我们不会记录任何代码数据。同时，您拥有文心快码生成的所有代码权益。

### 5、代码库索引安全

文心快码会对代码库进行语义索引，这使它能够在所有代码上下文中的问题，并通过引用现有实现来编写更好的代码。默认情况下，代码库索引处于启用状态。我们的代码库索引功能的工作原理如下：

- 打开一个项目后，它会扫描您在IDE中打开的文件夹，只有Git项目才会被索引，由 '.gitignore' 指定的文件和子目录将被忽略。然后，我们会计算所有文件的Hash，以确定哪些文件是变化的，并增量构建索引。向服务端传输文件时全部是碎片化的，基于Tree-Sitter进行语法树解析后切割，因此服务端接收到的代码片段都是「不完整」的，这进一步保障了代码逻辑安全。一旦向量建立完毕，碎片化的代码也会被清理。
- 在我们的服务器上，我们对碎片化的代码建立向量，并将向量存储在 PGVector 中。为了允许按文件路径过滤向量搜索结果，我们为每个向量存储一个混淆的相对文件路径，以及 Chunk 对应的行范围。我们还将向量存储在Redis的缓存中，按块的哈希值进行索引，以确保再次为同一代码库编制索引的速度要快得多（这对用户体验特别有用）。
- 在推理时，我们通过用户的问题计算一个向量，让 PGVector 进行最近邻搜索，将混淆后的文件路径和行范围发送回客户端，并在本地客户端上读取这些文件块。然后，我们将这些 Chunk 发送回服务器以回答用户的问题。这意味着我们的服务器或 PGVector 中没有存储任何源码。

### 6、账户安全

我们完全遵循百度账号协议，不会存储您的任何密码、手机号，不会明文存储您的邮箱。当您注销您的账号时，我们保证在30天内完全删除您的数据（我们会立即删除数据，但我们的一些数据库和云存储的备份时间不超过30天）。

## 产品定价

本文主要介绍文心快码的计费模式、产品价格等说明信息，方便您快速了解文心快码的相关计费内容。

### SaaS 版

### 计费模式

Comate面向个人用户和企业用户，提供月度、季度、年度的计费方式，并提供个人用户自动续费服务。同时，个人专业版及企业用户支持单独购买智能体请求包。

### 产品价格

Comate 可能不定期开展折扣活动，实际价格敬请参考[官网](#) 实时显示！

版本类型	个人标准版	个人专业版	企业专业版	企业专属版	智能体请求包
单价	免费	¥ 100/人/月 ¥ 270/人/季 ¥ 1000/人/年	¥ 150/人/月 ¥ 450/人/季 ¥ 1500/人/年	¥ 2500/人/年	150元智能体请求券 33元智能体请求券 18元智能体请求券
智能补全	免费	免费	免费	免费	永久有效
智能体权益	赠送智能体请求券 ¥10（首次）	赠送智能体请求券 ¥55/月	赠送智能体请求券¥72/ 人/月	赠送智能体请求券¥72/ 人/月	永久有效
起购数量	/	/	/	100个起购	/

## 🔗 智能体使用说明

### 请求按量计费

智能体请求券消耗额度=消耗的Tokens数x模型单价

PS： [Comate模型单价](#) 页面展示价格为实时参考价，请以账单为准。

### 付费方式

- 智能体代金券：企业智能体代金券全企业范围内可用，额度消耗优先级：赠送的智能体代金券>企业会员的智能体代金券（订阅周期内按月有效）>单独购买的智能体代金券（永久有效）。
- 按量后付费：企业可以开通按量后付费服务方式进行智能体消耗账单的结算。按量计费采用小时后付费方式结算，因此出账存在小时级的延时。只有在出账扣款操作完成后，账户才会进入欠费状态，在此之前你仍可以使用 Comate 企业版，并可能产生费用。

## 🔗 支付方式

购买Comate产品时，您可以选择以下支付方式：当您在[comate.baidu.com](https://comate.baidu.com)购买时，您可以使用支付宝进行支付。当您在智能云官网购买时：

- 百度智能云账户余额，可在 [管理控制台 - 财务总览](#) 处查看当前余额。
- 在线支付，支持使用微信支付和支付宝支付。同时，您可以在百度智能云管理[自动续费模式](#)，您可以设置自动续费或关闭自动续费。

## 🔗 退款说明

请您在购买本服务和支付费用前仔细阅读并理解[用户服务协议](#)和相关的收费规则、政策。如您不同意，请您停止支付行为。如果您实际进行了付费行为，则代表您同意本协议与相关收费规则，您无权转让本服务或要求百度退款；但如果因百度产品技术原因，导致您购买时被重复扣款或因百度产品技术原因导致您无法享有购买的服务时，您可提出退款申诉，经过百度核实后可为您办理退款，但您需要提供相应信息并配合百度处理。

## 🔗 私有化版

### 步骤一：预约演示

- 操作：请点击[此处](#)填写私有化部署预约演示表单。

## 步骤二：个性化报价

- 说明：我们将根据您的企业实际情况，包括但不限于企业规模、使用需求、定制化要求等因素，为您提供个性化的报价方案。

[下载：文心快码Comate](#)

# 常见问题

## 综合问题

### 个人版本、企业版本、私有化版本有什么区别？

个人版：提供业务代码与测试代码的生成、代码优化与修复、自然语言对话式技术问答等多维度辅助编码能力，同时支持100+语言与多种主流 IDE，让你的编码更快、更好、更简单。

企业版：在个人版本的基础之上，同时提供License 管理能力、完备的数据报表能力，助力企业分析应用效果、定位效能瓶颈，一站式赋能研发过程降本提效。

私有化部署版：涵盖企业版本全部能力，同时支撑大型企业规模化部署与应用，保障使用效果，维护数据安全。

### 文心快码支持哪些语言和IDE？

目前文心快码支持全部主流语言，在C、C#、C++、Java、Python、Go、JavaScript、TypeScript、HTML、CSS等语言表现更为出色。

支持的IDE包括：JetBrains全系，包括IntelliJ IDEA、GoLand、PyCharm、WebStorm、CLion、PhpStorm，以及Android Studio，VS Code，Visual Studio，Eclipse，Xcode等IDE插件，一键安装，即装即用。

### 我应该怎么去测试这个插件？

真正的打开一个实际开发的项目，去编写代码就好。随着输入的内容越多，大模型的理解越深刻，推荐的内容也就越精准和有效了。

## 安全问题与隐私协议

### 文心快码会收集我的代码数据？

使用过程中需要利用正在编写的文件中的上文，来推理出接下来的下文。因此确实会需要通过网络使用过程中将正在编写的代码文件内容传输给模型推理。我们承诺不使用这部分上传的代码做任何积累、分析、加工等后续动作。

私有化部署情况下，模型和服务都部署在客户私域内，数据不会流出。

SaaS及私有化场景均可放心使用！

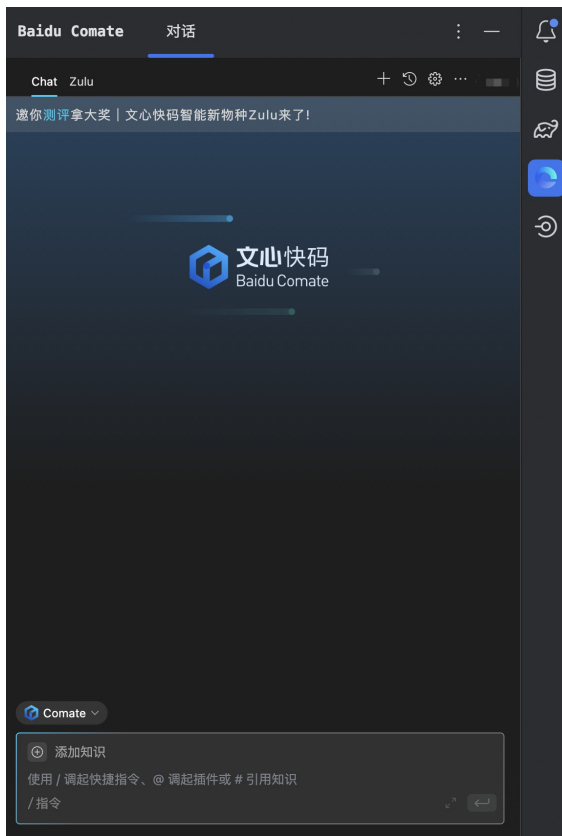
### 我的代码会和其他用户共享吗？

不会。我们根据隐私协议，遵循负责任的做法，确保您的代码片段不会被用作文心快码其他用户的建议代码。

## 插件使用问题

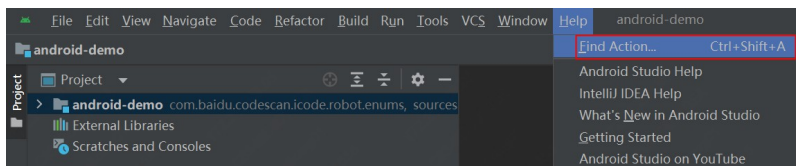
### Android Studio配置文心快码新版侧边栏

由于Android Studio默认的限制，部分高级功能默认未开启。为了您在Android Studio下使用文心快码时拥有更好的体验，推荐您按照下面的方式设置支持JCEF的Boot Runtime，体验更友好的会话、Zulu和开放平台等能力。

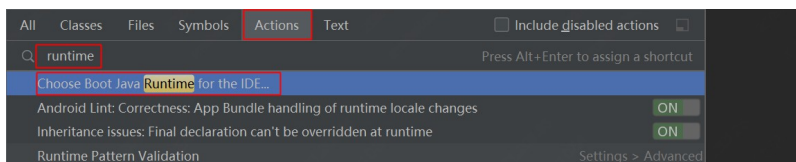


## 操作步骤

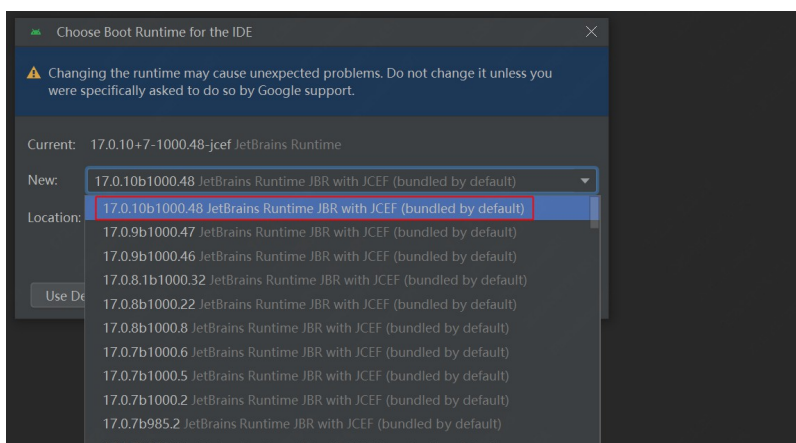
### 1. Help → Find Action



### 2. 输入"runtime" 找到 "Choose Boot Java Runtime for the IDE"



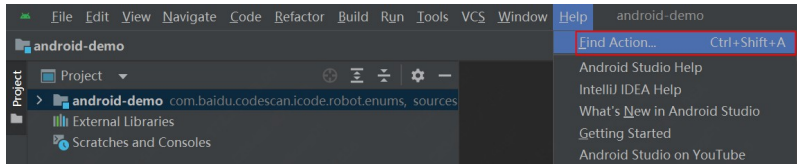
### 3. 选择一个支持JCEF的Runtime (通常选择最新即可)



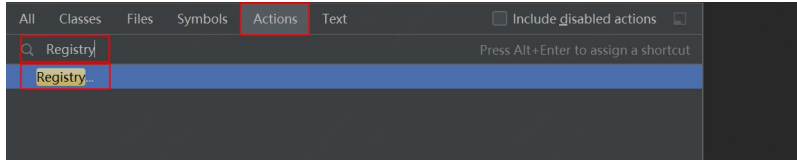
### 4. 点击【OK】后IDE会下载相应版本的Runtime，下载成功后重新启动IDE，如果文心快码侧边栏样式与上面的一致即可正常使用文心快码的所有功能，如果未能加载出文心快码侧边栏请参考下方教程

我的Runtime支持JCEF，但出现了白屏/功能不全

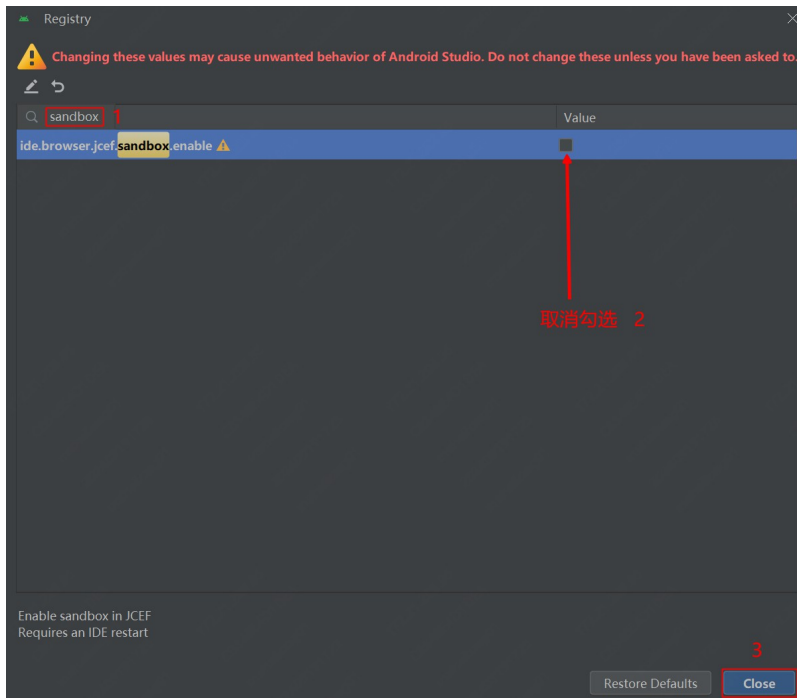
## 1 Help → Find Action



## 2 输入"Registry" 找到 "Registry..."



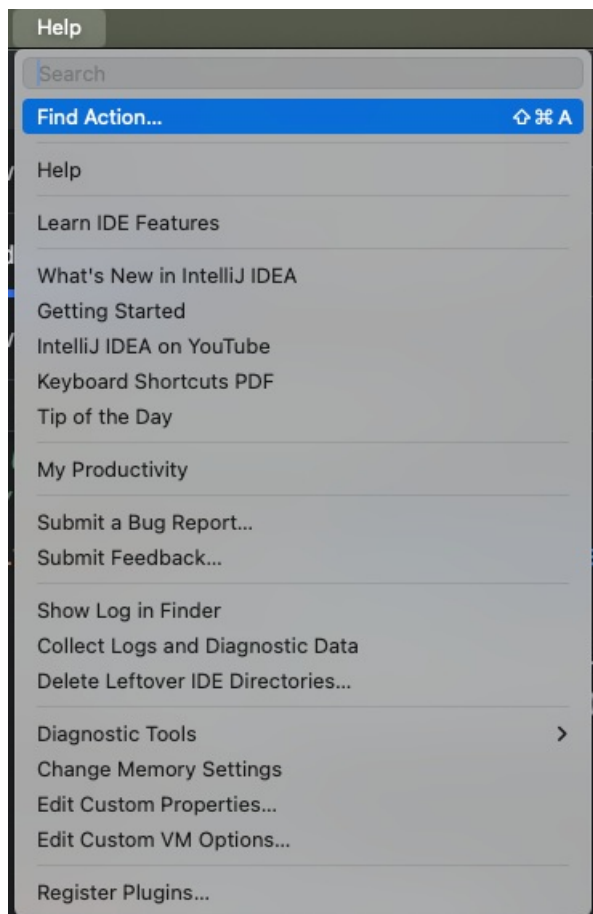
## 3. 输入 "sandbox" 找到 "ide.browser.jcef.sandbox.enable" 取消勾选后重启IDE



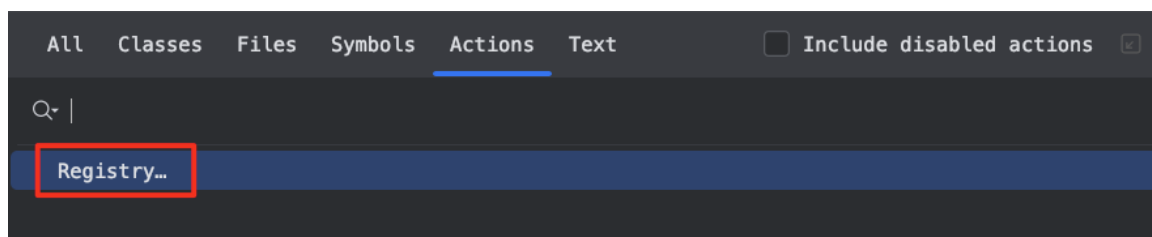
🔗 JetBrains IDE 2025.1版本，打开多个项目，无论在哪个项目中提问，对话都会在同一项目的侧边栏中展示

这个问题是由于JetBrains IDE在2025.1修改了内部的实现机制导致。如果您安装的插件版本大于等于3.6.1，请重启下IDE。如果仍旧存在问题，可以用下面的方式来手动关闭

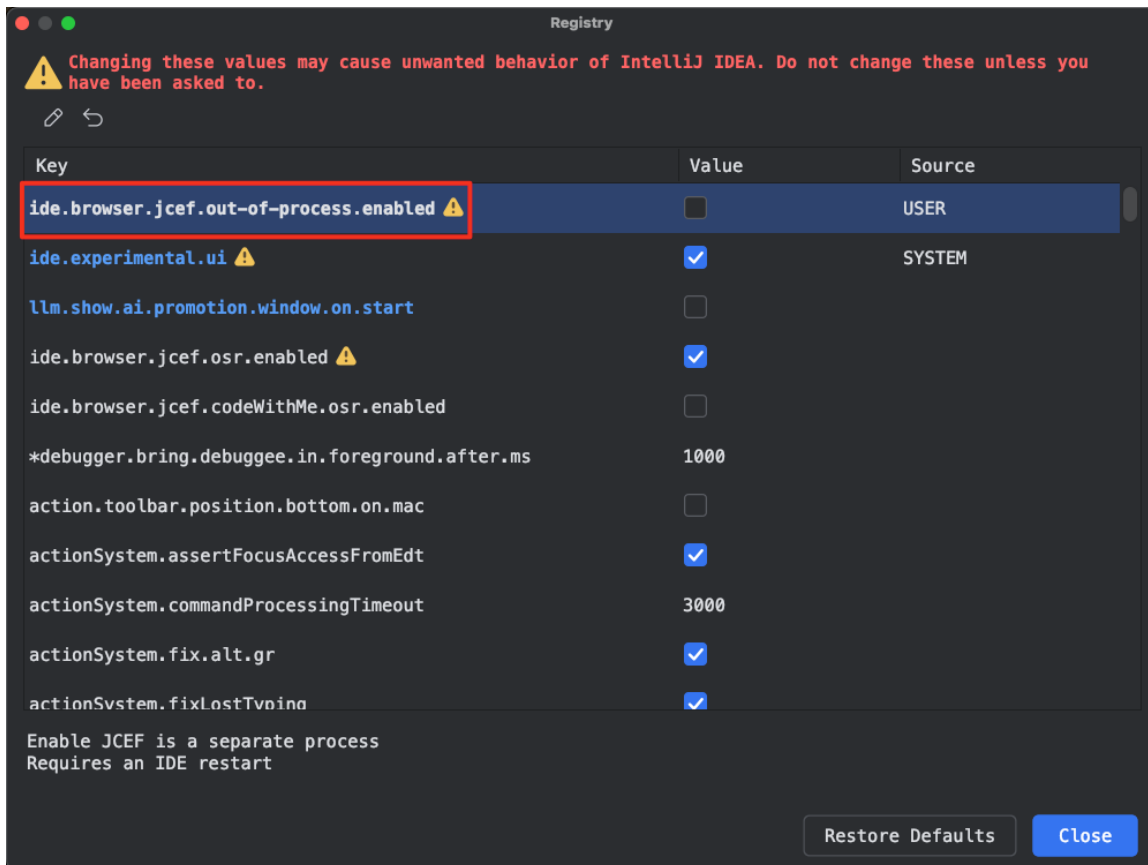
## 1. Help -&gt; Find actions



2. 弹出的框输入Registry



3. 查找ide.browser.jcef.out-of-process.enabled并将其Value的勾选框取消勾选



#### 4. 重启IDE确认配置生效

## 服务协议和隐私协议

### Baidu Comate用户服务协议

欢迎您体验BaiduComate服务并阅读《BaiduComate用户服务协议》！

《BaiduComate用户服务协议》（以下简称“本协议”）是您与北京百度网讯科技有限公司（以下简称“百度”）之间关于“BaiduComate”中百度提供的各项服务的法律协议，具有合同效力。

百度在此特别提醒，请您在使用BaiduComate服务前阅读并充分理解本协议，特别是免除或者限制责任的相应条款（该等免责、限制条款可能以**黑体加粗**、**颜色标记**或其他合理方式提示您注意），并选择“接受”或“不接受”。

百度可能会根据BaiduComate的整体规划等，对BaiduComate及/或本协议进行修改或变更，若您不同意BaiduComate及/或本协议的前述修改或变更，您可停止使用BaiduComate服务。您使用BaiduComate服务即视为您接受并同意本协议所有条款，包括但不限于前述修改及变更。如违反本协议约定，百度有权随时中止或终止该BaiduComate服务。

#### 一、定义和说明

##### 1.1 【BaiduComate】

BaiduComate服务是百度在comate.baidu.com网站、百度智能云智能代码助手产品及BaiduComate插件中提供的产品或服务。您如果需要使用BaiduComate服务时，需要您先完成百度账号的注册或登录，并购买对应的使用时长，即可使用本服务。

##### 1.2 【本服务】

在本协议中，“本服务”特指“BaiduComate服务”，指用户通过付费购买或好友赠送等方式获取的增值权益服务，包括但不限于BaiduComate插件使用权、数据分析等，BaiduComate具体服务内容会以“产品功能介绍”页面和相关服务页面公布、实际提供的内容为准。

百度郑重声明：本服务不影响百度向您提供的现有的一切免费服务并保证服务质量。您理解并同意，鉴于网络服务的特殊性，百度有权根据市场情况进行会员权益及特色会员服务的全部或部分变更、调整、取消、增加，并可能会对您已享有或正在享有

的会员权益造成影响。您同意，如您继续按照调整后的内容使用BaiduComate服务，您不要求百度承担任何责任。

### 1.3 【BaiduComate服务使用规则】

即BaiduComate服务提供方不时发布并修订的关于BaiduComate的用户守则、活动规则、使用规则、公告、提示、通知及在线FAQ等内容。

### 1.4 【协议更新】

百度可以对本协议下服务条款进行更新（包含变更、增加、减少相应的条款内容），一旦服务条款发生更新，百度会通过系统提示、信息推送和/或您在我方平台留下的联系方式通知和提醒您注意。更新条款一经正式发布，即为本协议不可分割的组成部分。如果您不同意变更后的协议，可以停止对本服务的使用。您继续使用百度提供的服务，则视为您已经接受本协议的全部修改。

### 1.5 【本协议】

本协议是基于《BaiduComate用户协议》、《BaiduComate隐私协议》、《使用百度前必读》、《百度支付协议》以及百度相关协议规范（简称为“百度服务条款”）制定，是其不可分割的组成部分，与其构成统一整体。本协议与上述内容存在冲突的，以本协议为准。

同时，百度不时发布的关于BaiduComate的相关协议、服务规则等也是本协议的一部分，您在使用BaiduComate服务的同时应遵守本协议、百度服务条款及百度旗下相关会员产品协议、服务规则等所有条款。

## 二、账号管理及使用规则

### 2.1 【账号管理】

2.1.1 您需要先注册或登录百度账号来登录BaiduComate，方可购买BaiduComate提供的各项收费服务，如您的手机号下有多个百度账号，您购买的账号权益将仅发放到您购买时登录的百度账号。您应正确使用及妥善保管、维护您的百度账号及密码，如发生任何泄漏、遗失、被盗等行为，而该等行为并非百度法定过错导致，所有损失将由您自行承担。

2.1.2 您在此授予百度保护您账号安全的相关权利，百度可定期或不定期采用不同的方式对您账号的使用安全进行检查、验证，包括但不限于主动联系您进行身份验证、短消息验证、邮箱认证及密码修改等。如您无法完成验证或无正当理由拒绝验证的，百度可合理怀疑您的账号出现异常或被盗，并中止向该账号提供服务或采取进一步措施。

### 2.2 【使用规则】

#### 2.2.1 【合法获取及使用】

(1) 使用BaiduComate服务及/或参加相关活动时，亦应通过百度官方公布/授权的方式进行，不得采取出售、转让、盗用、租赁其他用户账户等方式进行BaiduComate账号注册或通过出售、转让、盗用、转借其他BaiduComate账户等方式享用BaiduComate服务。任何以盗窃、利用系统漏洞、通过任何非百度官方或授权途径获得的BaiduComate服务（包括但不限于购买、租用、借用、分享、受让等方式获得）、恶意利用或破坏BaiduComate活动中获得的BaiduComate服务权益（包括但不限于账号、服务权益、服务激活码等）或参加活动获取福利（包括但不限于获得实体/虚拟礼品、服务权益、服务激活码等），均不获相关保护，且一经发现百度有权立即封停账户及/或取消其BaiduComate用户资格，且不予任何赔偿或退还费用/使用时长，所有产生的损失及责任由行为人自行承担。

(2) 您购买BaiduComate服务后，可享有BaiduComate的各项专属特权和增值服务，具体内容可参考价格页，同时应遵守BaiduComate服务的相关协议及服务规则等。

(3) 如您违反本协议或百度服务条款，百度有权利取消您的BaiduComate使用资格，百度可根据您的实际情况采取修改、限制、中止或终止全部或部分BaiduComate使用服务或追究您的法律责任等措施，百度也无需向您退还任何费用/使用时长，且您需承担您、第三方及/或百度由此遭受的损失。

#### 2.2.2 【账号独立】

(1) 同一主体拥有多个百度账号并进行BaiduComate用户登录的，每一个百度账号将被识别为一个独立的账号。

(2) 每个百度账号之间的充值记录、时长权益无法进行转移、迁徙、转让、赠与、售卖、租借、分享，无论该等账号是否由同

一、使用人拥有，请您在登录及/或充值/参加活动时注意区分，避免造成损失。

### 2.2.3 【账号找回】

您登录BaiduComate时使用的账号信息遗失、忘记密码，导致无法登录的，您如需找回的，需按照百度官方公布的账号找回流程的要求提供相应的信息和证明，并确保提供的信息和证明文件的合法真实有效，若提供的材料不真实或不符合要求，无法通过百度的安全验证，可能会导致账号找回失败，相应的风险和损失将由您自行承担。您理解百度对您的请求采取行动及处理均需要合理时间，除百度存在法定过错外，百度对您的损失不承担任何责任。

### 2.2.4 【妥善保管】

请您负责妥善且正确地保管、使用、维护您的BaiduComate用户资料、账户信息及账户密码。请您对您账户信息和账户密码采取必要和有效的保密措施。

## 三、服务内容及服务规范

### 3.1 【服务内容】

3.1.1 在您购买使用时长后，根据您的购买时长和类型情况将会享有对应的服务体验，包括但不限于在IDE内使用BaiduComate插件、企业用户可在控制台查看数据看板等，具体以相关服务页面公布、实际提供的内容为准。

3.1.2 您在购买BaiduComate使用时长后可享受的全部权益以百度官方发布的产品功能内容为准，百度有权基于自身业务发展需要变更全部或部分服务及权益。就前述权益调整百度会通过相应服务页面更新展示、系统提示、信息推送和/或您在百度平台留下的联系方式通知和提醒您注意，您也可通过百度官方网站或BaiduComate官方网站查询最新的产品功能及服务内容。同时我们也鼓励您定期查看本协议内容，以更好地保障您的权益。

3.1.3 为了向您提供更好的服务，百度可能会对产品功能及服务进行调整，包括但不限于：

- (1) 产品功能和服务具体内容的调整
- (2) 个人版用户和企业版用户的功能变更
- (3) 付款方式、价格等的变更
- (4) 其他平台公示的变更内容

### 3.2 【权益独立】

BaiduComate服务提供的会员权益与百度旗下产品会员提供的权益独立不冲突，即您在付费购买BaiduComate服务后，如您同时也开通百度旗下产品（包括但不限于文心一言、百度网盘、百度文库、百度阅读等）相应会员服务的，所涉及重合的权益内容您可叠加享受。

### 3.3 【服务规范】

您在使用BaiduComate服务时，需要注意遵守以下服务规范：

3.3.1 您在使用BaiduComate期间上传、生成、复制、存储、发布、传播的任何内容，包括但不限于账户头像、账户信息、输入及生成的文字及代码等，均需要遵守国家法律。

3.3.2 您不得自行或帮助他人上传、诱导生成及传播以下内容：

- (1) 违反国家规定的政治宣传和新闻信息；
- (2) 涉及国家安全和国防利益的信息；
- (3) 封建迷信和淫秽、色情信息；
- (4) 涉及博彩有奖、赌博游戏、“私服”、“外挂”等非法互联网活动的相关信息；
- (5) 违反国家民族和宗教政策的信息；
- (6) 妨碍互联网运行安全的信息；

(7) 侵害他人合法权益的信息和其他有损于社会秩序、公共道德的信息；

(8) 其他违法法律、法规、部门规章或国家政策的信息。

3.3.3 您确保不使用任何方式攻击BaiduComate的任何网页端或插件端，不得修改、改变、翻译BaiduComate所使用的软件、模型等技术，不得通过反向工程、反向编程或其他行为获取BaiduComate相关源代码，否则因此引起的一切法律后果将由您承担，我们有权依法追究您的法律责任。

3.3.4 未经BaiduComate同意，您不得直接将本产品及服务对外进行商业化，或以本产品及服务为基础进行二次开发后形成类似插件对外进行商业化。

#### 3.4 【服务性质】

基于虚拟产品的性质和特征，您所获取的虚拟产品不能够进行退货、换货、兑换现金，亦不得用于商业领域，如买卖、置换、抵押等。为了保证您的用户权益，请您通过官方渠道购买BaiduComate服务，一切通过非官方公布渠道取得的虚拟产品及其衍生服务均不获保护，百度有权中止或终止对您的服务；由此带来的损失，百度不承担相关责任。

#### 3.5 【设备及系统差异】

您通过百度指定渠道开通本服务后，可能会由于您使用的软件版本、设备、操作系统等不同以及其他第三方原因等导致您实际可使用的具体服务有差别，由此可能给您带来的不便，您表示理解，您不予追究或者豁免百度的相关责任。

#### 3.6 【关于收费】

本服务是百度提供的收费服务，您须在按照本服务的收费标准支付相应费用后，方可使用本服务。涉及收费环节时，我们将在相关用户界面上做出提示。如您拒绝支付费用，则无法继续享受BaiduComate的相关服务，但您仍可以有限制的免费使用本服务（详情见价格页面）。

此外，您可能也可以通过接受好友赠送等百度认可的其他方式享有、使用本服务。

百度可能会根据本服务的整体规划，对本服务收费标准、方式等进行修改和变更，前述修改、变更，百度将在相应服务页面进行更新展示。百度会根据实际状况，对不同阶段购买、续费的会员给予不同的优惠，具体优惠政策和收费规则以百度在相关服务页面公示的信息为准。您在此理解并同意，如果百度发现用户滥用收费规则获取优惠等任何作弊的行为，可以判定该等用户享有的优惠无效。

#### 3.7 【支付】

3.7.1 您应该通过百度指定的包括但不限于支付宝支付、微信支付等方式等现有方式或今后百度指定方式，依约支付。

请您在购买本服务和支付费用前仔细阅读并理解本协议和相关的收费规则、政策。如您不同意，请您停止支付行为。如果您实际进行了付费行为，则代表您同意本协议与相关收费规则，您无权转让本服务或要求百度退款；但如果因百度产品技术原因，导致您购买时被重复扣款或因百度产品技术原因导致您无法享有购买的服务时，您可提出退款申诉，经过百度核实后可为您办理退款，但您需要提供相应信息并配合百度处理。

3.7.2 您不得通过以下任何方式为自己或他人开通本服务：

- (1) 以营利、经营等非个人使用的目的为自己或他人开通本服务；
- (2) 通过任何机器人软件、蜘蛛软件、爬虫软件、刷屏软件等任何程序、软件方式为自己或他人开通本服务；
- (3) 通过任何不正当手段或以违反诚实信用原则的方式为自己或他人开通本服务；
- (4) 通过非百度指定的方式为自己或他人开通本服务；
- (5) 通过侵犯百度或他人合法权益的方式为自己或他人开通本服务；
- (6) 通过其他违反相关法律、行政法规、国家政策等的方式为自己或他人开通本服务；

3.7.3 您理解并认可，如果您通过第三方支付工具在百度平台账户支付或充值后可能产生的任何商业风险（包括但不限于不法分子利用您账户或银行卡等有价卡等进行违法活动，第三方代为充值后再退款导致您的服务权益无法实现；采用第三方黑产渠道

代为充值，或通过第三方非正规渠道购买远低于市场价值的BaiduComate的服务权益等），该等风险均有可能给您造成相应的经济损失，百度在充分履行其在本协议项下义务和符合法律规定的前提下，仅在法律有明确规定的情况和范围内，承担相应的责任。

### 3.8 【服务期限】

3.8.1 您理解并同意，本服务具有一定的服务期限。百度可能向您提供可选择的服务期限（如月度、季度、年度等，具体以产品页面为准）。除本协议另有约定外，本服务的期限以您自行选择并支付相应服务费用的期限为准，自您选择并购买本服务后生效。本服务的有效日期以BaiduComate服务开通的时间算起，直至选定的服务期限到期或您终止本服务/注销百度账号或您因违规使用导致账号封禁时终止。您可以在本服务的相应页面查询服务期限选项与剩余服务期限。

3.8.2 百度在此特别提示，BaiduComate服务下由百度或第三方提供相关权益内容可能有使用时间有效期的约定，如活动获得的权益等，您应当在各自权益载明的有效期内使用。逾期未使用导致权益失效的，即视为您自动放弃该权益。

3.8.3 您的BaiduComate使用时间到期前，百度将通过短信等方式提示您BaiduComate即将到期，您可再次通过购买/获赠/兑换等具体方式延长您的BaiduComate使用时长。若您的BaiduComate使用时间到期时，百度将终止提供BaiduComate对应服务。

3.8.4 请您理解，由于互联网服务的特殊性，为向您提供优质服务，百度须不定时进行服务器的维修、调整、升级等，且在黑客攻击、计算机病毒侵入、监管机构政策通知等不可抗力事件发生后，百度须解决故障并处理相关侵权事宜，因此您接受，本服务的服务期限包含百度解决故障、服务器维修、调整、升级、因第三方侵权处理等所需的合理时间，百度对上述情况所需用的时间不另行补偿。

### 3.9 【服务保障】

百度在为BaiduComate用户提供多项专属增值服务同时，尽最大努力确保BaiduComate服务的稳定性，为此提供各项技术及客服支持，包括但不限于：

3.9.1 开通、获赠成为BaiduComate用户期间所遇到的注册、登陆、购买、兑换、赠送等问题的网上咨询客户服务；

3.9.2 BaiduComate服务积极有效的内容支持和技术支持，包括日常运营、维护、修复及更新；

3.9.3 BaiduComate服务提供期间产生的常见问题解答、基本规则解释；

3.9.4 尽最大努力防止BaiduComate用户服务的服务器和技术设备遭受恶意攻击，以保障您的合法利益；

3.9.5 百度提供给您的其他技术和/或服务。

### 3.10 【保密责任】

我们深知个人信息对您的重要性，尊重BaiduComate用户的个人隐私是百度的责任，我们将严格遵守法律法规要求采取对应的安全保护措施，百度在未经BaiduComate用户的授权时不会公开或透露其BaiduComate注册资料及保存的非公开内容，但由于法律法规需要等原因除外。您可以通过《BaiduComate隐私协议》详细了解我们如何收集、使用、保护您的个人信息。

## 四、违约责任及责任限制

4.1 您在使用本服务的过程中，不得进行以下违约行为：

4.1.1 违反本协议约定的；

4.1.2 违反法律、法规、规章、条例以及任何具有法律效力之规范规定的；

4.1.3 破坏、影响百度对任何第三方提供本服务；

4.1.4 进行危害计算机网络安全的行为；

4.1.5 对百度及关联第三方（包括授权方和合作方）提供的本协议服务、活动造成不良影响，侵犯及/或关联第三方百度及/或其他用户的正当利益；

4.1.6 被有权机构认定侵犯任何第三方的合法权益的；

4.1.7 利用本服务获取非法利益的，包括但不限于通过转售、转让、转授权等行为不正当牟利；

4.1.8 其他侵犯百度合法利益的行为。

#### 4.2 【违约行为处理】

您确认并同意，如果百度发现或收到他人举报您有违反本协议任何行为的，百度有权依法进行独立判断并采取以下一项或多项措施处理：

(1) 根据相应规则立即删除、屏蔽或断开相关的信息；

(2) 采取包括但不限于中止或终止部分或全部本服务的措施；

(3) 采取扣除违约行为发生时间内的会员权益时长的措施；

(4) 百度无需向您退还任何费用，未使用的服务费用作为违约金归百度所有，您的相关使用权益的损失由您自行承担，百度在法律有明确规定的情况下承担相应的责任；

(5) 如您的行为使百度遭受损失（包括自身的直接经济损失、商誉损失及对外支付的赔偿金、和解款、律师费、诉讼费等间接经济损失），您应赔偿百度的上述全部损失；

(6) 如您的行为使百度遭受第三人主张权利，百度可在对第三人承担金钱给付等义务后就全部损失向您追偿。

#### 4.3 责任限制

4.3.1 关于本服务百度不提供任何种类的明示或暗示的担保或条件，包括但不限于本软件的适用性、无疏忽或无技术瑕疵问题、商业适售性、特定用途适用性等，对在任何情况下因使用或不能使用本软件所产生的直接、间接及后续的损害及风险由您自行承担；您利用本软件向任何第三方提供服务，相应的权利义务和责任均由您承担，百度不对由此造成的任何损失负责；

4.3.2 百度对本服务所涉的技术和信息的有效性，准确性，正确性，可靠性，质量，稳定，完整和及时性均不作承诺和保证；

4.3.3 如因本软件使用的第三方软件或技术引发的任何纠纷，应由该第三方负责解决，百度不承担任何责任，百度不对第三方软件或技术提供客服支持，若你需要获取支持，请与第三方联系。

### 五、服务的变更、中止及终止

5.1 本服务的中止或终止包含如下情况：

5.1.1 用户主动中止或终止，包括但不限于使用时长到期未进行续费等；

5.1.2 因为用户违约行为，百度主动中止或终止服务的；

5.1.3 因国家或相关部门要求或发生不可抗力事件时，百度中止或终止服务的；

5.1.4 其他根据法律法规或百度业务调整等应当或者需要中止或终止服务的；

5.2 中止或终止服务后，百度将尽最大努力通过合理方式通知用户。当您发现无法登录或享受服务时，可以咨询我们的在线客服。

5.3 中止或终止后的处理：

5.3.1 除法律规定的责任外，百度对本服务的中止或终止对用户和任何第三人不承担任何责任；

5.3.2 除法律规定或本协议另有约定外，已收取的费用不予退还；

5.3.3 正在进行的平台交易，百度将根据情况合理判断，是否继续进行或终止相关交易，并视情况决定是否进行退费或扣除相关费用作为违约金。

5.4 百度依据本协议的约定变更服务内容、服务方式以及本协议内容的，如果BaiduComate用户不同意百度的变更，有权选择取消并停止使用已经获取的对应的全部或部分服务；如果BaiduComate用户继续使用百度提供的上述服务，则视为BaiduComate用户已经接受百度的上述调整。

5.5 BaiduComate用户如果不同意条款的修改，可主动向百度提出终止服务，但百度不退还自开通服务至提出终止服务所在月的已享受服务的部分费用；如果BaiduComate用户继续享用服务，则视为BaiduComate用户已经接受条款的修改。

## 六、通知和送达

本协议项下百度对于用户所有的通知均可通过网页公告、手机短信或常规的信件传送等方式进行，该等通知于发送之日视为已送达用户。为便于您收取相关通知，您应确保预留的联系方式为正确及有效，并在变更后及时登录百度修改，如因您未提供正确的联系信息或未及时告知新的联系方式，导致无法及时收取通知，将由您自行承担由此产生的损失及法律后果。百度可通过上述方式之一或其中若干向您送达各类通知，而此类通知的内容或将对您产生重大有利或不利影响，请您务必确保联系方式为有效并请及时关注。

## 七、法律的适用和管辖

7.1 本协议的订立、执行和解释及争议的解决均应适用中华人民共和国法律。

7.2 如双方就本协议内容或其执行发生任何争议，双方应尽量友好协商解决；协商不成时，任何一方均可向百度所在地有管辖权的人民法院提起诉讼。

7.3 百度未行使或执行本服务协议任何权利或规定，不构成对前述权利或权利之放弃。

7.4 如本协议中的任何条款无论因何种原因完全或部分无效或不具有执行力，本协议的其余条款仍应有效并且有约束力。

7.5 在法律法规允许的范围内，百度有权对本协议进行解释。

## 八、其他

如您对本协议有任何疑问或问题，请通过comate@baidu.com联系我们

北京百度网讯科技有限公司

# Baidu Comate隐私协议

欢迎您使用Baidu Comate提供的服务！百度公司（本隐私政策中指北京百度网讯科技有限公司，以下亦称“我们”）非常重视您的隐私保护和个人信息保护。本隐私政策适用于您通过任何方式对Baidu Comate提供的各项服务的访问和使用。

您可以通过Baidu Comate实现智能代码助手服务（以下统称“BaiduComate产品或服务”）。我们仅会处理您使用产品或服务所需的个人信息。

本隐私政策适用于您通过Baidu Comate智能代码助手网页端、移动端等方式来访问和使用我们的产品和服务。

Baidu Comate包括但不限于Visual Studio Code、Jetbrains、Visual Studio、XCode等IDE的多个版本，请您在使用Baidu Comate时选择与所安装终端设备相匹配的版本。

本隐私政策旨在帮助您了解我们会收集哪些个人信息、为什么收集这些个人信息，会利用这些个人信息做些什么及如何保护这些个人信息。我们会遵循隐私政策收集和使用您的个人信息，但不会仅因您授权同意本隐私政策而采用强制捆绑的方式收集您全部的个人信息。我们会严格按照法律法规要求，在特定情况下征求您的单独同意。除本隐私政策之外，我们还可能会通过产品页面告知、弹窗提示、信息推送等方式向您说明我们处理您个人信息的规则，具体业务功能对应的个人信息处理情况将以具体规则为准，具体规则与本隐私政策具有相同效力。

除法律、行政法规另有规定外，针对我们如何处理您的个人信息，您享有知情权和决定权，有权限制或者拒绝我们或其他方对您的个人信息进行处理；同时，您有权要求我们对向您提供的个人信息处理规则进行解释说明。请您在使用Baidu Comate及相关产品和服务前仔细阅读并理解本隐私政策，以便做出适当的选择。

本隐私政策将帮助您了解以下内容：

1. 我们收集的信息及相应的使用目的
2. 我们如何共享、转让、公开披露、委托处理您的个人信息
3. 我们如何保存及保护您的个人信息

4. 您的权利
5. 我们如何处理未成年人的个人信息
6. 本政策如何更新
7. 如何联系我们

我们深知个人信息对您的重要性，也深知为您的信息提供有效保护是我们业务健康可持续发展的基石。感谢您对Baidu Comate的使用和信任！我们致力于维持您对我们的信任，恪守以下原则，保护您的个人信息：权责一致原则、目的明确原则、选择同意原则、最少够用原则、确保安全原则、主体参与原则、公开透明原则等。

我们郑重承诺，我们将按业界成熟的安全标准，采取相应的安全保护措施来保护您的个人信息。

### 1.我们收集的信息及相应的使用目的

我们将遵循合法、正当、必要的原则，收集和使用您的个人信息。具体而言，我们出于为您提供相应的业务功能而收集并处理您的个人信息。相应功能中，如果您提供的是他人个人信息，请您确保已取得相关主体的授权，我们保留随时核查该等个人信息合法来源的权利。

您需要向我们提供个人信息以供我们实现Baidu Comate的基本业务功能。为实现基本业务功能所必需的信息为必要个人信息，若您拒绝提供这些必要个人信息，我们将无法向您提供该基本业务功能及相应服务。在基本业务功能之外，您可以自主选择向我们提供部分个人信息以供我们实现Baidu Comate的扩展业务功能。我们将向您告知使用的目的并基于您的授权同意或实现您选择的扩展业务功能必需的目的处理这些个人信息。您拒绝提供该等扩展业务功能信息，不会影响您使用我们的基本业务功能，但如您拒绝提供实现扩展业务功能必需的个人信息，我们将无法为您提供扩展业务功能。

为方便您的理解，我们将所提供的业务功能、处理目的及其所需个人信息如下所列：

#### 1.1.账号服务

1.1.1. 您可以通过手机号创建百度通用账号，并且您可以完善您的账号信息。当注册、登录百度通用账号时，您需向我们提供您的手机号码。您可以选择设置账号名称、头像、登录密码、账号绑定邮箱。

1.1.2.您也可以使用第三方账号（如微信、微博、QQ、Github、Gitee）登录Baidu Comate。此时，您将授权Baidu Comate获取您在该第三方账号注册的公开信息（包括您的昵称、头像及您授权的其他信息）。我们将重新验证并绑定您的手机号码，以确认该第三方账号的真实性和关联性、满足账号实名制要求。

1.1.3. 为了满足法律法规的强制性要求、出于提供产品或服务所必需或者基于您对增强账号保护级别的需求，在您使用 Baidu Comate深度合成相关服务时或您有进一步需求时，您需提供您的真实身份信息（真实姓名、身份证号、刷脸信息，根据实际情况可能有所不同）以完成账号实名认证。为了验证您提供信息的准确性和完整性，或保障您的账号安全，我们可能会将您向百度提供的实名信息与法律法规允许的机构或政府机关授权的机构所保存的您的对应信息进行比对核验。如您不进行实名认证，并不会妨碍您使用Baidu Comate的其他业务功能，但可能会影响到您使用需要经过实名认证才能使用的业务功能。

#### 1.2. 智能问答和智能代码补全服务

1.2.1. 智能问答服务为您提供了智能代码问答、代码解释、自然语言生成代码、高级生成（AutoWork）、代码修复等服务。智能问答服务依赖于您输入的内容，我们需要收集并记录您输入的需求描述信息，并根据对应的知识或代码库为您生成对应的智能问答。我们理解您输入的问答内容、提交的信息反馈中可能包含他人的个人信息，如包含，请您务必取得他人的合法授权，避免造成他人个人信息的不当泄露。为向您提供连续性、一致化的使用体验，保障服务质量，我们会记录您的个人对话记录，包括您输入的文本/语音对话信息、以及基于上述信息形成的对话主题。您可以查找并管理您尚未删除的问答历史。请您知悉，为符合相关法律法规要求，履行网络信息安全义务，我们会在法律规定的时间内保留您的问答历史及对话主题。

1.2.2. 智能代码生成服务为您提供基于代码上下文的代码补全，包括但不限于代码续写、注释生成代码、生成单元测试、长函数拆分等代码生成类服务。智能代码生成服务依赖于您的代码上下文信息以完成代码补全。

1.2.3. 您基于本服务上传和生成的代码信息完全由您所有及控制，除本协议包括的场景和目的外，**在没有另行获得您的明确授权的情况下**

，不会将您的代码用于任何您没有授权的场景，我们不会对您的代码做任何存储、分析、加工、训练等后续动作。

1.2.4.在问答过程中，您可以对baidu comate所输出的内容进行评价，包括点赞、点踩、反馈等。我们将收集您的评价信息，包括您主动提交的报错原因，以便不断改进的输出内容质量。如您拒绝我们收集和处理前述个人信息，请您谨慎输入前述信息，但因此您可能会影响您正常使用提供的部分或全部功能。

### 1.3.新服务和功能运营及客服处理

Baidu Comate会不时针对新推出的服务和功能发起运营活动，如推出的新服务需要处理您的个人信息或申请设备系统权限，且此前未经过您的授权同意，我们会另行获得您的同意，其中，如涉及敏感个人信息的处理，我们会另行获得您的单独同意。

如果您参加相关运营活动，我们可能会通过运营活动界面收集包括但不限于您的**身份信息**、联系方式、**交易账号类信息**，以便与您取得联系、向您发运营奖励等。若您不参加相关运营活动则无需提供前述信息。

如您对于我们提供的产品服务有疑问、进行反馈或需要客服介入进行争议处理时，为了验证您的身份、定位您的问题、与您取得联系，可能需要您提供您的联系方式（如电子邮箱、手机号码等）**身份信息**。为了保障沟通质量，确保纠纷得到妥善解决，我们可能会保存您与我们的沟通记录，包括IM记录、**通话记录**、您提供的与您的争议解决相关的信息，您使用或提供的联系方式。

### 1.4.消息通知

对于您在使用产品及/或服务的过程中提供的您的联系方式（例如：联系电话、联系邮箱），我们在运营中可能会向其中的一种或多种发送多类通知，用于用户消息告知、身份验证、安全验证、用户使用体验调研等用途。此外，我们也可能会以短信、电话的方式，为您提供可能感兴趣的服务、功能或活动等商业性信息及其他商品或服务信息，如您不愿接受这些信息，可以通过手机短信中提供的退订方式进行退订，也可以直接与我们联系进行退订。

### 1.5.保证服务安全、优化和改善服务目的

当您使用Baidu Comate时，为了保障您正常使用Baidu Comate服务，更准确定位并解决您在使用Baidu Comate服务时遇到的问题，改进及优化Baidu Comate的服务体验，保障您的账号安全，预防、发现、调查欺诈、危害安全、非法或违反我们的协议、政策或规则的行为，以保护您、我们或我们的关联公司、合作伙伴及社会公众的合法权益，我们可能会收集您在使用过程中智能问答的代码区复制、插入等操作事件；码补全/续写的触发次数、采纳次数；代码补全/续写采纳项的排序位置；代码补全/续写采纳项的字符长度；代码补全/续写的平均响应时间；代码示例搜索关键词提示触发事件；设备信息（包括您正在使用的硬件信息、操作系统信息等）、设备连接信息（软件版本号、使用的语言、IP地址等）及其他与登录环境相关的信息来帮助我们进行服务情况分析，并进一步帮助我们能有效地增强您的使用体验。

### 1.6.事先征得授权同意的例外

请注意：在以下情形中，收集、使用个人信息无需事先征得您的授权同意：

- 为订立、履行个人作为一方当事人的合同所必需；
- 为履行法定职责或者法定义务所必需；
- 为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
- 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理个人信息；
- 依照法律法规的规定在合理的范围内处理您自行公开的个人信息，或者其他已经合法公开的个人信息（如合法的新闻报道、政府信息公开等渠道）；
- 法律、行政法规另有规定的情形。

### 1.7.个人信息的匿名化处理

在不公开披露、对外提供您个人信息的前提下，百度公司有权对匿名化处理后的用户数据库进行挖掘、分析和利用（包括商业性使用），有权对产品/服务使用情况进行统计并与公众/第三方共享匿名化处理后的统计信息。

### 1.8.其他您另行同意的目的

当我们要将信息用于本隐私政策未载明的其它用途时，会事先征求您的同意。当我们要将基于特定目的收集而来的信息用于其

他目的时，会事先征求您的同意。

## 2.我们如何共享、转让、公开披露、委托处理您的个人信息以及所接入的第三方

我们的某些服务将由授权合作伙伴提供，仅为实现本隐私政策中声明的目的，我们可能会与合作伙伴共享您的某些个人信息，以提供更好的客户服务和用户体验。我们仅会出于合法、正当、必要、特定、明确的目的共享您的个人信息，并且只会共享提供服务所必要的个人信息。除另行获得您的授权或法律法规另有要求，我们的合作伙伴无权将共享的个人信息用于任何其他用途。

**2.1.共享** 除非经过您本人事先单独同意或符合其他法律法规规定的情形，我们不会向除百度公司以外的第三方共享您的个人信息，但经过处理无法识别特定个人且不能复原的除外。

对我们与之共享个人信息的公司、组织和个人，我们会对其数据安全环境进行调查，与其签署严格的保密协定，要求他们按照依法采取保密和安全措施来处理个人信息。

在下列情况下，我们可能会与以下第三方共享您的个人信息：

### 2.1.1.关联公司或同一账号体系使用方

我们与我们的关联公司或同一账号体系使用方之间具备高度一致的数据安全保障措施和管理水平。为了共同向您提供服务、提高您使用产品的安全性和便利性、保障一致性的服务体验，在遵循法律要求的前提下，我们会在关联公司或同一账号体系使用方之间共享您的个人信息。在此情况下，我们只会共享必要的个人信息，如果我们共享您的敏感个人信息或接收方改变个人信息的使用及处理目的，将再次征求您的授权同意。

对于同一账号体系使用方，为了履行监管要求、保障账号安全性、进行一致化的展示以及与您取得联系，基于您的有效授权，我们可能会共享您的账号信息。

### 2.2.转让

我们不会将您的个人信息转让给除百度公司外的任何公司、组织和个人，但以下情况除外：

- 事先获得您的明确授权或同意；
- 满足法律法规、法律程序的要求或强制性的政府要求或司法裁定；
- 如果我们或我们的关联公司涉及合并、分立、清算、资产或业务的收购或出售等交易，您的个人信息有可能作为此类交易的一部分而被转移，我们将确保该等信息在转移时的机密性、向您告知接收方的名称或姓名和联系方式，并尽最大可能确保新的持有您个人信息的公司、组织继续受此隐私政策的约束，否则我们将要求该公司、组织重新向您征求授权同意。

### 2.3.公开披露

我们仅会在以下情况下，公开披露您的个人信息：

- 获得您的单独同意；
- 基于法律法规、法律程序、诉讼或政府主管部门强制性要求下。
- 对违规账号、违规行为进行处罚公告时，我们会披露相关账号的信息，这些信息不可避免包含账号相关认证信息及违规情况。

### 2.4.共享、转让、公开披露个人信息时事先征得授权同意的例外

在以下情形中，共享、转让、公开披露个人信息无需事先征得个人信息主体的授权同意：

- 为订立、履行个人作为一方当事人的合同所必需；
- 为履行法定职责或者法定义务所必需；
- 为应对突发公共卫生事件，或者紧急情况下为保护自然人的生命健康和财产安全所必需；
- 为公共利益实施新闻报道、舆论监督等行为，在合理的范围内处理个人信息；

- 依照法律法规的规定在合理的范围内处理您自行公开的个人信息，或者其他已经合法公开的个人信息（如合法的新闻报道、政府信息公开等渠道）；
- 法律、行政法规另有规定的情形。

## 2.5 委托处理与共同处理

为了更好地为您提供服务，我们可能委托第三方合作伙伴处理您的个人信息。我们仅会出于合法、正当、必要、特定、明确的目的委托第三方合作伙伴处理您的信息。我们会严格限制受托方的信息处理目的并要求受托方采取充分的数据安全保障手段。如果第三方合作伙伴将您的信息用于我们未委托的用途，其将另行征得您的同意。我们委托的授权合作伙伴包括广告、分析服务类的授权合作伙伴。我们会在遵守行业通行的安全技术前提下，委托这些合作伙伴处理与广告投放、广告效果、广告触达率有关的信息。未经您有效授权，我们不会将可以直接识别您个人身份的信息委托第三方合作伙伴处理。如您主动为特定第三方合作伙伴留下您的联系方式、地址信息等个人信息以供其与您取得联系，其将直接处理您的个人信息。

在共同处理场景下，我们同样会按照法律法规要求与合作方签署符合法律法规要求的相关协议，并履行相关个人信息保护义务。

## 3. 我们如何保存及保护您的个人信息

### 3.1. 保存期限

我们将在提供服务所需的期限内保存您的个人信息，但法律法规对保存期限另有规定、您同意留存更长的期限、保证服务的安全与质量、实现争议解决目的、技术上难以实现等情况下，在前述保存期限到期后，我们将依法、依约或在合理范围内延长保存期限。

在超出保存期限后，我们将根据法律规定删除您的个人信息或进行匿名化处理。

### 3.2. 保存地域

原则上，我们在中华人民共和国境内收集和产生的个人信息，将存储在中华人民共和国境内。在以下情形下，您的个人信息可能会被转移到您使用产品或服务所在国家/地区的境外管辖区，或者受到来自这些管辖区的访问：

- 1、适用的法律法规有明确规定；
- 2、获得您的单独同意；
- 3、为了履行与您订立的合同。

针对以上情形，我们会严格履行法律法规规定的义务，并按照本政策保护您的个人信息安全。

### 3.3. 安全措施

- 我们会以“最小化”原则收集、使用、存储和传输用户信息，并通过用户协议和隐私政策告知您相关信息的使用目的和范围。
- 我们非常重视信息安全。我们成立了专责团队负责研发和应用多种安全技术和程序等，我们会对安全管理负责人和关键安全岗位的人员进行安全背景审查，我们建立了完善的信息安全管理制度和内部安全事件处置机制等。我们会采取适当的符合业界标准的安全措施和技术手段存储和保护您的个人信息，以防止您的信息丢失、遭到被未经授权的访问、公开披露、使用、毁损、丢失或泄漏。我们会采取一切合理可行的措施，保护您的个人信息。我们会使用加密技术确保数据的保密性；我们会使用受信赖的保护机制防止数据遭到恶意攻击。
- 我们会对员工进行数据安全的意识培养和安全能力的培训和考核，加强员工对于保护个人信息重要性的认识。我们会对处理个人信息的员工进行身份认证及权限控制，并会与接触您个人信息的员工、合作伙伴签署保密协议，明确岗位职责及行为准则，确保只有授权人员才可访问个人信息。若有违反保密协议的行为，会被立即终止与百度公司的合作关系，并会被追究相关法律责任，对接触个人信息人员在离岗时也提出了保密要求。
- 我们提醒您注意，互联网并非绝对安全的环境，当您通过Baidu Comate中嵌入的第三方社交软件等与其他用户交互您的地理位置或行踪轨迹信息时，不确定第三方软件对信息的传递是否完全加密，注意确保您个人信息的安全。
- 我们也请您理解，在互联网行业由于技术的限制和飞速发展以及可能存在的各种恶意攻击手段，即便我们竭尽所能加强安全

措施，也不可能始终保证信息的百分之百安全。请您了解，您使用我们的产品和/或服务时所用的系统和通讯网络，有可能在我们控制之外的其他环节而出现安全问题。

- 根据我们的安全管理制度，个人信息泄露、毁损或丢失事件被列为最特大安全事件，一旦发生将启动公司最高级别的紧急预案，由安全部、公共事务部、法务部等多个部门组成联合应急响应小组处理。

### 3.4.安全事件通知

- 我们会制定网络安全事件应急预案，及时处置系统漏洞、计算机病毒、网络攻击、网络侵入等安全风险，在发生危害网络安全的事件时，我们会立即启动应急预案，采取相应的补救措施，并按照规定向有关主管部门报告。
- 个人信息泄露、毁损、丢失属于公司级特大安全事件，我们会负责定期组织工作组成员进行安全预案演练，防止此类安全事件发生。若一旦不幸发生，我们将按照最高优先级启动应急预案，组成紧急应急小组，在最短时间内追溯原因并减少损失。
- 在不幸发生个人信息安全事件后，我们将按照法律法规的要求，及时向您告知安全事件的基本情况和可能的影响、我们已采取或将要采取的处理措施、您可自主防范和降低的风险的建议、对您的补救措施等。我们将及时将事件相关情况以站内通知、短信通知、电话、邮件等您预留的联系方式告知您，难以逐一告知时我们会采取合理、有效的方式发布公告。同时，我们还将按照监管部门要求，主动上报个人信息安全事件的处置情况。请您理解，根据法律法规的规定，如果我们采取的措施能够有效避免信息泄露、篡改、丢失造成危害的，除非监管部门要求向您通知，我们可以选择不向您通知该个人信息安全事件。

## 4.您的权利

按照中国相关的法律、法规、标准，以及其他国家、地区的通行做法，我们保障您对自己的个人信息行使以下权利，如您对权利行使有疑问或主张，您也可以通过【7.如何联系我们】中的方式与我们取得联系。

### 4.1.查阅权

原则上您可以通过如下方式查阅您的个人信息：

- 账号信息：您可以随时登录passport.baidu.com，访问或编辑您的账号中的个人资料信息、更改您的密码、进行账号关联或身份认证等。

### 4.2.更正及补充权

当您发现我们处理的关于您的个人信息有错误或者不完整时，经对您的身份进行验证，且更正不影响信息的客观性和准确性的情况下，您有权对错误或不完整的信息作出更正或补充，您可以自行登录passport.baidu.com进行更正或补充，或通过反馈与报错等方式将您的更正或补充申请提交给我们。

### 4.3.删除权

在以下情形中，您可以向我们提出删除个人信息的请求：

- 如果我们违反法律法规或与您的约定处理您的个人信息；
- 如果我们的处理目的已实现、无法实现或者为实现处理目的不再必要；
- 如果我们停止提供产品或者服务，或者保存期限已届满；
- 如果您撤回同意；
- 法律、行政法规规定的其他情形。

当您从我们的服务中删除信息后，我们可能不会立即在备份系统中删除相应的信息，但会在备份更新时删除这些信息。请您知晓并理解，如果法律、行政法规规定的或本隐私政策说明的保存期限未届满，或者删除个人信息从技术上难以实现的，我们会停止除存储和采取必要的安全保护措施之外的处理。

### 4.4.撤回同意

每个业务功能需要一些基本的个人信息才能得以完成。对于额外收集的个人信息的收集和使用，您可以随时给予或收回您的授

权同意。

- 您可以登录passport.baidu.com，解除百度账号与第三方账号的绑定关系，也可以通过Baidu Comate的企业管理来关闭部分功能选项从而撤销部分授权。
- 当您撤回同意后，我们无法继续为您提供撤回同意所对应的服务，也将不再使用您相应的个人信息。但您撤回同意的决定，不会影响此前基于您的同意而开展的个人信息处理。

#### 4.5. 注销权

- 您可注销此前注册的账号。您可以登录passport.baidu.com，注销您的百度账号。
- \* 一旦您注销百度账号，将无法使用百度公司全线用户产品的服务，因此请您谨慎操作。我们为了保护您或他人的合法权益会结合您对百度公司各产品的使用情况判断是否支持您的注销请求。例如若您有未提现的稿费，或在百度糯米里有未消费的糯米储值卡，或您的网盘中还存有资料，则我们不会立即支持您的请求。除法律法规另有规定外，注销账号之后我们将停止为您提供产品和服务，并根据您的要求删除您的个人信息。
- \* 您通过第三方账号（如微信、微博、QQ、Github、Gitee）授权登录Baidu Comate时，需要向第三方申请注销账号。

#### 4.6. 复制权及可携带权

您有权复制我们收集的您的个人信息。在法律法规规定的条件下，同时符合国家网信部门规定指令和条件的，如果技术可行，您也可以要求我们将您的个人信息转移至您指定的其他主体。

您可以登录passport.baidu.com或通过本政策【“7. 如何联系我们”】与我们联系并获取您的个人信息资料副本。为保证信息安全，我们将会对您的身份进行必要的验证。

#### 4.7. 提前获知产品和服务停止运营权。

Baidu Comate愿一直陪伴您，若因特殊原因导致Baidu Comate产品停止运营，我们将在合理期间内在产品或服务的主页面或站内信或向您发送电子邮件或其他合适的能触达您的方式通知您，并将停止对您个人信息的收集，同时会按照法律规定对已收集的您的个人信息进行删除或匿名化处理。

#### 4.8. 获得解释的权利

您有权要求我们就个人信息处理规则作出解释说明。您可以通过第八部分中的联系方式与我们联系。

#### 4.9. 权利行使的例外

在以下情形中我们可能无法响应您的查阅、更正及补充、删除、复制及可携带、知情、决定的请求：

- 与国家安全、国防安全直接相关的；
- 与公共安全、公共卫生、重大公共利益直接相关的；
- 与犯罪侦查、起诉、审判和执行判决等直接相关的；
- 我们有充分证据表明您存在主观恶意或滥用权利的（如您的请求将危害公共安全和他人合法权益，或您的请求超出了一般技术手段和商业成本可覆盖的范围）；
- 响应个人信息主体的请求将导致您或其他个人、组织的合法权益受到严重损害的；
- 涉及商业秘密的；
- 法律、行政法规另有规定的情形。

#### 4.10. 如何响应您的权利请求

- 为保障安全，您可能需要提供书面请求，或以其他方式证明您的身份。我们可能会先要求您验证自己的身份，然后再处理您的请求。
- 对于您合理的请求，我们原则上不收取费用，但对多次重复、超出合理限度的请求，我们将视情收取一定成本费用。对于那

些无端重复、需要过多技术手段（例如，需要开发新系统或从根本上改变现行惯例）、给他人合法权益带来风险或者非常不切实际（例如，涉及备份磁带上存放的信息）的请求，我们可能会予以拒绝。

## 5.我们如何处理未成年人的个人信息 百度公司非常重视对未成年人信息的保护。

我们的产品和服务主要面向成年人。我们只会在受到法律法规允许、父母或其他监护人同意的情况下处理未成年人的个人信息。

如您是未满14周岁的未成年人，在使用Baidu Comate产品和服务前，应请您的父母或其他监护人仔细阅读，或在您的父母或其他监护人的监护、指导下共同仔细阅读本隐私政策以及《[百度儿童个人信息保护声明](#)》，并在征得您的父母或其他监护人同意的前提下使用我们的产品和服务，或向我们提供信息。

如果我们发现在未事先获得可证实的父母或其他监护人同意的情况下收集了未满14周岁的未成年人的个人信息，我们将会采取措施尽快删除相关信息。如果任何时候监护人有理由相信我们在未获监护人同意的情况下收集了未成年人的个人信息，请通过Baidu Comate中的意见反馈联系我们，我们会采取措施尽快删除相关数据。

## 6、隐私政策的修订

我们的隐私政策可能变更。

未经您明确同意，我们不会削减您按照本隐私政策所应享有的权利。我们会在本页面上发布对本隐私政策所做的任何变更。

对于重大变更，我们会在产品或服务的主要曝光页面或站内信或向您发送电子邮件或其他合适的能触达您的方式通知您。若您不同意该等变更，请您立即停止使用Baidu Comate产品和服务。

本政策所指的重大变更包括但不限于：

- 我们的服务模式发生重大变化。如处理个人信息的目的、处理的个人信息类型、个人信息的使用方式等；
- 我们在所有权结构、组织架构等方面发生重大变化。如业务调整、破产并购等引起的所有者变更等；
- 个人信息对外提供、转让或公开披露的主要对象发生变化；
- 您参与个人信息处理方面的权利及其行使方式发生重大变化；
- 我们负责处理个人信息安全的责任部门、联络方式及投诉渠道发生变化时；
- 个人信息安全影响评估报告表明存在高风险时。

如有本隐私政策未尽事宜，以《[百度隐私权保护声明](#)》为准。

## 7.如何联系我们

Baidu Comate的成长离不开各方用户达人的共同努力，我们非常感谢您对Baidu Comate数据更新、使用反馈方面的贡献。

您可以通过comate@baidu.com反馈您对Baidu Comate的建议以及在使用过程中遇到的问题，以帮助我们优化产品功能及服务，使更多用户更加便捷的使用我们的产品和服务。

您可以通过个人信息保护问题反馈平台（<http://help.baidu.com/personalinformation>）同我们联系。

您也可以通过如下联络方式同我们联系：

中国北京市海淀区上地十街10号

北京百度网讯科技有限公司法务部

邮政编码：100085

为保障我们高效处理您的问题并及时向您反馈，需要您提交身份证明、有效联系方式和书面请求及相关证据，我们会在验证您的身份后处理您的请求。

如果您对我们的回复不满意，特别是您认为我们的个人信息处理行为损害了您的合法权益，您还可以通过以下外部途径寻求解决方案：向北京市海淀区人民法院提起诉讼。

## 附：名词解释

**个人信息**是指以电子或者其他方式记录的与已识别或者可识别的自然人有关的各种信息，不包括匿名化处理后的信息。个人信息包括姓名、性别、出生日期、身份证件号码、个人生物识别信息、住址、通信通讯联系方式、通信记录和内容、账号密码、财产信息、征信信息、行踪轨迹、住宿信息、健康生理信息、交易信息等。

**敏感个人信息**是指一旦泄露或者非法使用，容易导致自然人的人格尊严受到侵害或者人身、财产安全受到危害的个人信息，包括生物识别、宗教信仰、特定身份、医疗健康、金融账号、行踪轨迹等信息，以及不满十四周岁未成年人的个人信息。

**设备**是指可用于使用百度产品和/或服务的装置，例如桌面设备、平板电脑或智能手机。

**设备信息**是指您正在使用的硬件信息、操作系统信息等。

**设备连接信息**是指我们的服务器所自动记录您在访问Baidu Comate时所发出的请求软件版本号、使用的语言、IP地址等。

**日志信息**是指我们的服务器所自动记录您在访问Baidu Comate时所发出的请求，例如您的IP 地址、浏览器的类型和使用的语言、硬件设备信息、操作系统的版本、网络运营商的信息、您访问服务的日期、时间、时长等您在使用我们的产品或服务时提供、形成或留存的信息。

**Cookie**是指支持服务器端（或者脚本）在客户端上存储和检索信息的一种机制，通过增加简单、持续的客户端状态来扩展基于Web的客户端/服务器应用。服务器在向客户端返回HTTP对象的同时发送一条状态信息，并由客户端保存。状态信息中说明了该状态下有效的URL范围。此后，客户端发起的该范围内的HTTP请求都将把该状态信息的当前值从客户端返回给服务器，这个状态信息被称为Cookie。

**去标识化**是指个人信息经过处理，使其在不借助额外信息的情况下无法识别特定自然人的过程。

**匿名化**是指通过对个人信息的技术处理，使得个人信息主体无法被识别，且处理后的信息不能被复原的过程。个人信息经匿名化处理后所得的信息不属于个人信息。

**百度平台**是指百度公司旗下各专门频道或平台服务（包括百度搜索、百度APP及衍生版、百度百科、百度知道、百度贴吧、百度手机助手及其他百度系产品<https://www.baidu.com/more/>）等网站、程序、服务、工具及客户端。

**关联公司**是指Baidu Comate的经营者北京百度网讯科技有限公司及其他与百度公司存在关联关系的公司的单称或合称。“关联关系”是指对于任何主体（包括个人、公司、合伙企业、组织或其他任何实体）而言，即其直接或间接控制的主体，或直接或间接控制其的主体，或直接或间接与其受同一主体控制的主体。前述“控制”指，通过持有表决权、合约或其他方式，直接或间接地拥有对相关主体的管理和决策作出指示或责成他人作出指示的权力或事实上构成实际控制的其他关系。

再次感谢您对Baidu Comate的信任和使用！

北京百度网讯科技有限公司

更新日期：2024年6月25日 生效日期：2024年1月8日

## Baidu Comate自动续费服务协议

### 一、服务条款的接受与修改

本协议系由北京百度网讯科技有限公司及其关联公司（以下简称“百度公司”）与所有使用BaiduComate服务的主体（包括但不限于个人、团队等）（以下简称“用户”）对关于BaiduComate自动续费委托扣款服务（下称“本服务”）所订立的协议。本协议描述百度公司与BaiduComate之间关于本服务的使用及相关方面的权利义务。“BaiduComate”是指依据《BaiduComate服务协议》及相关服务、收费规则，享受BaiduComate提供的各类网络增值服务的个人或单一实体，具体以“产品介绍”页面中描述为准。本协议构成BaiduComate用户（无论个人或者单位）使用百度公司所提供的本服务之先决条件，除非BaiduComate用户接受本协议条款，否则BaiduComate用户无权使用本服务，BaiduComate用户选择使用本服务行为将视为同意接受本协议各项条款的约束。

### 二、服务说明

2.1 本服务是出于BaiduComate用户对于自动续费的需求，在您已开通BaiduComate服务的前提下，避免您因疏忽或其他原因

导致未能及时续费造成损失而推出的服务，BaiduComate用户授权百度公司在BaiduComate用户有效期届满前后，在不验证BaiduComate用户账户密码、支付密码、短信校验码等信息的情况下从与BaiduComate用户账号绑定的第三方支付账户（以下统称“账户”）余额中代扣下一个计费周期的费用，计费周期包括但不限于月度、季度等（下同），您可以在购买百度公司提供的BaiduComate服务时，根据平台上显示的计费周期进行选择。该服务实现的前提是您已将其BaiduComate账号与上述账户绑定，且可成功从其上述账户中扣款。

2.2 自动续费具体指，基于2.1的前提，百度公司通过上述账户收取BaiduComate下一计费周期费用的扣费方式，您需保证百度公司可以从上述账户扣款成功，因上述账户中可扣款余额不足导致的续费失败，由您自行承担。如您未自主选择取消本服务，百度公司可能在您账户余额恢复充足时/后，继续为您提供自动续费服务（具体以各第三方支付渠道的规则为准），一旦扣款成功，百度公司将恢复您的BaiduComate的服务和自动续费服务。

2.3 BaiduComate用户在享受本服务时，应受《BaiduComate用户协议》的约束，本协议为《BaiduComate用户协议》补充协议，是其不可分割的组成部分，如本规则与《BaiduComate用户协议》存在冲突的，以本规则为准；您使用本服务即表示您已经同意本服务的条款以及百度公司针对本服务发出的各类公示。

2.4 百度公司向BaiduComate用户提供的本服务均仅限于BaiduComate用户在百度平台或百度指定的第三方平台使用，任何以恶意破解等非法手段将百度公司提供的本服务与百度平台或百度指定的第三方平台分离的行为，均不属于本协议中约定的本服务。由此引起的一切法律后果由行为人负责，百度公司有权依法追究行为人的法律责任。

2.5 您知悉并同意，您使用第三方支付渠道支付费用的，您还将受到该第三方服务条款、使用规则的约束。

### 三、双方的权利和义务

3.1 百度公司通过BaiduComate用户上述账户扣除下一计费周期费用的，应在BaiduComate用户开通自动续费服务后，在每个计费周期到期前24小时自动划扣，并将款项记入会员支付记录，并同时相应延长BaiduComate使用有效期。自动续费状态更新可能会存在显示延时，以实际支付自动续费签约状态为准，如BaiduComate自动续费未在计费周期到期前至少24小时前取消自动续费功能，将会自动续订及扣款。

3.2 如在扣费过程出现差错，百度公司和BaiduComate用户应密切配合查明原因，各自承担己方过错造成的损失；若因双方各自存在不均等过错造成损失的，由双方按过错的程度承担相应责任；因双方共同过错造成损失但无法区分各自过错的，由双方均摊责任。

3.3 百度公司可根据其业务开展或技术升级等情况变更或修改本协议的有关服务内容、规则及条款，百度公司将通过在BaiduComate相关页面公示等方式告知用户相关变更或修改内容，但无义务另行做个别通知。如果BaiduComate用户不同意本协议的修改，可以取消已经获取的本服务并停止使用；如果BaiduComate用户继续使用百度公司提供的服务，则视为BaiduComate用户已经接受本协议的全部修改。

3.4 百度公司可在扣款前通知用户，同时本服务由会员自主选择是否在要求的时间内取消，若BaiduComate用户选择不取消，则视为BaiduComate用户同意百度公司按照一定规则进行扣款尝试，一旦扣款成功，百度公司将为BaiduComate用户开通下一个计费周期的扣费服务。

3.5 若在自动续费扣款时/之前，BaiduComate用户的服务价格发生调整，应以现时有效的价格为准。

### 四、自动续费服务的开通

4.1 如您自BaiduComate相关的PC网页端/移动网页端开通本服务，则视为您授权百度公司有权从您的相关会员服务期限届满前后，从您授权绑定的包括但不限于：微信、支付宝、百度钱包、电信运营商代收、财付通等支付渠道中代扣下一个计费周期服务费，计费周期包括但不限于月度、季度等。

4.2 对于所选择的支付渠道，您有义务确保定期关注并确保其于该支付方式的账号项下有充足的余额用于满足自动续费的需求。如因前述原因瑕疵（包括但不限于：余额不足、余额不足以满足用户选择多个自动续费服务）而致使无法成功自动续费的，相关会员服务将无法继续。

4.3 为了充分保障您的知情权和选择权，我们将在您接受本服务前或者自动续费到期前五日，以短信通知等显著方式提请您注意。

### 五、自动续费服务有效期

5.1 除非百度公司另行指定并经您同意，否则您获得的自动续费服务均为长期有效，直至您自主选择取消自动续费服务。

5.2 百度公司将于您所选择的BaiduComate服务期届满前进行自动续费，如您有变更，请按如下第6条的方式进行变更，否则百度公司有权视您委托百度公司通过您默认的支付方式续取相应的服务费用。

5.3 百度公司特别提醒您：当您选择使用自动续费服务后，通过自动续费服务委托百度公司在您默认的支付方式所支付的费用，百度公司将视您选择认购的BaiduComate相关付费服务。因此，百度公司将不以任何方式对您所支付的费用予以退还。

## 六、自动续费服务的退订

6.1 您可自主选择退订本服务。通过如下解约流程，即可关闭自动续费会员服务：【支付宝自动续费的关闭：支付宝-我的-设置-支付设置-免密支付/自动扣款-选择相关服务-关闭服务】

6.2 自动续费套餐取消成功后，不影响您当期已生效的会员权益，会员权益到期后将不再续费。

6.3 您在选择终止本服务前已经委托百度公司自动续费扣款的指令仍然有效，百度公司对于基于该指令扣除的费用不予退还。

6.4 如您在开通本服务后希望更换自动续费套餐，请先取消本服务，并在选择您希望的自动续费套餐后重新购买本服务。

## 七、协议有效期限、终止

7.1 本协议自BaiduComate用户选择接受或使用本服务后生效，直至BaiduComate用户终止本服务/注销会员资格时终止。

7.2 BaiduComate用户有权随时在会员账号设置中选择终止本服务，终止本服务后，百度公司将停止向会员提供本服务。

7.3 BaiduComate用户在选择终止本服务前已经委托百度公司执行完毕的自动续费扣款的指令仍然有效，百度公司对于基于该指令扣除的费用不予退还，由会员承担相关责任，请您务必在通知提示的时间内退订服务。

## 八、违约责任

8.1 如果一方违约，守约方有权通过双方约定的争议解决方式获得补偿。

8.2 百度公司与BaiduComate用户应通过友好协商解决本协议履行过程中产生的争议，经协商无法解决的，任何一方均有权将争议提交至百度公司所在地有管辖权的人民法院诉讼解决。本协议的解释、效力和执行等有关问题均适用中华人民共和国法律。

北京百度网讯科技有限公司

# 发版记录

## Comate AI IDE 更新日志

### Update 1.7.0 (2026-04-27)

- 全新插件市场：新增插件市场，支持浏览和安装第三方插件，一键扩展开发能力
- 增强 Plan 模式模型选择：Plan 模式支持在 Build 阶段选择不同模型，灵活切换推理引擎
- 增强自定义 Subagent：自定义 Subagent 支持指定模型，可根据任务类型选用最适模型
- 优化异步命令执行：异步 bash 工具优化超时处理逻辑，长时间任务执行更稳定
- 诸多体验优化与问题修复

### Update 1.6.0 (2026-04-20)

- 全新 Hooks 能力：支持在 Agent 流程中配置 Hooks，可在关键节点插入自定义逻辑
- 增强 Skill 管理能力：支持 Skill 的启用/禁用切换、快速编辑、和卸载流程
- 增强 Skills 召回效果：Skills 信息现在会注入到第一条用户消息的 system reminder 中，提升上下文理解准确性

- 优化模型切换能力：修改模型展示开关后，对话输入框中的模型选择列表可实时生效，无需重新加载
- 增强 GLM 思考模型：GLM 思考模型支持新的 `clear_thinking` 参数，优化思维链输出控制
- 增强上下文压缩能力：压缩超限后支持渐进式裁剪重试，Claude/GLM 模型压缩时输出上限调大
- 优化上下文压缩策略：System 自动压缩机制优化，提升任务执行效率

#### Update 1.5.0 (2026-04-13)

- 全新 Skills 广场：新增 Skills 广场入口，便捷查找和管理 Skills
- 增强并行对话状态管理：对话列表增加状态标识，解决短对话没有未读状态的问题，并行任务退出后自动继续
- 新增异步命令执行：run\_command 工具支持异步执行模式，新增 `--background-timeout` 参数设定超时时间，长时间运行的任务不再阻塞 Agent workflow
- 增强 read\_file 工具：支持读取图片文件内容，优化 Excel 读取的行号规则与提示
- 增强 Markdown 图片渲染：对话中 Markdown 内容支持内嵌图片展示，图片 MIME 类型检测改为数据格式识别
- 增强上下文压缩能力：压缩超限后支持渐进式裁剪重试，优化压缩策略过滤有效摘要，上下文管理更智能
- 全新 Auto 模式路由策略：Auto 模式支持智能路由模型选择，根据任务类型自动匹配合适的模型（包括多模态任务自动识别），提升编码效率
- 优化 Slash 菜单：Slash 菜单分类展示重构，新增分类分隔线、图标和系统 Skill 排序
- 优化命令输出展示：命令输出处理 ANSI 特殊字符，显示更清晰
- 优化粘贴代码体验：粘贴包含 # 的内容时不再频繁弹出上下文选择窗口
- 优化 Spec 模式体验：点击 Spec 时不提前打开编辑区，等待文档生成后再展示，避免空白页面
- 优化 Agent 报错提示：完善多场景报错信息，提升问题可读性与透明度
- 优化 LSP 诊断超时：LSP 诊断超时时间增加到 60 秒，减少大型项目误报
- 修复并行任务退出后中断：并行两个任务时退出后没有继续执行的问题已修复
- 修复 find-skill 返回不全：find-skill 工具返回结果不全的问题已修复，新增分页支持

#### Update 1.4.0 (2026-03-30)

- 全新 Mission Mode：任务为中心，轻松管理复杂项目开发
  - 轻松切换多工作区
  - 同一工作区可管理多代码库
  - 全新任务列表
  - 自动匹配场景化研发模式
  - 执行结果清晰展示
- 新增语音输入：输入框支持语音输入，并自动优化开发者的表达，加速 Agent 交互效率
- 新增多 Workspace 支持：全面兼容 VS Code 多根工作区 (Multi-root Workspaces)。Agent 在执行过程中可跨全部工作区进行检索与编辑，并支持在主仓库层级配置与应用规则 (Rules)
- 新增元素编辑快捷模式：元素编辑功能引入快捷模式，并支持国际化
- Figma 设计稿支持 Android 规则：系统级 figma2code Skill 新增 Android 平台默认规则

- 支持多 Figma 设计稿上传：系统级 figma2code Skill 支持同时上传多个设计稿
- 优化 Agent 报错提示：完善多场景报错信息，提升问题可读性与透明度
- 优化 Token 使用效率：采用更精细的 token 估算策略，压缩成功摘要与错误日志，降低单轮输入消耗

#### Update 1.3.0 (2026-03-23)

- 自定义 Command：固化常用 Prompt，兼容社区规范，灵活扩展能力。
- 模型管理入口：新增 Models 设置，便捷管理常用模型。
- 全局 Rules 配置：跨 workspace 生效，统一规则约束。
- 开源 Skill 快捷安装：支持从 GitHub 安装社区 Skill，拓展能力边界。
- 蓝湖设计稿支持：直接发送设计稿，快速启动设计转代码流程。
- F2C Skill 增强：新增 iOS 设计转代码规则，支持 iOS 应用生成。
- AI Browser Use 能力：AI 可直接操作浏览器，支持打开网站、登录账号、点击页面 / 跳转等。

#### Version 1.2.0 (2026-03-17)

- Spec 架构全面升级：切换为新 Agent Loop，提升任务执行稳定性与交互体验
- 图片上传优化：移除像素限制，仅保留尺寸约束，优化异常提示

#### Version 1.1.0 (2026-03-09)

- Plan 模式全新升级：主动提问澄清需求、基于目标规划任务、充分对齐计划再执行，整体执行结果更符合预期
- Explore Subagent 能力增强：深度检索过程新增网页搜索、网页抓取，自动联网查询实时内容，并可调用 Skill 技能与 MCP 工具，主动扩展能力
- 新增知识集专有检索 Subagent：内置 Doc Explore Subagent，通过调用获取知识集目录列表、读取文档、语义检索等工具，精准定位知识集内容
- 多 Tab 对话体验优化：细化首屏新对话 Tab 展示效果，对话过多时支持滚动切换
- 回滚流程优化：重新组织操作回滚后对话上下文，提升重新对话效果稳定性
- 服务稳定性提升：优化降级和重试逻辑，极大减少请求中断情况
- IDE主题色适配：调整默认主题配色使整体风格更统一，在不同界面与组件中呈现更协调的视觉效果

#### Version 1.0.0 (2026-02-28) - Zulu 核心能力升级，Spec 模式正式发布

Spec 模式：先对齐，再执行的新一代开发方式 Spec Mode 是一种先对齐方案、再落地执行的全新开发方式，让复杂代码修改更可控、更高效：

- 对齐方案 (Doc)：AI 先根据你的需求，生成清晰的技术方案与文档，供你审阅、调整，确保方向一致。
- 拆解任务 (Tasks)：确认方案后，AI 自动拆解为可执行的详细任务列表。
- 按计划执行：你可以逐个执行或批量执行任务，随时暂停、回滚或调整。
- 总结复盘：任务完成后，自动生成变更总结，便于回顾与验收。
- Spec信息图

适用场景：多文件修改、复杂功能开发、架构重构等需要严谨规划的开发任务。

#### Agent 核心能力升级

### 核心能力升级

- 推理与执行能力重构：重构任务理解与工具调用链路，大幅提升复杂需求拆解与执行准确性。
- 长对话能力增强：优化对话压缩策略与 Token 管理机制，多轮交互更稳定。
- 全面支持 Skill 机制：兼容社区协议，支持全局与项目级 Skill，支持渐进加载与手动调起。
- 新增 Explore Subagent：专注代码库语义检索，提升代码定位与理解效率。
- MCP 架构重构：支持渐进式工具注册，调用更稳定、Token 消耗更低。
- Web 工具增强：支持网络内容检索与网页解析，实时上下文更丰富。
- 全新 Slash 指令体系：支持「/」快速调用系统能力，内置多种系统级指令。
- 支持多模态输入：按模型能力区分图文混合输入，扩展 AI 使用场景。

### 体验优化

- Agent 输出升级：强化关键信息，减少冗余，阅读效率更高。
- Subagent 展示优化：调整渲染逻辑与视觉层级，信息结构更清晰。
- 输入体验优化：升级输入框交互与视觉，Agent/模型选择更直观。
- 终端操作优化：支持 bash 代码块一键插入终端执行，操作更便捷。
- Mermaid 渲染增强：支持渲染后复制代码，使用更灵活。
- Loading 交互优化：增加随机 loading 文本提示，交互体验更丰富。

### v0.13.0 (2026-02-11)

#### Spec与编辑体验双优化

- Spec 任务执行可控性提升：支持执行一部分任务，任务的暂停、回滚，可灵活调整AI任务执行流程。
- 网页元素可视化编辑上线：选中页面元素可直接编辑，实时预览修改效果，支持多次撤销回退。
- AI侧边栏适配与视图优化：兼容新版IDE内核，右侧导航区域可调整；优化Spec Changes视图，提升diff展示效果。
- 远程开发环境兼容 Ubuntu 18：在远程开发场景下支持 Paddle 于 Ubuntu 18 系统运行，确保旧版本环境可正常使用
- 元素编辑支持撤销回退：在元素级编辑中支持撤销与回退操作，所有编辑变更可统一应用与恢复
- 提升安装阶段配置导入兼容性：在导入其他 IDE（含 Cursor）配置时自动处理快捷键冲突，确保插件功能可正常使用

### v0.12.0 (2026-01-16)

#### 语义检索全量开放，多维度体验升级

- 上线马年皮肤：祝您马到成功
- 语义检索全量开放：覆盖更多检索场景，显著提升结果相关性与命中率。
- Spec 优化：支持 Doc、Task 添加至对话，CodeLens 新增颜色标识，提升展示效果与二次编辑体验。
- 多任务并行执行：支持并行开启多个对话窗口，同时处理代码理解、需求开发、功能测试等任务，提升开发效率。
- 对话 Token 用量可视化：直观查看 Token 消耗情况，便于主动管理上下文，提升对话稳定性与生成效果。
- 支持 AGENTS.md 文件：兼容社区规范文件，可选择 Rules 或 AGENTS.md 为 Agent 提供上下文说明与行为约束。
- 手动触发上下文压缩：自主控制对话上下文规模，优化长对话场景下的稳定性与生成效果。

- 历史消息交互升级：采用轻量浮层展示，历史任务浏览与对话回溯更高效。
- 提示词优化：协助优化你的输入，补充更多信息，生成更精准

#### v0.11.0 (2026-01-05)

上新Spec列表

- 上新 Spec 列表：新增“ Spec 列表”，支持便捷管理各类 Spec 会话。
- 实验室功能上线：设置页新增“加入实验室”开关，开启后可提前体验 beta 版本功能。

#### v0.10.0 (2025-12-22)

Agent 细粒度回滚上线，个性化侧边栏 + 青松圣诞主题，邀你体验！

- Agent 回滚机制升级：支持更细粒度回滚，可回退至每一次提示词执行前、每一次工具调用前的代码状态，提升代码修改的可控性与安全性。
- 个性化支持：新增侧边栏图片更换能力，支持自定义展示风格，打造更具个性的使用体验。
- 青松圣诞主题上线：新增圣诞氛围主题，快来体验！
- Spec 交互优化：Spec模式下，从右侧单文件改动跳转时直接定位对应文件，减少上下文切换，提升编辑效率。

#### v0.9.0 (2025-12-16)

发布 Page Builder 页面设计智能体

- Page Builder 智能体上线：新增专精于页面设计的智能体，清晰描述需求即可产出好看好用的页面。
- 页面元素提问增强：针对选中页面元素提问的场景进行了优化，准确性有所提升。

#### v0.8.0 (2025-12-2)

检索工具升级，更智能更流畅

- 上下文管理：统一添加上下文方式，支持 + 号/# 号双指令。
- 粉熊皮肤：上线暖暖粉熊、热烈粉熊两套皮肤。
- MCP 能力：全局通用：支持创建跨 Workspace 全局 MCP，减少重复安装。
- 行间命令：调优建议升级：将「调优建议」指令升级为由 Zulu Agent 提供回答，更贴近上下文，代码融合更智能。
- 代码补全：样式优化，使代码修改差异更清晰可见。
- 体验优化：多处细节完善。

#### v0.7.0 (2025-11-25)

Spec 模式上线

- Spec 模式：上线 Spec 模式：先对齐、再执行，高效高质量开发。通过在执行前对齐需求与实现路径，让 AI 的计划与步骤清晰呈现、可被审阅与调整，减少偏差、错误和遗漏，实现更稳定、更可信的开发过程。
- 体验优化：已知问题修复。

#### v0.6.0 (2025-11-20)

插件市场完善，护眼体验升级

- 插件市场：补充常用插件，同时优化插件市场稳定性与速度。

- 网页预览: 升级网页预览调试功能, 优化交互逻辑与响应效率。
- 主题: 新增两种护眼主题皮肤, 适配长时间开发场景。

#### v0.5.0 (2025-11-3)

内核升级, 移动端调试新上线

- Code OSS 内核版本升级: 同步至 1.105.0 版本, 提升编辑器启动、代码加载及插件运行流畅度。
- 页面预览升级: 新增移动端预览调试功能, 支持模拟移动端运行场景。
- 已知问题修复: 提升 IDE 运行稳定性与使用体验。

#### v0.4.0 (2025-10-13)

Agent First 形态全新升级、多功能体验优化

- 全新升级为 Agent First 形态: Chat 与 Zulu 整合为统一 Agent 形式, 提供丰富官方 Agent 资源, 深度适配问答、规划、编码等核心场景; 支持便捷切换与对话内上下文共享, 复杂任务处理效率显著提升。
- CodeLens 指令全面升级: 实现最佳 Agent 自动切换机制, 生成效果更精准; 新增 CodeLens 可配置功能, 灵活性进一步增强。
- Agent 文档检索能力: 支持在上下文类型中选择知识, Agent 会自动检索相关文档内容辅助回答, 生成质量大幅提升。
- MCP 功能优化: 新增 MCP 自动启用机制, 添加后将所有官方 Agent 默认开启, 操作流程更便捷。
- 新增画布列表功能: 支持多画布快速切换, 新增收藏与搜索功能, 画布管理效率全面提升。

#### v0.3.0 (2025-9-15)

网页预览升级、Agent 全面增强

- 网页预览升级: 网页预览支持选择元素快速修改指令, 修改更快捷。
- 网页预览支持 Figma: 网页预览融合 Figma 生成代码, 现已支持打开 Figma 链接后点选元素添加到对话, 体验更流畅。
- Agent 上下文理解优化: 策略升级, 让系统在处理多段内容时更智能、更连贯, 提高响应的准确性和相关性。
- Agent 工具安全增强: 防止对敏感文件的非法写入与修改, 提升系统安全性。
- Agent 视觉优化: Agent 检索工具视觉优化。移除卡片样式, 界面更简洁直观, 提升使用体验。
- 高对比度主题优化: 持续优化主流主题展示效果, 提升美观度与可用性。

#### v0.2.0 (2025-9-8)

Subagent、多智能体协同

- 智能体: 支持 Subagent, 多智能体协同。主智能体可调度子智能体协同工作, 自动完成任务拆解、分工与执行, 显著提升复杂任务处理能力。

## VSCoDe端插件更新日志

### 大版本升级: v4.0.0 (2026-02-25)

- 推理与执行能力重构: 重构任务理解与工具调用链路, 提升复杂需求拆解与执行准确性
- 长对话能力增强: 优化对话压缩策略与 Token 管理机制, 多轮交互更稳定
- 全面支持 Skill 机制: 兼容社区协议, 支持全局与项目级 Skill, 支持渐进加载与手动触发

- 新增 Explore Subagent：专注代码库语义检索，提升代码定位与理解效率
- MCP 架构重构：支持渐进式工具注册与调用，稳定性与 Token 使用效率提升
- Web 工具增强：支持网络内容检索与网页解析，提供更具时效性的上下文信息
- 全新 Slash 指令体系：输入「/」快速调用系统能力，内置多种系统级指令
- 支持多模态输入：按模型能力区分图文混合输入，扩展 AI 使用场景
- Agent 输出样式升级：强化关键信息表达，减少冗余内容，提升阅读效率
- Subagent 展示优化：简化视觉结构与渲染逻辑，层级关系更清晰
- 输入框交互重构：优化 Agent 与模型选择方式，整体输入体验更流畅
- 终端操作优化：支持将 bash 代码块一键插入终端并执行
- Mermaid 渲染增强：支持渲染后复制源码，便于二次编辑
- Loading 文案优化：增加随机提示文本，提升等待过程的可感知性
- 任务执行可控性增强：支持任务回滚与 Subtask 异常自动重建，提升复杂流程稳定性
- 长连接稳定性优化：优化 SSE 超时策略，支持长时间连续对话
- 消息处理性能提升：提升查询与响应效率，整体交互更流畅
- 文件编辑持久化修复：修复编辑未落盘问题，确保文件变更即时保存

#### v3.33.0 (2026-02-09)

- 修复已知问题。

#### v3.31.0 (2026-01-26)

- 提示词优化：协助优化你的输入，补充更多信息，生成更精准。
- 超级补全展示效果优化：改写建议在采纳前以更清晰的视觉标识突出显示，采纳后自动回归原生代码样式，减少视觉干扰。

#### v3.24.1 (2026-01-22)

- 提示词优化：协助优化你的输入，补充更多信息，生成更精准
- 超级补全展示效果优化：改写建议在采纳前以更清晰的视觉标识突出显示，采纳后自动回归原生代码样式，减少视觉干扰

#### v3.24.0 (2026-01-13)

- 语义检索全面上线：语义检索能力已全量开放，覆盖更多检索场景，显著提升结果相关性与命中率。
- 支持并行执行多任务：并行开启多个对话窗口，同时处理代码理解、新需求实现与已有功能测试等不同任务，显著提升整体开发效率。
- 对话 Token 用量可视化：支持直观查看对话的 Token 消耗情况，通过主动管理上下文以更好适配模型的上下文窗口，提升对话稳定性与生成效果。
- 支持 AGENTS.md：现已支持符合社区规范的 AGENTS.md 文件，您可根据项目习惯选择使用 Rules 或 AGENTS.md，为 Agent 提供统一的上下文说明与行为约束。
- 支持手动触发上下文压缩：可自主控制对话中的上下文规模，提升长对话场景下的稳定性与生成效果。
- 历史消息交互升级：采用更轻量的浮层展示方式，历史任务的浏览与对话回溯更加高效便捷。
- 超级代码补全体验升级：通过分布式推荐显著提升首次展示内容的准确率，优化生成内容与原代码的 Diff 展示，更清晰且不

打扰。

### v3.23.0 (2025-12-22)

- Checkpoint 机制升级：支持更细粒度的回滚能力，可回退至每一次提示词执行前或每一次工具调用前的代码状态，显著提升代码修改过程的可控性与安全性。
- 超级代码补全 Diff 样式全面焕新：升级新增、修改、删除等场景下生成代码的 Diff 展示方式，让代码变更一目了然，确保审阅时不再遗漏细节或误采纳代码，提升决策精准度与视觉舒适度。
- Agent 工作区创建行为优化：在未打开工作区的情况下提问时，将统一使用默认工作区并自动创建独立的 Demo 目录，既保证使用流程顺畅，又有效避免不必要的目录与窗口干扰。

### v3.22.0 (2025-12-08)

- 全新的项目级Memory：在同一项目内可持续保留上下文与关键偏好信息，并随使用自动更新进化，使智能体更好地理解项目规则、适应用户习惯，提供始终一致且更贴合项目语境的生成体验。
- Agent 效果优化：升级系统提示词，降低因模型幻觉导致的文件变更失败问题，生成结果更稳健。
- 上下文理解增强：当前文件与选中代码将默认纳入上下文，并由模型自动判断其与对话的相关性，上下文理解更智能。
- 知识集上下文可指定单文档：支持在指定的自定义文档集合下，对特定自定义文档进行精准检索，进一步提升查询的可控性与结果相关性。
- 知识集检索结果支持跳转：检索结果中的知识来源地址支持一键跳转查看，提升溯源效率与知识查阅体验。
- 快捷指令支持非工作区代码：现已支持在工作区外文件与未保存代码使用快捷指令，使用场景更灵活。
- 侧边栏生成的代码支持快速融合：生成的代码现可一键插入至光标位置，编辑更高效。
- 对话操作升级：支持对话复制、以及对话效果反馈，便于快速复用内容、反馈建议。
- 对话体验优化：模型生成过程中也可编辑输入框内容，操作更连贯、人机交互过程更自然。
- 输入体验优化：粘贴代码后可自动识别并展示来源文件及行号，代码定位更高效。
- 修复其他已知问题。

### v3.21.0 (2025-12-01)

- 上下文检索逻辑升级：重构检索重排逻辑，提升代码库检索结果排序准确性与相关度。
- 上下文管理优化：统一对话输入框中添加上下文的操作方式，同时支持使用 + 号和 # 号，提升操作一致性与易用性。
- 代码库索引扩展：现已支持 Kotlin 文件的向量化处理，进一步提升代码理解与检索效果。
- MCP 能力升级：支持创建全局 MCP，可在多个 Workspace 间通用，降低重复安装成本，使用更高效便捷。
- 快捷指令优化：将「优化建议」指令升级为由 Zulu Agent 提供回答，更贴近上下文，代码融合更智能。
- GBK 文件兼容性增强：新增对 GBK 编码文件的注释与文件对比中的中文显示支持，提升编码文件编辑与对比体验。
- Agent 工具优化：改进读文件工具在处理失败时的容错机制，确保对话流程不受影响；优化写文件工具手动放弃操作，支持同时删除文件及其所在目录，提高操作的可靠性与一致性。
- 其他诸多交互与体验改进。

### v3.20.0 (2025-11-10)

- 全新 Agent 市场：内置丰富的垂类 Agent，可快速使用；同时提供统一设置页面，方便集中管理与展示。
- Agent 支持项目级共享：支持在项目范围内便捷共享自定义 Agent，方便团队复用经验与落地流程。

- Agent 支持基于 AI 创建：简化 Agent 生成流程，提升使用便捷性与效率。
- Rules 支持自动创建：支持基于 AI 自动创建规则，简化配置方式、提高落地效率。
- 单对话支持自动压缩：单次会话现支持上下文自动压缩，有效提升长对话场景下上下文的连续性。
- 对话输入框全面升级：全面优化上下文选择与状态展示，新增上下文推荐与检索能力，并进行了多项样式与细节调整，带来更智能、更顺畅的输入体验。
- MCP 效果提升：调整解析逻辑、增强模型输出容错能力，提升 MCP 输出内容有效性。
- Agent 工具优化：优化执行命令时的操作文案，采用更简洁直观的表达方式，提升交互体验。

### v3.19.0 (2025-10-27)

- 上下文检索全面升级：支持全新检索方式，检索速度更快，兼容更多文件类型与更大规模索引；同时新增可视化索引管理，检索体验更高效直观。
- Agent 支持语义检索：新增基于用户意图的语义检索能力，使 Agent 更精准理解需求，提升生成结果的质量与可靠性。
- Agent 上下文扩展：现支持将 DOCX 与 XLSX 文件作为上下文使用，提升模型对文档内容的理解与处理能力。
- Agent 工具执行效率提升：现已支持并行读取文件，大幅提升 Agent 的生成速度与效率。
- Agent 效果优化：修复了循环调用工具及过度主动触发的问题，提升 Zulu 的调用稳定性与响应准确性。
- MCP 入口优化：将 MCP 入口前置，用户可更快速访问与使用 MCP 功能，提升操作便捷性。
- MCP 功能优化：在添加 Server 时新增必填信息提示，有效提高安装和配置的正确率。
- 网络配置优化：修复了插件代理相关问题，提升代理网络下使用的稳定性与兼容性。
- 其他诸多体验优化和问题修复。

### v3.18.0 (2025-10-13)

- 全新升级为 Agent First 形态：Chat 与 Zulu 合并为统一 Agent 形式，提供丰富官方 Agent，深度支持问答、规划、编码等核心场景，同时支持便捷切换与对话内上下文共享，提升复杂任务处理效率。
- CodeLens 指令全面升级：自动切换最佳 Agent 回答，生成效果更精准。
- Agent 支持文档检索：在上下文类型中选择知识，Agent 在回答过程中会自动检索相关文档内容，显著提升生成质量。
- MCP 升级：现已支持在添加 MCP 后自动为所有官方 Agent 启用，使用更加便捷。
- 改写功能增强：现在即使未打开目标文件，也可支持跨文件跳转，提升多文件改写体验。
- 改写效果增强：独立维护多 Workspace 的编辑历史和诊断信息，使改写结果更精准可靠。
- 对话输入框体验优化：集中优化了输入框的交互体验，提升输入流畅度与整体操作舒适性。

### v3.17.0 (2025-09-15)

- Agent 检索工具视觉优化。移除卡片样式，界面更简洁直观，提升使用体验。
- Agent 上下文理解优化。策略升级，让系统在处理多段内容时更智能、更连贯，提高响应的准确性和相关性。
- Agent 工具安全增强。防止对敏感文件的非法写入与修改，提升系统安全性。
- Agent 性能优化。改进内存管理机制，降低超大文件读取时的内存占用，提升稳定性与性能。

### v3.16.0 (2025-09-08)

- 支持 Subagent，多智能体协同。主智能体可调度子智能体协同工作，自动完成任务拆解、分工与执行，显著提升复杂任务

处理能力。

### v3.15.0 (2025-09-01)

- Zulu 重大升级，支持自定义智能体。支持可根据项目需求创建专属智能体，自定义能力与行为，打造深度契合个人习惯或团队流程的智能体。
- 新增 Plan Agent。支持交互式需求澄清，并将任务自动拆解为可执行步骤，生成 plan.md 文件，助力高效解决复杂任务。
- 新增 Todo List 模式。复杂需求可自动拆解为简单步骤，并按顺序动态执行，提升任务管理与执行效率。
- Zulu 支持联网搜索。新增 Web Search 工具，可实时获取网络信息，辅助代码开发与问题解决。
- Zulu 检索性能优化。通过改进遍历算法、支持本地索引以及意图识别检索，提升查询速度与结果精准度。
- 改写功能升级。现支持跨文件预测改写，可自动识别非当前文件的其他关联代码，提升批量修改与重构效率。
- 历史 Query 支持方向键快速选择。在输入框中可通过方向键快速选择历史查询记录，提升操作便捷性与输入效率。
- 预览图片优化。调整关闭按钮大小，避免过小导致难以点击，提升图片预览交互体验。
- 【企业版】支持个人级自定义提示词。企业和团队级提示词的权限在前端显性化展示，方便管理与使用。
- 【企业版】自定义模型权限优化。支持按照部门维度配置访问权限，实现更精细的模型管理与团队协作控制。

### v3.13.0 (2025-08-15)

- Zulu 智能体执行指令时支持复用相同终端。显著减少了每次操作中打开的新终端数量，提升使用连贯性与资源利用效率。
- 快速设置命令与 MCP 自动执行。在对话输出过程中，一键设置后续任务为自动执行，提升执行效率与体验连贯性。
- 本地代码库知识增强索引支持 SVN 仓库，助力更全面的上下文理解与智能生成。
- 支持导出对话中图片。对话中生成的图片可快速右键复制，便于保存与后续使用，提升使用效率。
- 【企业版】对话支持多模型配置。现已支持自定义模型供应商并支持使用千帆平台的其他大模型，可按需灵活切换，满足不同场景下的生成需求。
- 【企业版】Zulu 新增 MAX 模式选择。内置 SOTA 模型，生成效果更强、更精准。
- 【企业版】知识集权限体系全面升级，适配网页与插件端问答场景。知识集现已与部门架构绑定，支持企业级、部门级、个人级三种访问权限，插件端与 Web 端问答颗粒度与权限一致。
- 历史会话记录支持滚动加载，替代分页展示，浏览更顺滑。
- 修复其他已知问题，提升用户体验。

### v3.12.0 (2025-07-21)

- 超级补全体验升级。在生成预测代码后，系统将同时展示 Tab 跳转入口与改写内容，方便快捷定位修改位置与进行内容对比，提升改写效率与操作流畅度。
- 默认启用智能 Agent 模式。升级后，所有用户将优先使用 Zulu 智能体进行任务处理，享受更智能、更一致的使用体验；后续使用会记住您上一次选择的模式。
- 关联私域知识集。Comate 在提供代码补全建议时，会参考您的上传的私域知识集，生成更贴合团队规范的代码。

### v3.11.0 (2025-07-07)

- 超级补全支持智能 import (VScode)。随着代码编写，识别出需要引入的类、函数、组件时，智能提示所需的 import 语句，无需手动查找引用路径。
- 对话中选择上下文更直观便捷。我们优化了自动上下文提取逻辑，并提升了手动添加文件时的交互体验与界面样式，让上下

文管理更高效、更可控。

- MCP 工具的使用和查看更方便。MCP 工具执行时修改为默认展开显示执行请求参数和返回结果，便于直观查看执行情况。

#### v3.10.0 (2025-06-24)

- AI 代码审查 (Code Review)。问答功能支持 AI 代码审查。您可以选中一段代码、指定一个或多个代码文件、或一段代码变更 (diff)，Comate 将依据您的企业规范 (上传至知识库中) 或默认通用编码规范，进行代码审查。
- 智能采纳更准确。生成代码更好地融入当前上下文，减少手动调整。

#### v3.9.0 (2025-06-03)

- 优化了从 Figma 设计稿生成前端代码的效果，您可以在 Zulu 对话框中通过输入斜线命令 / 使用此功能。
- 页面内置了精选的 MCP Server 列表，方便快速添加和管理。
- 支持创建和管理多条规则，可为每条规则设定不同的生效方式，包括自动生效、手动加入对话上下文、以及针对特定文件生效。此外，Zulu 规则还兼容您项目中已有的 Cursor rules 配置。
- 问答功能支持创建自定义的 Prompt (指令)，您可以在问答对话框中输入斜线命令 / 快速调用这些预设指令。
- 在问答中，当 Comate 生成的代码包含新文件名时，系统将能正确创建并命名新文件。
- 当问答结果包含网络搜索信息时，系统会添加引文标注，遵循 Markdown 语法格式 (例如 <sup>[1]</sup>)

#### v3.8.0 (2025-05-19)

- Zulu 智能体现在支持自动执行识别到的命令，让任务执行更连贯。
- 在与 Zulu 智能体交互时，您可以将代码恢复到对话历史中的任一版本。
- Chat 功能增强。
- 您可以选中多个代码块作为上下文与 Chat 进行交互。
- 超级代码补全体验升级。
- 当触发“改写”时，生成的代码建议会拆分为两个步骤展示。您可以分步预览和确认，确保每一处修改都符合您的预期。

#### v3.7.0 (2025-04-24) 更新了产品 Logo，统一文心快码的品牌视觉体验。

- Zulu 现已支持 MCP，与您的应用无缝衔接。
- Zulu 支持创建和应用自定义规则，满足个性化编码需求。
- 优化调试体验，可在终端错误信息上右键快速触发调试选项。
- 代码超级补全优化：无论追加、修改代码，还是采纳前后，您的光标位置始终保持不变。
- Tab 键采纳的提示框与光标保持同一行，操作更流畅。
- 新增代码部分以幽灵字符展示 (无绿色背景)，采纳后以绿色高亮显示，清晰直观。

#### v3.6.0 (2025-04-16)

- 修复已知问题，提升用户体验。

#### v3.4.0 (2025-03-06)

Zulu智能体发布公测：自动编程智能体，提供从代码生成到业务理解的完整智能服务。

**应用场景：**

- 全自然语言实现需求，不写一行代码。

- 理解代码库业务逻辑，帮助你迅速上手新项目。
- 帮你自动搭建开发环境和启动服务，不用一行一行输入命令。

#### 产品功能：

- 多工具自主调用：灵活调用多种实用工具，涵盖生成、执行命令等各个方面。
- 跨文件智能编辑：支持同时处理多个文件，智能同步修改关联代码。
- Diff 代码清晰展示：流式 diff 实时展示代码变化，更有效地进行代码对比和审查。
- 多模态高效交互：支持在会话中添加图片，让开发场景更加多元便捷。
- 代码效果实时预览：一键生成网页/游戏预览链接，浏览器实时呈现运行效果。

#### 新功能上线

- 接入 DeepSeek 模型，需升级为付费用户。
- 新增超级代码补全，可自动修正拼写错误，自动重命名变量，精准预测下一处要编辑的光标位置，智能重写目标位置的多行代码，需手动开启。
- 新增智能采纳功能，侧边栏问答生成的代码 diff 可以准确的合并到文件中。

#### 产品改进

- 提升函数注释、代码解释和调优建议的生成效果。
- 优化特定场景 Java 依赖图检索策略，提升续写效果。
- 优化了问答对话中止、状态处理等边界情况的处理逻辑，提升整体稳定性。
- 优化代码样式，调整行内代码颜色以匹配编辑器配色。
- 针对 Windows 系统优化进程创建策略，减少资源占用。
- 兼容 Windows 7 系统，满足更多用户的使用需求。

#### v3.3.0 (2025-01-03)

- 优化全链路的性能，带来更流畅的交互体验。
- 修复其他已知问题，提升用户体验。

#### v3.2.0 (2024-12-30)

- 修复其他已知问题，提升用户体验。
- Fixed other known issues to enhance user experience.

#### v3.1.0 (2024-12-18)

- 对话支持可选更快模型、更强模型，基础模型全新升级到128K。
- 安全智能体新增支持指定单文件、目录扫描。
- 修复其他已知问题，提升用户体验。
- Conversations now support optional faster and more powerful models, with the base model upgraded to 128K.
- The security agent now supports specifying single files and directory scanning.
- Fixed other known issues and improved user experience.

**v3.0.0 (2024-11-12)**

开发全流程升级，发布编码智能体、Debug智能体、单测智能体、安全智能体、全栈编程智能体。

Complete development workflow upgrade, introducing Development Agent, Debug Agent, Unit Test Agent, Security Agent, and Full-Stack Programming Agent.

**v2.17.0 (2024-10-28)**

- 修复其他已知问题，提升用户体验。
- Fixed other known issues and improved user experience.

**v2.16.0 (2024-10-14)**

- 优化首页推荐问题内容。
- 修复其他已知问题，提升用户体验。
- Optimized recommended question content on the homepage.
- Fixed other known issues and improved user experience.

**v2.15.0 (2024-09-30)**

- 行间对话交互优化：更佳输出效果。
- Paddle 插件体验优化，新增指令代码转换的功能指引更清晰。
- 修复其他已知问题，提升用户体验。
- Improved inline dialogue interaction: Better output results.
- Enhanced Paddle plugin experience, with clearer guidance for new instruction code conversion functionality.
- Fixed other known issues and improved user experience.

**v2.14.0 (2024-09-20)**

- 从终端报错跳转到对应代码行时提供分析与修复的快捷按钮，直面问题现场。
- Paddle 插件支持根据目录将文件中 Pytorch 框架转化为飞桨框架。
- 提问后增加关联问题建议，开拓提问角度和思路，让提问更深入和全面。
- 行间对话交互优化：更佳的流式输出效果，更准确的触发范围识别。
- 修复其他已知问题，提升用户体验。
- Added a CodeLens to provide fix suggestions for the line of code referenced in terminal error stacks.
- Added support for translating code written with PyTorch to the PaddlePaddle framework in the Paddle plugin.
- Added follow-up question suggestions after initial queries, broadening the scope and perspective of inquiries for more comprehensive and in-depth understanding.
- Improved the inline chat experience: smoother edit streaming in your editor and more accurate target code selection.
- Fixed other known issues to enhance user experience.

**v2.13.0 (2024-09-03)**

- 行间对话生成的代码支持流式输出了，更快看到编辑的效果，提升协同效率。
- 支持手动触发代码库索引构建，当回复未能参考代码库最新的代码时，可以手动触发更新。

- 对话侧边栏迎来样式全面升级，减少干扰元素，功能指示更清晰，便于专注在内容上。
- 优化对话回复中的 Mermaid 流程图样式，提升阅读体验。
- 修复其他已知问题，提升用户体验。
- Code generated in inline conversations now supports streaming output, allowing you to see editing effects faster and improving collaboration efficiency.
- Added support for manually triggering codebase index building. When responses don't reference the latest code in the codebase, you can manually trigger an update.
- The conversation sidebar has received a comprehensive style upgrade, reducing distracting elements and providing clearer function indicators for better focus on content.
- Improved the style of Mermaid flowcharts in conversation replies, enhancing the reading experience.
- Fixed other known issues to improve user experience.

#### v2.12.0 (2024-08-19)

- Comate 行间对话更新了，支持对划选的代码进行提问或编辑，清晰展示代码变更，快使用快捷键 `cmd+I`(macOS) / `ctrl+I`(Windows) 体验吧！
- 新增【智能生成 commit message】
- 新增「添加日志」按钮，一键为函数内部增加打印日志，便于 debug。
- 光标悬浮在任意插件时，浮窗显示更详细的插件信息和快捷提问按钮，插件能力更清晰。
- 光标悬浮在不同知识类型时会浮窗显示详细信息、并提供了快捷示例，激发灵感。
- 对话消息过多后自动隐藏历史消息，滚动到顶部后再加载，减小渲染开销。
- 提升首次打开对话侧边栏时的显示速度，提问 0 距离。
- 修复其他已知问题，提升用户体验。
- Comate inline dialogue has been updated, supporting querying or editing selected code, clearly displaying code changes. Try it with the shortcut `cmd+I` (macOS) / `ctrl+I` (Windows)!
- Added "Smart commit message generation"
- Added "Add log" button, allowing one-click addition of print logs inside functions for easier debugging.
- When hovering over any plugin, a floating window displays more detailed plugin information and quick question buttons, making plugin capabilities clearer.
- When hovering over different knowledge types, a floating window displays detailed information and provides quick examples, inspiring creativity.
- Automatically hides historical messages when there are too many dialogue messages, loading them again when scrolling to the top, reducing rendering overhead.
- Improved display speed when opening the dialogue sidebar for the first time, enabling zero-delay questioning.
- Fixed other known issues and improved user experience.

#### v2.11.0 (2024-07-31)

- 新增【续写状态】展示，实时展示生成状态。

- 优化多轮对话体验，生成更连贯的上下文。
- Diff 视图状态下支持一键采纳。
- 修复其他已知问题，提升用户体验。
- Added the display of "Continuation Status" to showcase the real-time generation status.
- Enhanced the multi-turn dialogue experience for more coherent context generation.
- Introduced one-click adoption support in Diff view mode.
- Fixed other known issues to improve user experience.

#### v2.10.0 (2024-07-17)

- 提升对话以及续写的速度。
- 修复其他已知问题，提升用户体验。
- Improved the speed of conversations and continuations.
- Fixed other known issues to enhance user experience.

#### v2.9.0 (2024-07-01)

- 使用能力更精准快捷，现在它不仅支持传统的关键字搜索，还增加了拼音搜索功能。
- 优化官方知识集展示形态。
- 修复其他已知问题，提升用户体验。
- Enhanced search capabilities are now more precise and efficient, supporting not only traditional keyword search but also pinyin search.
- Optimize the presentation form of the official knowledge base
- Fixed other known issues to enhance user experience.

#### v2.7.0 (2024-06-19)

- 新增【智能Debug】功能。当编译运行报错时，通过快捷按钮或勾选错误日志，一键定位错误并给出修复代码。
- 新增【代码安全】功能，企业版用户在管理员开启该功能后可在侧边栏中使用。插件会扫描代码中的安全漏洞，并提供修改建议。用户采纳后可一键修复安全漏洞。
- 支持混合云服务部署方案。
- 优化Go语言续写推荐效果。
- 优化用户个人账号信息的展示。
- 修复其他已知问题，提升用户体验。
- New AI Debug Feature: Easily locate and fix errors with a single click using shortcut buttons or by selecting error logs when compilation or runtime errors occur.
- New Code Security Feature: Available in the sidebar for enterprise users once enabled by the administrator. This plugin scans for security vulnerabilities in the code and provides suggestions for fixes. Users can apply these suggestions and resolve security issues with one click.
- Support for Hybrid Cloud Service Deployment.

- Enhanced Go Language Inline Suggestion.
- Improved Display of Personal Account Information.
- Fixed other known issues, improving user experience.

#### v2.6.0 (2024-06-04)

- 修改侧边栏启动样式：新增使用引导，帮助用户快速熟悉侧边栏功能。
- 续写推荐方式可配置：允许用户在设置中自定义续写推荐的方式，选择单行或多行优先。
- 注释生成代码交互优化：注释生成代码时，增加实时加载状态提示。
- 修复其他已知问题，提升用户体验。
- Improved the on-boarding experience of the chat interface: Added a few examples on the welcome page to help you quickly explore our latest features.
- Customize auto-completion setting: Allow users to customize how continuation recommendations are displayed in settings, choosing between single-line or multi-line prioritization.
- Optimize comment-to-code generation interaction: Provide real-time loading status when generating.
- Fixed other known issues, improving user experience.

#### v2.5.0 (2024-05-21)

- 搜索功能升级：输入「#」后选择「网络搜索」，可以轻松在侧边栏中进行网络搜索，获取更多相关信息。
- 历史提问便捷切换：通过键盘上的上下键，可以在侧边栏中快速切换历史提问内容，优化使用体验。
- 修复其他已知问题，提升用户体验。
- Improved Search: You can now search the web by typing "#" and selecting "Web Search" in the sidebar.
- Quickly Switch Historical Questions: Use the up and down arrow keys to switch between previous questions in the sidebar.
- Fixed Other Known Issues: We've resolved some known problems to enhance your experience.

#### v2.4.0 (2024-04-30)

- 在 Vue 文件中，JS/TS 代码中的函数支持通过快捷按钮，快速生成注释、生成单测等能力。
- 新增了对 JS/TS 代码中箭头函数的快捷按钮的支持。
- 在侧边栏中新增唤起「快捷指令」、「插件」和「知识」的按钮。
- 对调优建议指令进行了针对性优化，效果提升明显。
- 修复其他已知问题，提升用户体验。
- In Vue files, functions within JS/TS code now support quick generation of comments, unit tests etc through shortcut buttons.
- Added support for shortcut buttons in JS/TS code for arrow functions.
- New buttons for invoking "Commands", "Agents", and "Context" have been added to the sidebar.
- The optimization suggestion command has been specifically optimized, resulting in a significant improvement.
- Fixed other known issues, improving user experience.

#### v2.3.0 (2024-04-17)

- 对英文版做了进一步的支持，为全球用户提供智能编码服务：英文版个人官网、用户手册、更新日志上线；插件内支持英文使用；AutoWork高级生成能力支持英文对话。
- 在Comate中可以关联网页了，支持更个性化的问答：你可以通过在侧边栏中通过「#」关联你的URL链接，Comate可以帮你深度解读信息，并生成更贴合你需求背景的回答。
- 发布「工具箱」插件，支持Base64编解码、JSON和YAML互转等能力：你可以在侧边栏中通过「@」唤起「工具箱」插件，或者通过「/」直接调起快捷指令。
- 修复其他已知问题，提升用户体验。
- Further support has been added for the English version, providing intelligent coding services for global users: The English version of the personal profile, quick start, and change log are now available. English language is supported within the extension, and AutoWork's advanced generation capability now supports English chat.
- Now you can now link web pages, allowing for more personalized question and answer experiences in Baidu Comate: By using "#" in the sidebar, you can associate your URL links, enabling Comate to deeply interpret information and generate answers that are more tailored to your specific needs and background.
- We've released the "Toolbox" plugin, which supports Base64 encoding/decoding, JSON and YAML conversion, and more: You can summon the "Toolbox" plugin in the chat using "@" or directly activate shortcut commands using "/".
- Fixed other known issues, improving user experience.

#### v2.2.0 (2024-03-31)

- 现在，我们的应用已经支持英文，为更多全球用户提供便利。
- 对话区代码操作现在支持新建文件和插入到终端，代码采纳过程更加顺畅。
- 对话框现在支持 # 引用多个文件或者文件来进行提问，已获得更准确的回答。
- 对停止生成功能的交互进行了优化，使其变得更加容易操作。
- 修复其他已知问题，提升用户体验。
- Our application now supports English for more global users.
- Chat area now supports creating new files and inserting into terminals, making the code adoption process smoother.
- Chat now support referencing multiple files or directories with # for more accurate answers.
- Optimized the interaction for stopping generation, making it easier to operate.
- Fixed other known issues, improving user experience.

#### v2.1.0 (2024-03-20)

- 新增飞桨PaddlePaddle官方插件，支持代码生成，智能问答。
- 新增百度智能小程序官方插件，支持代码生成，智能问答。
- 现在，我们支持在消息流式输出过程中对代码进行操作，可以更快的采纳或复制代码。
- 对消息的渲染性能进行了优化，现在渲染长消息的速度提升了4倍，有效地解决了滚动卡顿的问题。
- 对采纳操作和Tab键的交互逻辑进行了智能化处理，让您在使用过程中享受到更流畅的用户体验。
- 我们对代码库相关检索逻辑进行了深度优化，提升了检索效率。
- 修复其他已知问题，提升用户体验。

- Added official plugin PaddlePaddle, supporting code generation and intelligent Q&A.
- Added official plugin 百度智能小程序, supporting code generation and intelligent Q&A.
- Now supports code manipulation during message stream output for faster adoption or copying of code.
- Optimized rendering performance of messages, resulting in a 4x speed increase in rendering long messages, effectively addressing scroll lag issues.
- Intelligently handled the interaction logic for adoption actions and Tab key, providing a smoother user experience.
- Deeply optimized code library related search logic, improving retrieval efficiency.
- Fixed other known issues, enhancing user experience.

#### v2.0.0 (2024-03-01)

- 我们正式发布了Comate+开放平台，现支持通过插件扩展Comate能力，提供强大的开放性与定制性，让你的编程现场从未有过的强大。
- ——发布多个插件，通过Git插件连接Github、Gitee、GitLab等多个常用工具，为你智能生成提交信息、创建Issue，快来通过【@】【/】【#】在Comate侧边栏中尝试最新的智能化开发吧。
- ——上线插件配置中心与知识中心。Comate的能力可以通过配置呈现更贴合你需求的效果；你还可以通过知识中心上传私域知识，让Comate更懂你，生成更个性化、更准确的内容与代码。
- ——通过丰富的开放基座，企业可以通过Comate的插件形式将自身服务、能力、知识等连接到编程现场，增强推荐效果、提供个性化能力，为组织量身打造自有智能代码助手。
- 上线了AutoWork，它能深度解读你的代码库，洞悉必要的背景知识，为你生成量身定制的代码。
- 新增对XML语言的智能补全提示，让您在编写XML文件时也能得到便利的帮助。
- 现在，当我们进行多行代码推荐时，您可以选择逐行采纳，让您根据需要更精准地把握每一行代码。
- 新增了插件内的小调研功能，让我们可以及时了解您对各项功能的满意度，便于我们进行针对性的优化。
- 导航栏新增了意见反馈入口，我们随时关注并解决用户遇到的问题。
- 我们对对话区的样式进行了全新改版，现在的页面信息呈现更加清晰，界面更加清爽。
- 代码检索策略进行了优化，提高您查找代码的速度和准确性，对检索和生成的界面进行了视觉效果升级，使用更加方便并拥有更好的视觉感受。
- 修复其他已知问题，提升用户体验。
- Officially launched the ComatePlus platform, now supporting extension of Comate capabilities through plugins, providing powerful openness and customization, making your programming scene more powerful than ever.
- Released multiple plugins, such as Git plugins connecting to Github, Gitee, GitLab, etc., intelligently generating commit messages, creating issues. Try out the latest intelligent development through [@] [/] [#] in the Chat panel.
- Launched Plugin Configuration Center and Docs Center. Comate's capabilities can be presented more fittingly through configuration; you can also upload domain-specific docs through the Docs Center, making Comate understand you better and generate more personalized, accurate content and code.
- Enterprises can connect their own services, capabilities, and knowledge to the programming scene through the rich open base of Comate in the form of plugins, enhancing recommendation effects, providing personalized capabilities, and tailoring intelligent code assistants for organizations.
- Launched AutoWork, which deeply interprets your codebase, understands necessary background docs, and generates

customized code for you.

- Added intelligent autocompletion prompts for XML language, providing convenient assistance when writing XML files.
- Now, when performing multi-line code suggestions, you can choose to accept line by line, allowing you to grasp each line of code more accurately as needed.
- Added a mini survey feature within plugins to promptly understand your satisfaction with various features, facilitating targeted optimization.
- Added a feedback entry to the navigation bar, where we constantly monitor and address issues encountered by users.
- Completely redesigned the style of the Chat area for clearer page information and a fresher interface.
- Optimized code retrieval strategy, improving the speed and accuracy of code lookup, and upgraded the visual effects of retrieval and generation interfaces for easier use and better visual experience.
- Fixed other known issues, enhancing user experience.

#### **v1.14.1 (2024-02-02)**

- 迎新年，贺新春，龙年主题活动开启，参与活动有机会赢积分、兑京东卡！
- 对话框顶部新增“帮助文档”和“设置”两个入口，便于您更加方便地获取帮助和进行配置。
- 对插件的分类和标签进行更新，使您在搜索 Comate 插件时更加轻松
- 我们将文案“自然语言生成代码”更改为“开始对话”，使其更容易理解
- 我们增加了消息的前置判断，防止在未输入有效内容时发送了消息
- 对消息的渲染样式进行了优化，以呈现更好的视觉效果
- 修复其他已知问题
- Celebrate the new year with the Dragon Year theme event! Engage for an opportunity to earn credit scores or redeem JD cards!
- Added "Help" and "Settings" entries at the top of the Chat area for easier access to help and configuration.
- Updated plugin categorization and tags to make it easier for you to search for Comate plugins.
- Changed the phrase "Natural Language Code Generation" to "Start Chat" for easier understanding.
- Added pre-checking for messages to prevent sending messages when no valid content is entered.
- Optimized message rendering styles for better visual effects.
- Fixed other known issues.

#### **v1.13.11 (2024-01-19)**

- 优化代码生成异常时的提示信息
- 修复部分 C++ 语言的文件不显示函数功能按钮的问题
- 修复侧边栏流式输出问答偶现输出中断的问题
- Enhanced error messages for code generation exceptions.
- Fixed the issue where certain C++ language files were not displaying buttons above functions.
- Resolved intermittent interruptions in the Chat panel output.

**v1.13.1 (2024-01-02)**

- 新增数据看板，助力企业研发效率提升
- 购买会员权益后可直接扫码登录使用，无需配置 License
- 优化代码解释功能的上下文获取策略，提升解释准确率
- Introduced a data dashboard to boost enterprise development efficiency.
- VIP Members can now directly log in via QR code after purchasing membership benefits, eliminating the need for license configuration.
- Optimized context retrieval strategies for code interpretation, enhancing accuracy.

**v1.13.0 (2023-12-19)**

- 支持百度账号认证，扫码快速登录
- Jupyter Notebook 文档支持代码续写功能
- 优化 Comate 推荐和 IDE 自带推荐框冲突的问题
- 修复链接远端机器时不显示函数功能按钮的问题
- Added support for Baidu account authentication, enabling quick login via QR code scanning.
- Jupyter Notebook documents now support code suggestions.
- Resolved conflicts between Comate suggestions and built-in IDE suggestion boxes.
- Fixed the issue where buttons above functions were not displayed when linking to remote machines.

**v1.12.1 (2023-12-05)**

- 优化侧边栏流式输出问答时，自动向下滑动的体验问题
- 优化打开侧边栏后输入框自动聚焦的问题，提升问答体验
- 修复函数注释结果的行首空格问题
- 修复侧边栏没有历史消息时仍可以清空消息的问题
- 修复获取默认用户名失败时插件激活失败的问题
- Improved the user experience of automatic downward scrolling during Chat output streaming.
- Enhanced the focus behavior of the input box upon opening the Chat, improving the chat experience.
- Fixed the leading whitespace issue in function comment results.
- Fixed the ability to clear messages in the Chat even when there are no historical messages.
- Resolved the issue of plugin activation failure when the default username retrieval fails.

**v1.2.0 (2023-11-21)**

- 增加侧边栏会话中查看生成代码变更的功能
- 增加生成结果自定义反馈的功能
- 增加敏感信息过滤的功能
- 支持侧边栏的流式输出，内容输出更流畅

- 支持行中推荐
- Added functionality to view code changes in Chat.
- Introduced customizable feedback for generated results.
- Implemented sensitive code filtering capabilities.
- Enabled smooth streaming output in the Chat, enhancing content delivery.
- Supported suggestions in the middle of line.

#### v1.1.0 (2023-11-07)

- 优化存在推荐内容时删除字符不流畅的体验问题
- 优化取消本次推荐命令并默认绑定快捷键的设置问题
- 修复单测生成不支持 C++ 语言的问题
- 修复获取不到设备 ID 的问题
- 修复侧边栏生成结果部分语言没有语法高亮的问题
- Improved the user experience of deleting characters when suggestion exist.
- Streamlined the process of canceling suggestions and default shortcut key bindings.
- Fixed the issue where unit test generation did not support the C++ language.
- Resolved the problem of failing to retrieve device IDs.
- Fixed the lack of syntax highlighting for some languages in the Chat results.

#### v1.0.0 (2023-10-24)

- 行级/函数级实时续写：单行续写、多行续写、注释生成代码（函数注释）
- 代码生成：自然语言描述生成代码、增强生成
- 单元测试生成：单测生成、接口文档生成测试代码
- 代码优化：长函数拆分代码重构、自然语言优化代码
- 代码注释生成：函数注释、行间注释
- 智能 CLI
- 技术问答
- Real-time Suggestions at the line/function level: single-line suggestions, multi-line suggestions, code generation from comments (function comments).
- Code generation: code generation from natural language descriptions, enhanced generation.
- Unit test generation: unit test generation, API documentation generation for test code.
- Code optimization: refactoring of long functions, optimization of code through natural language.
- Docstring generation: function docstring, inline comments.
- Intelligent CLI.
- Technical Q&A.

## JetBrains端插件更新日志

### v3.22.2 (2026-02-09)

- 历史记录修复：解决历史记录面板显示异常问题，恢复会话列表正常展示

### v3.22.1 (2026-01-26)

- 提示词优化：协助优化你的输入，补充更多信息，生成更精准
- 超级补全展示效果优化：修改代码涉及折叠区域时，将自动展开对应代码块，确保变更内容完整可见

### v3.22.0 (2026-01-12)

- 语义检索全面上线：语义检索能力已全量开放，覆盖更多检索场景，显著提升结果相关性与命中率。
- 支持并行执行多任务：并行开启多个对话窗口，同时处理代码理解、新需求实现与已有功能测试等不同任务，显著提升整体开发效率。
- 对话 Token 用量可视化：支持直观查看对话的 Token 消耗情况，通过主动管理上下文以更好适配模型的上下文窗口，提升对话稳定性与生成效果。
- 支持 AGENTS.md：现已支持符合社区规范的 AGENTS.md 文件，您可根据项目习惯选择使用 Rules 或 AGENTS.md，为 Agent 提供统一的上下文说明与行为约束。
- 支持手动触发上下文压缩：可自主控制对话中的上下文规模，提升长对话场景下的稳定性与生成效果。
- 历史消息交互升级：采用更轻量的浮层展示方式，历史任务的浏览与对话回溯更加高效便捷。
- 超级代码补全体验升级：优化生成内容与原代码的 Diff 展示，更清晰且不打扰；新增跨行位置预测，支持一步采纳代码，操作更加高效便捷。
- 支持主动加入「实验室」版本：在设置中开启开关后，即可使用预览版本，抢先体验最新能力。
- 日志管理优化：Comate 日志下载目录已调整为项目根目录，便于统一管理 with 快速定位。

### v3.21.0 (2025-12-22)

- 超级代码补全 Diff 样式全面焕新：升级新增、修改、删除等场景下生成代码的 Diff 展示方式，让代码变更一目了然，确保审阅时不再遗漏细节或误采纳代码，提升决策精准度与视觉舒适度。
- Checkpoint 机制升级：支持更细粒度的回滚能力，可回退至每一次提示词执行前或每一次工具调用前的代码状态，显著提升代码修改过程的可控性与安全性。
- Agent 变更文件流程优化 (VSCode)：Agent 生成完成后不再自动打开变更文件，减少不必要的窗口切换干扰，将查看与编辑主动权完全交还给用户。
- Agent 工作区创建行为优化：在未打开工作区的情况下提问时，将统一使用默认工作区并自动创建独立的 Demo 目录，既保证使用流程顺畅，又有效避免不必要的目录与窗口干扰。

### v3.20.1 (2025-12-09)

- 修复已知问题

### v3.20.0 (2025-12-08)

- 上下文理解增强：当前文件与选中代码将默认纳入上下文，并由模型自动判断其与对话的相关性，上下文理解更智能。
- 知识集检索能力增强：支持在指定的自定义文档集合下，对特定自定义文档进行精准检索，进一步提升查询的可控性与结果相关性。
- 知识集检索展示优化：检索结果中的知识来源地址支持一键跳转查看，提升溯源效率与知识查阅体验。侧边栏生成的代码支

持快速融合：生成的代码现可一键插入至光标位置，编辑更高效。

- 对话操作升级：支持对话复制、以及对话效果反馈，便于快速复用内容、反馈建议。
- 对话体验优化：模型生成过程中也可编辑输入框内容，操作更连贯、人机交互过程更自然。
- 输入体验优化：粘贴代码后可自动识别并展示来源文件及行号，代码定位更高效。
- 修复：优化了超级代码补全删除了正确空行的问题。
- 修复：侧边栏对话区域，粘贴图片不可用的问题。

#### v3.19.0 (2025-12-01)

- 上下文检索逻辑升级：重构检索重排逻辑，提升代码库检索结果排序准确性与相关度。
- 上下文管理优化：统一对话输入框中添加上下文的方式，支持使用 + 和 #，提升操作一致性与易用性。
- 代码库索引扩展：新增 Kotlin 文件的向量化处理，进一步提升代码理解与检索效果。
- MCP 能力升级：支持创建全局 MCP，可在多个 Workspace 间复用，降低重复安装成本，提高使用效率。
- Agent 工具优化：提升读文件工具的容错能力；优化写文件工具的放弃逻辑，支持同时删除文件及其所在目录，提高可靠性。
- GBK 文件兼容性增强：新增 GBK 编码文件在注释与文件对比视图中的中文正确显示支持。
- 行间命令优化：将「调优建议」指令升级为由 Zulu Agent 提供回答，增强上下文理解与生成质量。其他多项交互与体验优化。
- 修复代码补全在部分场景中多行新增未展示的问题。
- 修复代码超级补全中 AI 内容与 IDE 推荐内容无法同时展示的问题。修复快捷指令中“最小化 ICON 展示”配置项不生效的问题。

#### v3.18.3 (2025-11-17)

- 修复已知问题

#### v3.18.2 (2025-11-12)

- 修复已知问题

#### v3.18.1 (2025-11-11)

- 修复已知问题

#### v3.18.0 (2025-11-10)

- 全新 Agent 市场：内置丰富的垂类 Agent，可快速使用；同时提供统一设置页面，方便集中管理与展示。
- Agent 支持项目级共享：支持在项目范围内便捷共享自定义 Agent，方便团队复用经验与落地流程。Agent 支持基于 AI 创建简化 Agent 生成流程，提升使用便捷性与效率。
- Rules 支持自动创建：支持基于 AI 自动创建规则，简化配置方式、提高落地效率。
- 对话输入框全面升级：全面优化上下文选择与状态展示，新增上下文推荐与检索能力，并进行了多项样式与细节调整，带来更智能、更顺畅的输入体验。
- MCP 效果提升：调整解析逻辑、增强模型输出容错能力，提升 MCP 输出内容有效性。
- Agent 工具优化：优化执行命令时的操作文案，采用更简洁直观的表达方式，提升交互体验。

- 完善行间快捷指令设置：新增行号旁快捷指令的开关，统一 Codelens 整体设置项，使用更灵活。

#### v3.17.0 (2025-10-27)

- 上下文检索全面升级：支持全新检索方式，检索速度更快，兼容更多文件类型与更大规模索引；同时新增可视化索引管理，检索体验更高效直观。
- Agent 支持语义检索：新增基于用户意图的语义检索能力，使 Agent 更精准理解需求，提升生成结果的质量与可靠性。  
Agent 上下文扩展：现支持将 DOC 与 PDF 文件作为上下文使用，提升模型对文档内容的理解与处理能力。
- Agent 工具执行效率提升：现已支持并行读取文件，大幅提升 Agent 的生成速度与效率。
- Agent 效果优化：修复了循环调用工具及过度主动触发的问题，提升 Zulu 的调用稳定性与响应准确性。
- MCP 入口优化：将 MCP 入口前置，用户可更快速访问与使用 MCP 功能，提升操作便捷性。
- MCP 功能优化：在添加 Server 时新增必填信息提示，有效提高安装和配置的正确率。
- 网络配置优化：修复了插件代理相关问题，提升代理网络下使用的稳定性与兼容性。

#### v3.16.1 (2025-10-14)

- 修复已知问题

#### v3.16.0 (2025-10-13)

- 全新升级为 Agent First 形态：Chat 与 Zulu 合并为统一 Agent 形式，提供丰富官方 Agent，深度支持问答、规划、编码等核心场景，同时支持便捷切换与对话内上下文共享，提升复杂任务处理效率。
- Codelens 指令全面升级：自动切换最佳 Agent 回答，生成效果更精准。
- Agent 支持文档检索：在上下文类型中选择知识，Agent 在回答过程中会自动检索相关文档内容，显著提升生成质量。
- MCP 升级：现已支持在添加 MCP 后自动为所有官方 Agent 启用，使用更加便捷。
- 改写功能增强：现在即使未打开目标文件，也可支持跨文件跳转，提升多文件改写体验。
- 改写效果增强：优化上下文处理逻辑，确保模型输出结果准确可靠呈现。
- 对话输入框体验优化：集中优化了输入框的交互体验，提升输入流畅度与整体操作舒适性。

#### v3.15.0 (2025-09-15)

- Zulu 检索工具视觉优化：移除卡片样式，界面更简洁直观，提升使用体验。
- Zulu 上下文理解优化：策略升级，让系统在处理多段内容时更智能、更连贯，提高响应的准确性和相关性。
- Zulu 工具安全增强：防止对敏感文件的非法写入与修改，提升系统安全性。
- Zulu 性能优化：改进内存管理机制，降低超大文件读取时的内存占用，提升稳定性与性能。
- 插件性能与稳定性优化：提升响应速度，增强稳定性，修复已知问题。

#### v3.14.1 (2025-09-11)

- 修复已知问题

#### v3.14.0 (2025-09-08)

- 支持 Subagent，多智能体协同。主智能体可智能调度子智能体进行任务拆解与分工执行，显著提升复杂任务的处理效率。

#### v3.13.1 (2025-09-02)

- 修复已知问题

**v3.13.0 (2025-09-01)**

- Zulu 支持联网搜索：新增 Web Search 工具，可实时获取网络信息，辅助代码开发与问题解决。
- Zulu 检索性能优化：通过改进遍历算法、支持本地索引以及意图识别检索，提升查询速度与结果精准度。
- 改写功能升级：现支持跨文件预测改写，可自动识别非当前文件的其他关联代码，提升批量修改与重构效率。
- 历史 Query 支持方向键快速选择：在输入框中可通过方向键快速选择历史查询记录，提升操作便捷性与输入效率。
- 预览图片优化：调整关闭按钮大小，避免过小导致难以点击，提升图片预览交互体验。
- 性能优化：通过多项改进显著提升 Comate 响应速度，改善使用流畅性。
- 【企业版】支持个人级自定义提示词
- 企业和团队级提示词的权限在前端显性化展示，方便管理与使用。
- 【企业版】自定义模型权限优化
- 支持按照部门维度配置访问权限，实现更精细的模型管理与团队协作控制。

**v3.12.1 (2025-08-28)**

- 修复已知问题

**v3.12.0 (2025-08-25)**

- 新增 自定义智能体：支持可根据项目需求创建专属智能体，自定义能力与行为，打造深度契合个人习惯或团队流程的智能体。
- 新增 Todo List 模式：复杂需求可自动拆解为简单步骤，并按顺序动态执行，提升任务管理与执行效率。

**版本：v3.11.0 - 2025.08.15****新增 Plan Agent**

- 支持交互式需求澄清，并将任务自动拆解为可执行步骤，生成 plan.md 文件，助力高效解决复杂任务。  
**本地代码库知识增强索引支持 SVN 仓库**
- 现已支持对 SVN 代码库进行知识增强索引，助力更全面的上下文理解与智能生成。  
**历史会话记录滚动加载**
- 历史会话记录支持滚动加载，替代分页展示，浏览更顺滑。

**Version: v3.11.0 - 2025.08.15****New: Plan Agent**

- Supports interactive requirement clarification and automatically breaks down tasks into executable steps, generating a plan.md file to facilitate efficient resolution of complex tasks.  
**Local Codebase Knowledge Enhancement Index Supports SVN Repositories**
- Knowledge enhancement indexing for SVN code repositories is now supported, aiding in more comprehensive context understanding and intelligent generation.  
**Historical Session Records Support Scroll Loading**
- Historical session records now support scroll loading, replacing paginated display for smoother browsing.

**版本：v3.10.0 - 2025.07.22****超级补全体验升级**

- 在生成预测代码后，系统将同时展示 Tab 跳转入口与改写内容，方便快速定位修改位置与进行内容比对，提升改写效率与操作流畅度。

### 智能体执行指令时支持复用相同终端

- 显著减少了每次操作中打开的新终端数量，提升使用连贯性与资源利用效率。
- 关联私域知识集。
- Comate 在提供代码补全建议时，会参考您的上传的私域知识集，生成更贴合团队规范的代码。

### 默认启用智能 Agent 模式

- 升级后，所有用户将优先使用 Zulu 智能体进行任务处理，享受更智能、更一致的使用体验；后续使用会记住您上一次选择的模式。

Version: v3.10.0 - 2025.07.22

### Super Completion Experience Upgrade

- After generating predictive code, the system will now display both the Tab jump entry point and the rewritten content simultaneously. This makes it easier to quickly locate modification positions and compare content, improving rewrite efficiency and operational fluency.

### Agent Execution Supports Reusing the Same Terminal

- Significantly reduces the number of new terminals opened for each operation, enhancing usage continuity and resource utilization efficiency.
- Reference Private Knowledge Sets: Comate will reference your uploaded private knowledge sets when providing code completion suggestions, generating code that better aligns with team standards.

### Smart Agent Mode Enabled by Default

- After the upgrade, all users will prioritize using the Zulu Agent for task processing, enjoying a smarter and more consistent user experience. Subsequent usage will remember your last selected mode.

版本 : v3.9.1 - 2025.07.17

- 修复已知问题。

Version: v3.9.0 - 2025.07.17

- Fixed known issues.

版本 : v3.9.0 - 2025.07.07

### Zulu 智能体

- MCP 工具的使用和查看更方便。MCP 工具执行时修改为默认展开显示执行请求参数和返回结果，便于直观查看执行情况。

### 代码补全

- 超级补全支持自动检测 Linter 报错并生成修复后代码。代码中出现 Linter 报错，可查看自动生成的修复代码并 Tab 采纳。

### 优化

- 对话中选择上下文更直观便捷。优化了自动上下文提取逻辑，提升了手动添加文件时的交互体验与界面样式，让上下文管理更高效、更可控。
- 兼容性优化。通过简单的系统设置，体验更友好的会话、Zulu和开放平台等能力。

Version: v3.9.0 - 2025.07.07

### Zulu Agent

- Easier MCP Tool Usage and Viewing: When MCP tools are executed, the execution request parameters and return results are now displayed expanded by default, allowing for intuitive viewing of the execution status.

### Code Completion

- Super Completion Supports Automatic Linter Error Detection and Fixes: When Linter errors appear in your code, you can

now view automatically generated fix code and adopt it with Tab.

#### Optimizations

- More Intuitive Context Selection in Conversations:\*\* Optimized the automatic context extraction logic and improved the interaction experience and UI style when manually adding files. This makes context management more efficient and controllable.
- Compatibility Enhancements:\*\* Through simple system settings, enjoy a more user-friendly experience with conversational features, Zulu, and the open platform capabilities.

#### 版本 : v3.8.2 - 2025.06.27

- 修复已知问题。

#### Version: v3.8.2 - 2025.06.27

- Fixed known issues.

#### 版本 : v3.8.1 - 2025.06.26

- 修复已知问题。

#### Version: v3.8.1 - 2025.06.26

- Fixed known issues.

#### 版本 : v3.8.0 - 2025.06.24

##### Zulu 智能体

- 智能体执行失败自动重试。无论是对话生成失败，还是工具调用出错，Zulu 将自动尝试重试，显著提升成功率与稳定性。

##### 代码补全

- 关联私域知识集。Comate 在提供代码补全建议时，会参考您上传的私域知识，生成更符合企业规范的代码内容。
- 支持智能 import。补全建议中将自动识别并提示所需的 import 语句，提升编码效率。

##### 问答

- AI 代码审查功能上线。您可以选中代码段、多个文件或一段 diff，由 Comate 基于通用或企业上传的编码规范进行自动审查。
- 智能采纳更精准。问答生成的代码更好地融入当前上下文，减少重复追加和手动调整。

#### Version: v3.8.0 - 2025.06.24

##### Zulu Agent

- Automatic Retry on Agent Execution Failure: Whether it's a conversational generation failure or a tool call error, Zulu will automatically attempt retries, significantly improving success rates and stability.

##### Code Completion

- Reference Private Knowledge Sets: Comate will reference your uploaded private knowledge when providing code completion suggestions, generating code that better adheres to your company's standards.
- Intelligent Import Support: Completion suggestions will now automatically identify and prompt for necessary import statements, enhancing coding efficiency.

##### Chat

- AI Code Review Feature Launched: You can now select code snippets, multiple files, or a diff for Comate to automatically review based on general or enterprise-uploaded coding standards.
- Smarter Code Adoption: Code generated by Chat now integrates better with the current context, reducing redundant

additions and manual adjustments.

#### 版本 : v3.7.2 - 2025.06.13

- 修复已知问题。

#### Version: v3.7.2 - 2025.06.13

- Fixed known issues.

#### 版本 : v3.7.1 - 2025.06.12

- 修复已知问题。

#### Version: v3.7.1 - 2025.06.12

- Fixed known issues.

#### 版本 : v3.7.0 - 2025.06.03

##### Zulu 智能体

- F2C (Figma 设计稿转代码) 功能现已集成到 Zulu 中。我们优化了从 Figma 设计稿生成前端代码的效果，您可以在 Zulu 对话框中通过输入斜线命令 / 使用此功能。
- MCP Server 新增了专属的展示与配置页面。页面内置了精选的 MCP Server 列表，方便快速添加和管理。
- 全新规则 (Rules) 管理界面。支持创建和管理多条规则，可为每条规则设定不同的生效方式，包括自动生效、手动加入对话上下文、以及针对特定文件生效。此外，Zulu 规则还兼容您项目中已有的 Cursor rules 配置。

##### 问答

- 自定义 Prompt 与指令。问答功能支持创建自定义的 Prompt (指令)，您可以在问答对话框中输入斜线命令 / 快速调用这些预设指令。
- 智能代码采纳。问答生成的代码片段可以被智能地合并到代码文件中，为您减少手动代码合并工作。
- 新文件采纳体验提升。在问答中，当 Comate 生成的代码包含新文件名时，系统将能正确创建并命名新文件。
- 查看变更与预览。在“调优建议”生成的代码块上，新增了“查看变更”的能力。
- 网络搜索结果自动引用标注。当问答结果包含网络搜索信息时，系统会添加引文标注，遵循 Markdown 语法格式 (例如 <sup>[1]</sup>)。

##### 代码补全

- 参考最近打开的文件。代码补全会参考您最近在 IDE 中打开过的文件内容，使补全建议更精准。
- 改写场景支持在新的改写内容展示时同时展示行号，方便定位新生成的代码位置。

##### 其他

- 侧边栏界面简化。侧边栏界面移除了重复的“对话”入口。
- 一些已知问题的修复和性能优化。

#### Version: v3.7.0 - 2025.06.03

##### Zulu Agent

- Figma to Code Integration: The Figma to Code functionality is now integrated into Zulu. We've optimized the process of generating front-end code from Figma designs. You can access this feature within the Zulu chat interface using the slash command `/f2c`.
- MCP Server Dedicated Page: MCP Server now features a dedicated display and configuration page. This page includes a curated list of MCP Servers, making it easier to add and manage them quickly.

- **New Rules Management Interface:** A brand new Rules management interface has been introduced. It supports the creation and management of multiple rules, allowing you to define different activation methods for each rule, including: automatic activation, manual inclusion in the conversation context, and activation for specific files. Furthermore, Zulu Rules are compatible with your existing Cursor rules configurations.

#### Chat

- **Custom Prompts and Instructions:**The Chat feature now supports creating custom prompts (instructions). You can input slash commands like `/my_custom_prompt` in the Chat dialog to quickly invoke these preset instructions.
- **Intelligent Code Adoption:**Code snippets generated by Chat can now be intelligently merged into your code files, reducing the need for manual code merging.
- **Improved New File Adoption Experience:**When Comate-generated code in Chat includes a new filename, the system will now correctly create and name the new file.
- **View Changes and Previews:**The "Tune-up Suggestions" generated code blocks now include a "View Changes" capability, allowing for clearer code comparison.
- **Automatic Citation for Web Search Results:**When Chat results include information from web searches, the system will now add citation markers in Markdown format (e.g., <sup>[1]</sup>).

#### Code Completion

- **Reference Recently Opened Files:**Code completion will now reference the content of files you've recently opened in your IDE, leading to more precise completion suggestions.
- **Rewrite Scenario Line Number Display:**When rewrite content spans across multiple screens, line numbers will now be displayed to facilitate locating newly generated code.

#### Other

- **Simplified Sidebar Interface:**The duplicate "Conversations" entry has been removed from the sidebar interface.
- Fixes for several known issues and performance optimizations.

#### 版本 : v3.6.1 - 2025.05.26

- 修复已知问题。

#### Version: v3.6.1 - 2025.05.26

- Fixed known issues.

#### 版本 : v3.6.0 - 2025.05.19 功能发布

- Zulu 智能体现在支持自动执行识别到的命令，让任务执行更连贯。
- 在与 Zulu 智能体交互时，您可以将代码恢复到对话历史中的任一版本。
- MCP Server 新增了专属的展示与配置页面。页面内置了精选的 MCP Server 列表，方便快速添加和管理。 **优化**
- 当改写内容跨越多屏显示时，会清晰展示行号，方便定位。
- 现在您可以通过多次按下 ESC 键来取消代码补全的推荐。
- Tab 跳转等操作符号的颜色和字号，现在能够自适应当前使用的主题。
- 修复已知问题。

#### Version: v3.6.0 - 2025.05.19

#### Feature Releases

- The Zulu Agent now supports automatic execution of identified commands, allowing for more seamless task completion.

- When interacting with the Zulu Agent, you can now revert code to any version within the conversation history.
- MCP Server has gained a dedicated display and configuration page. This page includes a curated list of MCP Servers for quick addition and management.

#### Optimizations

- When rewritten content spans across multiple screens, line numbers are now clearly displayed for easier navigation.
- You can now cancel code completion suggestions by pressing the ESC key multiple times.
- The color and font size of operational symbols like Tab jumps are now adaptive to the currently used theme.
- Fixed known issues.

#### 版本 : v3.5.1 - 2025.04.24

- 修复已知问题。

#### Version: v3.5.1 - 2025.04.24

- Fixed known issues.

#### 版本 : v3.5.0 - 2025.04.22

- 支持MCP开放生态。
- 代码改写升级，支持智能跨屏跳转。
- 修复已知问题。

#### Version: v3.5.0 - 2025.04.22

- Support for MCP open ecosystem.
- Upgraded code rewriting with support for intelligent cross-screen navigation.
- Fixed known issues.

#### 版本 : v3.4.1 - 2025.04.18

- 兼容Jetbrains 25\* 版本。

#### Version: v3.4.1 - 2025.04.18

- Compatible with JetBrains 25\* versions.

#### 版本 : v3.4.0 - 2025.04.17

- 代码改写优化，支持智能跳转。
- 修复已知问题。

#### Version: v3.4.0 - 2025.04.17

- Code rewriting optimization with support for intelligent navigation.
- Fixed known issues.

#### 版本 : v3.3.4 - 2025.03.24

- 修复已知问题。

#### Version: v3.3.4 - 2025.03.24

- Fixed known issues.

**版本：v3.3.3 - 2025.03.14**

- 修复已知问题。

**Version: v3.3.3 - 2025.03.14**

- Fixed known issues.

**版本：v3.3.2 - 2025.03.10**

- 新增超级代码补全，可自动修正拼写错误，自动重命名变量，精准预测下一处要编辑的光标位置，智能重写目标位置的多行代码（需手动开启）。

**Version: v3.3.2 - 2025.03.10**

- Introduced Super Code Completion, which automatically corrects spelling errors, renames variables, accurately predicts the next cursor position for editing, and intelligently rewrites multiple lines of code at a target location (requires manual activation).

**版本：v3.3.1 - 2025.03.06**

**Zulu智能体全新发布**

- 自动编程智能体，提供从代码生成到业务理解的完整智能服务。
- 全自然语言实现需求，不写一行代码。
- 理解代码库业务逻辑，帮助你迅速上手新项目。
- 帮你自动搭建开发环境和启动服务，不用一行一行输入命令。

#### **产品功能**

- 多工具自主调用：灵活调用多种实用工具，涵盖生成、执行命令等各个方面。
- 跨文件结构编辑：支持同时处理多个文件，智能同步修改关联代码。
- Diff代码高亮显示：流式地展示代码变更，更有效地进行代码对比和审查。
- 多模态高效交互：支持在会话中添加图片，让开发场景更加多元便捷。
- 代码效果实时预览：一键生成网页/游戏预览链接，浏览器实时呈现运行效果。

#### **新功能**

- JetBrains系列IDE支持查看历史对话。

#### **产品改进**

- 提升函数注释、代码解释和调优建议的生成效果。
- 优化特定场景Java依赖包检索策略，提升续写效果。
- 优化了问答对话式UI，状态处理等边界情况的处理逻辑，提升整体稳定性。
- 针对Windows系统优化流程创建策略，减少资源占用。
- 兼容MacOS 7系统，满足更多用户的使用需求。

**Version: v3.3.1 - 2025.03.06**

**Zulu Agent is newly released**

- An automated programming agent that offers comprehensive intelligent services from code generation to business understanding.

- Meet requirements with pure natural language without writing a single line of code.
- Comprehend the business logic of the codebase to help you quickly get started with new projects.
- Automatically set up the development environment and start services without entering commands line by line.

#### Product Features

- Autonomous multi - tool invocation: Flexibly call a variety of practical tools, covering aspects such as generation and command execution.
- Cross - file structure editing: Support simultaneous handling of multiple files and intelligently synchronize modifications to related code.
- Highlighted diff code display: Streamingly present code changes for more effective code comparison and review.
- Efficient multi - modal interaction: Support adding pictures to conversations, making the development scenario more diverse and convenient.
- Real - time code effect preview: Generate preview links for web pages/games with one click and display the running effects in the browser in real time.

#### New Features

- JetBrains series of IDEs support viewing historical conversations.

#### Product Improvements

- Enhance the generation effect of function comments, code explanations, and optimization suggestions.
- Optimize the retrieval strategy for Java dependency packages in specific scenarios to improve the code continuation effect.
- Optimize the Chat UI and the processing logic for boundary cases such as state handling to enhance overall stability.
- Optimize the process creation strategy for the Windows system to reduce resource consumption.
- Be compatible with MacOS 7 system to meet the usage needs of more users.

#### 版本 : v3.2.0 - 2025.02.21

- 企业级用户，侧边栏问答支持DeepSeek-R1。

#### Version: v3.2.0 - 2025.02.21

- For enterprise users, Chat now supports DeepSeek-R1.

#### 版本 : v3.1.2 - 2025.01.22

- 单测智能体支持windows系统。
- 修复已知问题。

#### Version: v3.1.2 - 2025.01.22

- Unit Test Agent now supports Windows systems.
- Fixed known issues.

#### 版本 : v3.1.1 - 2024.12.30

- 修复已知问题。

#### Version: v3.1.1 - 2024.12.30

- Fixed known issues.

**版本 : v3.1.0 - 2024.12.17**

- 侧边栏问答支持模型切换。
- 新增接口文档生成。
- 弱网卡顿优化。
- 修复已知问题。

**Version: v3.1.0 - 2024.12.17**

- Chat now supports model switching.
- Added API documentation generation.
- Optimized performance in weak network conditions.
- Fixed known issues.

**版本 : v3.0.6 - 2024.12.13**

- 修复已知问题

**Version: v3.0.6 - 2024.12.13**

- Fixed known issues.

**版本 : v3.0.5 - 2024.12.02**

- 修复已知问题

**Version: v3.0.5 - 2024.12.02**

- Fixed known issues.

**版本 : v3.0.4 - 2024.11.21**

- 对快捷键进行优化。

**Version: v3.0.4 - 2024.11.21**

- Optimized keyboard shortcuts.

**版本 : v3.0.3 - 2024.11.18**

- 修复卡顿问题。

**Version: v3.0.3 - 2024.11.18**

- Fixed stuttering issues.

**版本 : v3.0.2 - 2024.11.15**

- 修复已知问题。

**Version: v3.0.2 - 2024.11.15**

- Fixed known issues.

**版本 : v3.0.1 - 2024.11.13**

- 修复已知问题。

**Version: v3.0.1 - 2024.11.13**

- Fixed known issues.

**版本 : v3.0.0 - 2024.11.10**

- 开发全流程升级，发布编码智能体、Debug智能体、单测智能体、安全智能体、全栈编程智能体。

**Version: v3.0.0 - 2024.11.10**

- Full development lifecycle upgrade, releasing Coding Agent, Debug Agent, Unit Test Agent, Security Agent, and Full-Stack Programming Agent.

**版本 : v2.13.5 - 2024.11.05**

- 修复多项目打开关闭其一时Comate异常。

**Version: v2.13.5 - 2024.11.05**

- Fixed an issue where Comate behaved abnormally when multiple projects were opened or closed sequentially.

**版本 : v2.13.4 - 2024.10.28**

- 优化配置不同步问题。

**Version: v2.13.4 - 2024.10.28**

- Optimized configuration synchronization issues.

**版本 : v2.13.3 - 2024.10.24**

- 修复已知问题。

**Version: v2.13.2 - 2024.10.18**

- Fixed known issues.

**版本 : v2.13.2 - 2024.10.18**

- 侧边栏新增意图识别，根据用户问题自动挂载相关知识，提高问答质量。

- 修复已知问题。

**Version: v2.13.2 - 2024.10.18**

- Added intent recognition to the sidebar; automatically mounts relevant knowledge based on user questions to improve Chat quality.

- Fixed known issues.

**版本 : v2.13.1 - 2024.10.14**

- 修复已知问题。

**Version: v2.13.1 - 2024.10.14**

- Fixed known issues.

**版本 : v2.13.0 - 2024.09.30**

- Git Commit Message自动生成，可在Git Commit Changes页面点击Comate图标，自动为您生成提交信息。

- 支持Java文件级别的函数注释自动生成，只需在文件上右键点击并选择“生成函数注释”选项，即可轻松为该文件中全部函数生成注释。

**Version: v2.13.0 - 2024.09.30**

- Git Commit Message Auto-Generation: Click the Comate icon on the "Git Commit Changes" page to automatically generate commit messages for you.
- Java File-Level Function Comment Auto-Generation: Easily generate comments for all functions in a Java file by right-clicking on the file and selecting the "Generate Function Comments" option.

**版本 : v2.12.0 - 2024.09.20**

- 全新上线【单测 Agent】能力，自动生成合格的测试用例，有效地提高代码覆盖率。
- 支持手动触发代码库索引构建，当回复未能参考代码库最新的代码时，可以手动触发更新。
- 侧边栏相关问题推荐，当您在侧边栏做出问答时，模型主动帮您预测相关问题。
- Pycharm控制台异常爆栈显示comate图标，点击可进行AutoDebug修复。

**Version: v2.12.0 - 2024.09.20**

- Newly Launched [Unit Test Agent] Capability: Automatically generates qualified test cases, effectively improving code coverage.
- Supports manual triggering of code repository index building. When replies fail to reference the latest code in the repository, you can manually trigger an update.
- Sidebar related issue recommendations: When you ask questions in the sidebar, the model proactively predicts related issues for you.
- PyCharm console abnormal stack overflow displays the Comate icon; clicking it allows for AutoDebug fixes.

**版本 : v2.11.1 - 2024.09.14**

- 企业安全审查功能。

**Version: v2.11.1 - 2024.09.14**

- Enterprise security review feature.

**版本 : v2.11.0 - 2024.09.03**

- 企业数据管理升级。
- 修复语言不推荐问题。

**Version: v2.11.0 - 2024.09.03**

- Enterprise data management upgraded.
- Fixed issues with language recommendations.

**版本 : v2.10.1 - 2024.08.28**

- 修复侧边栏账户类型显示问题。

**Version: v2.10.1 - 2024.08.28**

- Fixed an issue with the display of account types in the sidebar.

**版本 : v2.10.1 - 2024.08.26**

- 私有域配置优化。

**Version: v2.10.1 - 2024.08.26**

- Optimized private domain configuration.

**版本 : v2.10.0 - 2024.08.19**

- inline-chat编辑区问答功能，当您选中代码后点击Comate icon呼出输入框，无需跳出编码区即可进行技术问题（开启方式：Settings-文心快码（Baidu Comate）-允许展示InlineChat提示）。
- 为函数添加日志功能，帮你的代码补充必要的日志打印。
- 调优建议功能优化。
- 网络请求优化。
- 修复Android Studio侧边栏显示问题。

**Version: v2.10.0 - 2024.08.19**

- Inline Chat for Code Editing: When you select code and click the Comate icon, an input box will appear, allowing you to ask technical questions without leaving the coding area. (To enable: Go to Settings - Wenxin Kuaima (Baidu Comate) - Allow InlineChat Prompts).
- Add Logging to Functions: Adds necessary log printing to your code.
- Optimized suggestion feature.
- Optimized network requests.
- Fixed an issue with Android Studio sidebar display.

**版本 : v2.9.1 - 2024.08.15**

- 修复部分语言不推荐代码的问题。

**Version: v2.9.1 - 2024.08.15**

- Fixed an issue where code recommendations were not provided for some languages.

**版本 : v2.9.0 - 2024.08.13**

- 修复已知问题。

**Version: v2.9.0 - 2024.08.13**

- Fixed known issues.

**版本 : v2.8.0 - 2024.08.09**

- IDE内控件展示优化。

**Version: v2.8.0 - 2024.08.09**

- Optimized the display of controls within the IDE.

**版本 : v2.7.0 - 2024.08.08**

- 全新的代码解释功能，全局视角理解代码，输出流程图辅助理解。
- Comate推荐与IDE原生推荐共存体验优化。
- 弱网环境体验优化。

**Version: v2.7.0 - 2024.08.08**

- Brand new code explanation feature: Understand code from a global perspective with flowchart output to aid

comprehension.

- Optimized the experience of Coexisting Comate recommendations and native IDE recommendations.
- Optimized experience in low-network environments.

#### 版本 : v2.6.0 - 2024.07.31

- 增强多轮对话功能。
- 侧边栏首页新增用户使用引导，方便快速上手插件能力。

#### Version: v2.6.0 - 2024.07.31

- Enhanced multi-turn conversation capabilities.
- The sidebar homepage now includes user onboarding guides for easy adoption of plugin features.

#### 版本 : v2.5.0 - 2024.07.19

- AutoWork增加全网检索能力「#网络搜索」。
- 智能Debug功能生成代码场景，新增【查看变更】和【采纳】按钮；支持修复方案中报错文件跳转。
- 修复采纳等快捷键修改后原快捷键仍然生效的问题。
- 修复Python生成函数注释位置异常bug。

#### Version: v2.5.0 - 2024.07.19

- AutoWork now has full network search capability "[#Network Search]".
- For code generation scenarios in the Smart Debug feature, "View Changes" and "Accept" buttons have been added; supports jumping to error files within fix suggestions.
- Fixed an issue where original shortcuts remained active after modifying them (e.g., for "Accept").
- Fixed a bug where Python generated function comments appeared in an abnormal location.

#### 版本 : v2.4.3 - 2024.07.08

- 修复异常情况导致的“当前License已经在其它设备上使用”问题。

#### Version: v2.4.3 - 2024.07.08

- Fixed the "Current license is already in use on another device" issue caused by exceptional circumstances.

#### 版本 : v2.4.2 - 2024.07.01

- 支持企业私有域配置，如果您的企业内网中限制了对公网的访问，您可以配置一个私有域名，然后将comate的流量转发到我们的公网服务。
- 修复Ubuntu及Windows系统中Comate侧边栏展示异常的问题。

#### Version: v2.4.2 - 2024.07.01

- Supports enterprise private domain configuration. If your company's internal network restricts public internet access, you can configure a private domain and forward Comate's traffic to our public network service.
- Fixed an issue where the Comate sidebar displayed abnormally on Ubuntu and Windows systems.

#### 版本 : v2.4.1 - 2024.06.28

- 开放平台中【飞桨】插件内置相关知识，支持提问开发相关的问题。

**Version: v2.4.1 - 2024.06.28**

- The [PaddlePaddle] plugin in the open platform now includes relevant built-in knowledge, supporting questions related to development.

**版本 : v2.4.0 - 2024.06.27**

- 插件适配IDE代理配置。
- 优化侧边栏样式，用户可在右上角查看【个人信息】或者执行【登出账户】操作。

**Version: v2.4.0 - 2024.06.27**

- The plugin now adapts to IDE proxy configurations.
- Optimized sidebar style. Users can view [Personal Information] or perform [Logout] operations in the top right corner.

**版本 : v2.3.0 - 2024.06.19**

- 新增【智能Debug】功能。当编译运行报错时，通过快捷按钮或划选错误日志，一键定位错误并给出修复代码。
- 新增【代码安全】功能，企业版用户在管理员开启该功能后可在侧边栏中使用。插件会扫描代码中的安全漏洞，并提供修改建议。用户采纳后可一键修复安全漏洞。
- Comate已索引常见公开API文档作为公共知识集，您可在侧边栏输入#后选择API，搜索选中一个文档，将此文档作为知识然后输入自然语言可进行生成调用代码、Mock、自动化测试等操作。

**Version: v2.3.0 - 2024.06.19**

- **[Smart Debug] Feature Added:** When compilation or runtime errors occur, you can locate the error and get suggested fixes with a single click by using the shortcut button or by selecting the error log.
- **[Code Security] Feature Added:** Enterprise users can use this feature in the sidebar after the administrator enables it. The plugin scans code for security vulnerabilities and provides modification suggestions. Users can fix security vulnerabilities with a single click after accepting the suggestions.
- Comate has indexed common public API documentation as a public knowledge set. You can type "#" in the sidebar to select an API, search for and select a document, and then use this document as knowledge to generate calling code, mocks, automated tests, and other operations by inputting natural language.

**版本 : v2.2.0 - 2024.06.04**

- AutoWork支持选中代码提问，您可在编辑区选中代码并在侧边栏@AutoWork后，针对选中的代码进行技术问答。
- 新增注释生成代码提示功能，您可以在编辑区输入完doc注释后按下回车键，Comate会提示您正在生成代码，此时您稍加停留，等待代码生成。

**Version: v2.2.0 - 2024.06.04**

- AutoWork supports code selection for questions. You can select code in the editor, then type "@AutoWork" in the sidebar to ask technical questions about the selected code.
- Added a new "comment to code" generation prompt feature. After you finish typing a doc comment in the editor and press Enter, Comate will prompt you that code is being generated. Stay for a moment to wait for the code to be generated.

**版本 : v2.1.0 (2024-05-21)**

- 单测续写自动识别UT框架，当您正在编写单元测试方法时，Comate会自动检测当前的UT框架为您推荐更优质代码。
- 单测支持scala、kotlin语言，可通过选中代码右键生成对应的单测用例。
- Comate侧边栏输入框支持上下键切换输入历史。

- 解决编码时Comate推荐影响IDE自带提示问题，现在Comate推荐的代码与IDE自带的提示会同时出现；Tab采纳Comate推荐，回车采纳IDE自带推荐。
- 侧边栏输出的代码内容，您可以点击【新建文件】将其保存到新建文件。
- Comate侧边栏新增缩放配置，现在可以在Comate设置栏选择Comate侧边栏缩放比例，以适应您的IDE UI缩放配置。

**Version: v2.1.0 (2024-05-21)**

- Unit test continuation auto-identifies UT framework; when you are writing unit test methods, Comate automatically detects the current UT framework to recommend higher-quality code for you.
- Unit tests support Scala and Kotlin languages; you can right-click on selected code to generate corresponding unit test cases.
- The Comate sidebar input box supports switching input history with the up and down keys.
- Resolved the issue where Comate recommendations affected IDE built-in prompts during coding; now Comate's recommended code and the IDE's built-in prompts appear simultaneously; Tab accepts Comate's recommendation, and Enter accepts the IDE's built-in recommendation.
- For code content output in the sidebar, you can click "New File" to save it to a new file.
- The Comate sidebar has added zoom configuration; you can now select the Comate sidebar zoom ratio in the Comate settings to adapt to your IDE UI zoom configuration.

**版本：v2.0.6 (2024-05-17)**

- 修复Ubuntu及Windows系统中Comate侧边栏展示异常的问题。

**Version: v2.0.6 (2024-05-17)**

- Fixed display issues of the Comate sidebar in Ubuntu and Windows systems.

**版本：v2.0.5 (2024-05-10)**

- 修复侧边栏输入卡顿问题。

**Version: v2.0.5 (2024-05-10)**

- Fixed sidebar input lag.

**版本：v2.0.4 (2024-04-29)**

- Autowork使用#调起知识时，支持输入网页链接，将获取到的页面内容作为上下文供模型增加生成效果 侧边栏问答新增安全增强模式，可在配置页开启选项。
- 支持自动检测单测测试框架，优化单测续写体验。
- 优化侧边栏显示，侧边栏/help支持查看帮助文档。
- 开放平台插件性能优化。

**Version: v2.0.4 (2024-04-29)**

- When using "#" to invoke knowledge in Autowork, you can now input web links. The fetched page content will be used as context to enhance model generation results.
- Chat now has a new security enhancement mode, which can be enabled in the configuration page.
- Supports automatic detection of unit test frameworks to optimize the unit test continuation experience.

- Optimized sidebar display; the sidebar/help now supports viewing documentation.
- Open platform plugin performance optimization.

**版本 : v2.0.3 (2024-04-23)**

- 修复Autowork新增知识集异常。

**Version: v2.0.3 (2024-04-23)**

- Fixed an issue with exceptions when adding new knowledge sets in Autowork.

**版本 : v2.0.2 (2024-04-17)**

- 新建文件时推送预测代码，IDEA中新建Bean、Entity、Dao等文件时Comate会为您自动预测生成文件内容。
- 支持go语言单测续写能力。
- 开放平台内容支持流式方式输出。
- 跨文件分析算法优化。
- 侧边栏预加载，降低首次打开耗时。

**Version: v2.0.2 (2024-04-17)**

- Push predicted code when creating new files. When creating files like Beans, Entities, or DAOs in IDEA, Comate will automatically predict and generate file content for you.
- Supports Go language unit test continuation capability.
- Open platform content supports streaming output.
- Cross-file analysis algorithm optimization.
- Sidebar pre-loading to reduce initial opening time.

**版本 : v2.0.1 (2024-04-10)**

- 兼容241版本Jetbrains IDE。

**Version: v2.0.1 (2024-04-10)**

- Compatible with JetBrains IDE version 241.

**版本 : v2.0.0 (2024-04-01)**

- 正式接入Comate+开放平台，现支持通过插件扩展Comate能力，提供强大的开放性与定制性，让你的编程现场从未有过的强大。
- AutoWork：深度解读你的代码库，洞悉必要的背景知识，为你生成量身定制的代码。
- 代码推荐：支持java、go和python跨文件依赖上下文，让模型更懂项目代码。
- 通过“#”选择当前文件、代码库、目录、文件，或知识集，生成更符合诉求的代码。
- 新增单测续写能力，当您正在为某个方法编写单元测试时，Comate会准确定位到被测方法并为其流式续写单测，您可以按行采纳或全部采纳。
- 我们对对话区的样式进行了全新改版，现在的页面信息呈现更加清晰，界面更加清爽。

**Version: v2.0.0 (2024-04-01)**

- Officially integrated with the Comate+ Open Platform. It now supports extending Comate's capabilities through plugins,

offering powerful openness and customization, making your programming environment more powerful than ever.

- AutoWork: Deeply interprets your codebase and understands necessary background knowledge to generate tailor-made code for you.
- Code Recommendations: Supports cross-file dependency context for Java, Go, and Python, allowing the model to better understand your project's code.
- Use "#" to select the current file, codebase, directory, file, or knowledge set to generate code that better meets your needs.
- New Unit Test Continuation Capability: When you are writing unit tests for a method, Comate will accurately locate the method under test and stream unit tests for it. You can accept them line by line or all at once.
- We have completely revamped the style of the chat area. The page now presents information more clearly and offers a fresher interface.

#### 版本 : v1.3.5 (2024-03-01)

- 导航栏现在新增了“帮助文档”和“意见反馈”的入口，使您能更快速地获取帮助信息。
- 现在，我们的应用已经支持英文，为更多全球用户提供便利。
- 对话区代码操作现在支持新建文件和插入到终端，代码采纳过程更加顺畅。
- 对话框现在支持 # 引用多个文件或者文件来进行提问，已获得更准确的回答。
- 对停止生成功能的交互进行了优化，使其变得更加容易操作。
- 修复其他已知问题，提升用户体验。

#### Version: v1.3.5 (2024-03-01)

- Now you can reach "Docs" and "Feedback" at Chat.
- Our application now supports English for more global users.
- Chat area now supports creating new files and inserting into terminals, making the code adoption process smoother.
- Chat now support referencing multiple files or directories with # for more accurate answers.
- Optimized the interaction for stopping generation, making it easier to operate.
- Fixed other known issues, improving user experience.

#### 版本 : v1.3.4 (2024-02-02)

- 在Java和Go语言的单元测试中，新增上下文联想能力，大幅提升了测试用例的准确性
- 在GoLand中，我们新增了支持注释生成代码的功能
- 对Java单元测试生成效果进行了大量优化，显著提升了用例正确率，保障您的代码质量
- 在新的数据看板功能中，支持企业根据组织架构下钻研发数据，为您的决策提供更加详尽的数据支持
- 修复了因括号未闭合导致的问题，生成的代码更加健壮
- 修复了缩进和采纳存在的冲突问题，采纳体验更加丝滑
- 修复其他已知问题。

#### Version: v1.3.4 (2024-02-02)

- In the unit tests of Java and Go languages, the addition of context-awareness significantly enhances the accuracy of test cases.
- In GoLand, we have added functionality to support code generation from comments.
- Significant optimizations have been made to the generation of Java unit test results, greatly improving the correctness of test cases and ensuring the quality of your code.
- In the new data dashboard feature, support is provided for enterprises to drill down into development data based on organizational structure, providing more detailed data support for your decision-making.
- Fixed issues caused by unclosed parentheses, resulting in more robust generated code.
- Fixed conflicts between indentation and adoption, resulting in a smoother adoption experience.
- Fixed other known issues.

**版本 : v1.3.3 (2024-01-12)**

- 状态条动态展示 Comate 工作状态
- 优化代码补全场景，提升 coding 与推荐的流畅性
- 修复 Comate 推荐与 IDE 原生快捷操作冲突问题

**Version: v1.3.3 (2024-01-12)**

- The status bar dynamically displays Comate's working status.
- Optimized code completion scenarios to enhance the smoothness of coding and recommendations.
- Fixed conflicts between Comate recommendations and native IDE shortcut operations.

**版本 : v1.3.2 (2023-12-19)**

- 新增百度账号认证，扫码快速登录功能
- 修复 IDEA 假死的问题

**Version: v1.3.2 (2023-12-19)**

- Added Baidu account authentication, enabling quick login through QR code scanning.
- Fixed the issue of IDEA freezing.

**版本 : v1.3.1 (2023-12-15)**

- 支持根据 License 获取用户配置信息
- 优化插件配置页面
- 优化弱网环境代码推荐体验

**Version: v1.3.1 (2023-12-15)**

- Support retrieving user configuration information based on license.
- Optimize the plugin configuration page.
- Enhance code recommendation experience in weak network environments.

**版本 : v1.3.0 (2023-11-21)**

- 增强 Java 单测生成质量

- 优化侧边栏流式输出呈现形式
- 修复代码解释快捷指令的问题

**Version: v1.3.0 (2023-11-21)**

- Enhance the quality of Java unit test generation.
- Optimize the presentation format of sidebar streaming output.
- Fix the issue with the shortcut command for code explanation.

**版本 : v1.2.1 (2023-11-14)**

- 支持侧边栏的流式输出，内容输出更流畅
- 修复行间注释和函数拆分采纳失败的问题

**Version: v1.2.1 (2023-11-14)**

- Support streaming output in the Chat for smoother content display.
- Fixed the issue of failed adoption of line comments and function splitting.

**版本 : v1.2.0 (2023-10-27)**

- Python 语言增加“生成单测”功能
- 修改 Java 语言行前图标展示逻辑的问题

**Version: v1.2.0 (2023-10-27)**

- Added "Generate Unit Test" functionality to the Python language.
- Fixed the issue with the display logic of line prefix icons in the Java language.

**版本 : v1.0.0 (2023-10-24)** 欢迎使用 Baidu Comate，结合文心大模型，打造你的专属智能代码助手，具备以下能力：

- 行级/函数级实时续写：单行续写、多行续写、注释生成代码（函数注释）
- 代码生成：自然语言描述生成代码、增强生成
- 代码优化：长函数拆分代码重构、自然语言优化代码
- 单元测试生成：单测生成、接口文档生成测试代码
- 代码注释生成：函数注释、行间注释
- 智能 CLI
- 技术问答

**Version: v1.0.0 (2023-10-24)**

- Real-time Suggestions at the line/function level: single-line suggestions, multi-line suggestions, code generation from comments (function comments).
- Code generation: code generation from natural language descriptions, enhanced generation.
- Code optimization: refactoring of long functions, optimization of code through natural language.
- Unit test generation: unit test generation, API documentation generation for test code.
- Comment generation: function comments, inline comments.

- Intelligent CLI.
- Chat.

## VS端插件更新日志

### v3.21.0 (2025-12-01)

- 上下文检索逻辑升级：重构检索重排逻辑，提升代码库检索结果排序准确性与相关度。
- 上下文管理优化：统一对话输入框中添加上下文的操作方式，同时支持使用 + 号和 # 号，提升操作一致性与易用性。
- 代码库索引扩展：现已支持 Kotlin 文件的向量化处理，进一步提升代码理解与检索效果。
- MCP 能力升级：支持创建全局 MCP，可在多个 Workspace 间通用，降低重复安装成本，使用更高效便捷。
- 快捷指令优化：将「优化建议」指令升级为由 Zulu Agent 提供回答，更贴近上下文，代码融合更智能。
- GBK 文件兼容性增强：新增对 GBK 编码文件的注释与文件对比中的中文显示支持，提升编码文件编辑与对比体验。
- Agent 工具优化：改进读文件工具在处理失败时的容错机制，确保对话流程不受影响；优化写文件工具手动放弃操作，支持同时删除文件及其所在目录，提高操作的可靠性与一致性。
- 其他诸多交互与体验改进。

### v3.20.0 (2025-11-10)

- 全新 Agent 市场：内置丰富的垂类 Agent，可快速使用；同时提供统一设置页面，方便集中管理与展示。
- Agent 支持项目级共享：支持在项目范围内便捷共享自定义 Agent，方便团队复用经验与落地流程。
- Agent 支持基于 AI 创建：简化 Agent 生成流程，提升使用便捷性与效率。
- Rules 支持自动创建：支持基于 AI 自动创建规则，简化配置方式、提高落地效率。
- 单对话支持自动压缩：单次会话现支持上下文自动压缩，有效提升长对话场景下上下文的连续性。
- 对话输入框全面升级：全面优化上下文选择与状态展示，新增上下文推荐与检索能力，并进行了多项样式与细节调整，带来更智能、更顺畅的输入体验。
- MCP 效果提升：调整解析逻辑、增强模型输出容错能力，提升 MCP 输出内容有效性。
- Agent 工具优化：优化执行命令时的操作文案，采用更简洁直观的表达方式，提升交互体验。

### v3.19.0 (2025-10-27)

- 上下文检索全面升级：支持全新检索方式，检索速度更快，兼容更多文件类型与更大规模索引；同时新增可视化索引管理，检索体验更高效直观。
- Agent 支持语义检索：新增基于用户意图的语义检索能力，使 Agent 更精准理解需求，提升生成结果的质量与可靠性。
- Agent 上下文扩展：现支持将 DOCX 与 XLSX 文件作为上下文使用，提升模型对文档内容的理解与处理能力。
- Agent 工具执行效率提升：现已支持并行读取文件，大幅提升 Agent 的生成速度与效率。
- Agent 效果优化：修复了循环调用工具及过度主动触发的问题，提升 Zulu 的调用稳定性与响应准确性。
- MCP 入口优化：将 MCP 入口前置，用户可更快速访问与使用 MCP 功能，提升操作便捷性。
- MCP 功能优化：在添加 Server 时新增必填信息提示，有效提高安装和配置的正确率。
- 网络配置优化：修复了插件代理相关问题，提升代理网络下使用的稳定性与兼容性。

- 其他诸多体验优化和问题修复。

### v3.18.0 (2025-10-13)

- 全新升级为 Agent First 形态：Chat 与 Zulu 合并为统一 Agent 形式，提供丰富官方 Agent，深度支持问答、规划、编码等核心场景，同时支持便捷切换与对话内上下文共享，提升复杂任务处理效率。
- CodeLens 指令全面升级：自动切换最佳 Agent 回答，生成效果更精准。
- Agent 支持文档检索：在上下文类型中选择知识，Agent 在回答过程中会自动检索相关文档内容，显著提升生成质量。
- MCP 升级：现已支持在添加 MCP 后自动为所有官方 Agent 启用，使用更加便捷。
- 改写功能增强：现在即使未打开目标文件，也可支持跨文件跳转，提升多文件改写体验。
- 改写效果增强：独立维护多 Workspace 的编辑历史和诊断信息，使改写结果更精准可靠。
- 对话输入框体验优化：集中优化了输入框的交互体验，提升输入流畅度与整体操作舒适性。

### v3.17.0 (2025-09-15)

- Agent 检索工具视觉优化。移除卡片样式，界面更简洁直观，提升使用体验。
- Agent 上下文理解优化。策略升级，让系统在处理多段内容时更智能、更连贯，提高响应的准确性和相关性。
- Agent 工具安全增强。防止对敏感文件的非法写入与修改，提升系统安全性。
- Agent 性能优化。改进内存管理机制，降低超大文件读取时的内存占用，提升稳定性与性能。

### v3.16.0 (2025-09-08)

- 支持 Subagent，多智能体协同。主智能体可调度子智能体协同工作，自动完成任务拆解、分工与执行，显著提升复杂任务处理能力。

### v3.15.0 (2025-09-01)

- Zulu 重大升级，支持自定义智能体。支持可根据项目需求创建专属智能体，自定义能力与行为，打造深度契合个人习惯或团队流程的智能体。
- 新增 Plan Agent。支持交互式需求澄清，并将任务自动拆解为可执行步骤，生成 plan.md 文件，助力高效解决复杂任务。
- 新增 Todo List 模式。复杂需求可自动拆解为简单步骤，并按顺序动态执行，提升任务管理与执行效率。
- Zulu 支持联网搜索。新增 Web Search 工具，可实时获取网络信息，辅助代码开发与问题解决。
- Zulu 检索性能优化。通过改进遍历算法、支持本地索引以及意图识别检索，提升查询速度与结果精准度。
- 改写功能升级。现支持跨文件预测改写，可自动识别非当前文件的其他关联代码，提升批量修改与重构效率。
- 历史 Query 支持方向键快速选择。在输入框中可通过方向键快速选择历史查询记录，提升操作便捷性与输入效率。
- 预览图片优化。调整关闭按钮大小，避免过小导致难以点击，提升图片预览交互体验。
- 【企业版】支持个人级自定义提示词。企业和团队级提示词的权限在前端显性化展示，方便管理与使用。
- 【企业版】自定义模型权限优化。支持按照部门维度配置访问权限，实现更精细的模型管理与团队协作控制。

### v3.13.0 (2025-08-15)

- Zulu 智能体执行指令时支持复用相同终端。显著减少了每次操作中打开的新终端数量，提升使用连贯性与资源利用效率。
- 快速设置命令与 MCP 自动执行。在对话输出过程中，一键设置后续任务为自动执行，提升执行效率与体验连贯性。
- 本地代码库知识增强索引支持 SVN 仓库，助力更全面的上下文理解与智能生成。

- 支持导出对话中图片。对话中生成的图片可快速右键复制，便于保存与后续使用，提升使用效率。
- 【企业版】对话支持多模型配置。现已支持自定义模型供应商并支持使用千帆平台的其他大模型，可按需灵活切换，满足不同场景下的生成需求。
- 【企业版】Zulu 新增 MAX 模式选择。内置 SOTA 模型，生成效果更强、更精准。
- 【企业版】知识集权限体系全面升级，适配网页与插件端问答场景。知识集现已与部门架构绑定，支持企业级、部门级、个人级三种访问权限，插件端与 Web 端问答颗粒度与权限一致。
- 历史会话记录支持滚动加载，替代分页展示，浏览更顺滑。
- 修复其他已知问题，提升用户体验。

### v3.12.0 (2025-07-21)

- 超级补全体验升级。在生成预测代码后，系统将同时展示 Tab 跳转入口与改写内容，方便快捷定位修改位置与进行内容比对，提升改写效率与操作流畅度。
- 默认启用智能 Agent 模式。升级后，所有用户将优先使用 Zulu 智能体进行任务处理，享受更智能、更一致的使用体验；后续使用会记住您上一次选择的模式。
- 关联私域知识集。Comate 在提供代码补全建议时，会参考您的上传的私域知识集，生成更贴合团队规范的代码。

### v3.11.0 (2025-07-07)

- 超级补全支持智能 import (VScode)。随着代码编写，识别出需要引入的类、函数、组件时，智能提示所需的 import 语句，无需手动查找引用路径。
- 对话中选择上下文更直观便捷。我们优化了自动上下文提取逻辑，并提升了手动添加文件时的交互体验与界面样式，让上下文管理更高效、更可控。
- MCP 工具的使用和查看更方便。MCP 工具执行时修改为默认展开显示执行请求参数和返回结果，便于直观查看执行情况。

### v3.10.0 (2025-06-24)

- AI 代码审查 (Code Review)。问答功能支持 AI 代码审查。您可以选中一段代码、指定一个或多个代码文件、或一段代码变更 (diff)，Comate 将依据您的企业规范（上传至知识库中）或默认通用编码规范，进行代码审查。
- 智能采纳更准确。生成代码更好地融入当前上下文，减少手动调整。

### v3.9.0 (2025-06-03)

- 优化了从 Figma 设计稿生成前端代码的效果，您可以在 Zulu 对话框中通过输入斜线命令 / 使用此功能。
- 页面内置了精选的 MCP Server 列表，方便快速添加和管理。
- 支持创建和管理多条规则，可为每条规则设定不同的生效方式，包括自动生效、手动加入对话上下文、以及针对特定文件生效。此外，Zulu 规则还兼容您项目中已有的 Cursor rules 配置。
- 问答功能支持创建自定义的 Prompt（指令），您可以在问答对话框中输入斜线命令 / 快速调用这些预设指令。
- 在问答中，当 Comate 生成的代码包含新文件名时，系统将能正确创建并命名新文件。
- 当问答结果包含网络搜索信息时，系统会添加引文标注，遵循 Markdown 语法格式（例如 <sup>[1]</sup>）

### v3.8.0 (2025-05-19)

- Zulu 智能体现在支持自动执行识别到的命令，让任务执行更连贯。
- 在与 Zulu 智能体交互时，您可以将代码恢复到对话历史中的任一版本。

- Chat 功能增强。
- 您可以选中多个代码块作为上下文与 Chat 进行交互。
- 超级代码补全体验升级。
- 当触发“改写”时，生成的代码建议会拆分为两个步骤展示。您可以分步预览和确认，确保每一处修改都符合您的预期。

v3.7.0 (2025-04-24) 更新了产品 Logo，统一文心快码的品牌视觉体验。

- Zulu 现已支持 MCP，与您的应用无缝衔接。
- Zulu 支持创建和应用自定义规则，满足个性化编码需求。
- 优化调试体验，可在终端错误信息上右键快速触发调试选项。
- 代码超级补全优化：无论追加、修改代码，还是采纳前后，您的光标位置始终保持不变。
- Tab 键采纳的提示框与光标保持同一行，操作更流畅。
- 新增代码部分以幽灵字符展示（无绿色背景），采纳后以绿色高亮显示，清晰直观。

v3.6.0 (2025-04-16)

- 修复已知问题，提升用户体验。

v3.4.0 (2025-03-06)

Zulu智能体发布公测：自动编程智能体，提供从代码生成到业务理解的完整智能服务。

**应用场景：**

- 全自然语言实现需求，不写一行代码。
- 理解代码库业务逻辑，帮助你迅速上手新项目。
- 帮你自动搭建开发环境和启动服务，不用一行一行输入命令。

**产品功能：**

- 多工具自主调用：灵活调用多种实用工具，涵盖生成、执行命令等各个方面。
- 跨文件智能编辑：支持同时处理多个文件，智能同步修改关联代码。
- Diff 代码清晰展示：流式 diff 实时展示代码变化，更有效地进行代码对比和审查。
- 多模态高效交互：支持在会话中添加图片，让开发场景更加多元便捷。
- 代码效果实时预览：一键生成网页/游戏预览链接，浏览器实时呈现运行效果。

**新功能上线**

- 接入 DeepSeek 模型，需升级为付费用户。
- 新增超级代码补全，可自动修正拼写错误，自动重命名变量，精准预测下一处要编辑的光标位置，智能重写目标位置的多行代码，需手动开启。
- 新增智能采纳功能，侧边栏问答生成的代码 diff 可以准确的合并到文件中。

**产品改进**

- 提升函数注释、代码解释和调优建议的生成效果。
- 优化特定场景 Java 依赖图检索策略，提升续写效果。

- 优化了问答对话中止、状态处理等边界情况的处理逻辑，提升整体稳定性。
- 优化代码样式，调整行内代码颜色以匹配编辑器配色。
- 针对 Windows 系统优化进程创建策略，减少资源占用。
- 兼容 Windows 7 系统，满足更多用户的使用需求。

#### v3.3.0 (2025-01-03)

- 优化全链路性能，带来更流畅的交互体验。
- 修复其他已知问题，提升用户体验。

#### v3.2.0 (2024-12-30)

- 修复其他已知问题，提升用户体验。
- Fixed other known issues to enhance user experience.

#### v3.1.0 (2024-12-18)

- 对话支持可选更快模型、更强模型，基础模型全新升级到128K。
- 安全智能体新增支持指定单文件、目录扫描。
- 修复其他已知问题，提升用户体验。
- Conversations now support optional faster and more powerful models, with the base model upgraded to 128K.
- The security agent now supports specifying single files and directory scanning.
- Fixed other known issues and improved user experience.

#### v3.0.0 (2024-11-12)

开发全流程升级，发布编码智能体、Debug智能体、单测智能体、安全智能体、全栈编程智能体。

Complete development workflow upgrade, introducing Development Agent, Debug Agent, Unit Test Agent, Security Agent, and Full-Stack Programming Agent.

#### v2.17.0 (2024-10-28)

- 修复其他已知问题，提升用户体验。
- Fixed other known issues and improved user experience.

#### v2.16.0 (2024-10-14)

- 优化首页推荐问题内容。
- 修复其他已知问题，提升用户体验。
- Optimized recommended question content on the homepage.
- Fixed other known issues and improved user experience.

#### v2.15.0 (2024-09-30)

- 行间对话交互优化：更佳输出效果。
- Paddle 插件体验优化，新增指令代码转换的功能指引更清晰。
- 修复其他已知问题，提升用户体验。

- Improved inline dialogue interaction: Better output results.
- Enhanced Paddle plugin experience, with clearer guidance for new instruction code conversion functionality.
- Fixed other known issues and improved user experience.

#### v2.14.0 (2024-09-20)

- 从终端报错跳转到对应代码行时提供分析与修复的快捷按钮，直面问题现场。
- Paddle 插件支持根据目录将文件中 Pytorch 框架转化为飞桨框架。
- 提问后增加关联问题建议，开拓提问角度和思路，让提问更深入和全面。
- 行间对话交互优化：更佳的流式输出效果，更准确的触发范围识别。
- 修复其他已知问题，提升用户体验。
- Added a CodeLens to provide fix suggestions for the line of code referenced in terminal error stacks.
- Added support for translating code written with PyTorch to the PaddlePaddle framework in the Paddle plugin.
- Added follow-up question suggestions after initial queries, broadening the scope and perspective of inquiries for more comprehensive and in-depth understanding.
- Improved the inline chat experience: smoother edit streaming in your editor and more accurate target code selection.
- Fixed other known issues to enhance user experience.

#### v2.13.0 (2024-09-03)

- 行间对话生成的代码支持流式输出了，更快看到编辑的效果，提升协同效率。
- 支持手动触发代码库索引构建，当回复未能参考代码库最新的代码时，可以手动触发更新。
- 对话侧边栏迎来样式全面升级，减少干扰元素，功能指示更清晰，便于专注在内容上。
- 优化对话回复中的 Mermaid 流程图样式，提升阅读体验。
- 修复其他已知问题，提升用户体验。
- Code generated in inline conversations now supports streaming output, allowing you to see editing effects faster and improving collaboration efficiency.
- Added support for manually triggering codebase index building. When responses don't reference the latest code in the codebase, you can manually trigger an update.
- The conversation sidebar has received a comprehensive style upgrade, reducing distracting elements and providing clearer function indicators for better focus on content.
- Improved the style of Mermaid flowcharts in conversation replies, enhancing the reading experience.
- Fixed other known issues to improve user experience.

#### v2.12.0 (2024-08-19)

- Comate 行间对话更新了，支持对划选的代码进行提问或编辑，清晰展示代码变更，快使用快捷键 `cmd+I(macOS)` / `ctrl+I(Windows)` 体验吧！
- 新增【智能生成 commit message】
- 新增「添加日志」按钮，一键为函数内部增加打印日志，便于 debug。

- 光标悬浮在任意插件时，浮窗显示更详细的插件信息和快捷提问按钮，插件能力更清晰。
- 光标悬浮在不同知识类型时会浮窗显示详细信息、并提供了快捷示例，激发灵感。
- 对话消息过多后自动隐藏历史消息，滚动到顶部后再加载，减小渲染开销。
- 提升首次打开对话侧边栏时的显示速度，提问 0 距离。
- 修复其他已知问题，提升用户体验。
- Comate inline dialogue has been updated, supporting querying or editing selected code, clearly displaying code changes. Try it with the shortcut cmd+I (macOS) / ctrl+I (Windows)!
- Added "Smart commit message generation"
- Added "Add log" button, allowing one-click addition of print logs inside functions for easier debugging.
- When hovering over any plugin, a floating window displays more detailed plugin information and quick question buttons, making plugin capabilities clearer.
- When hovering over different knowledge types, a floating window displays detailed information and provides quick examples, inspiring creativity.
- Automatically hides historical messages when there are too many dialogue messages, loading them again when scrolling to the top, reducing rendering overhead.
- Improved display speed when opening the dialogue sidebar for the first time, enabling zero-delay questioning.
- Fixed other known issues and improved user experience.

#### v2.11.0 (2024-07-31)

- 新增【续写状态】展示，实时展示生成状态。
- 优化多轮对话体验，生成更连贯的上下文。
- Diff 视图状态下支持一键采纳。
- 修复其他已知问题，提升用户体验。
- Added the display of "Continuation Status" to showcase the real-time generation status.
- Enhanced the multi-turn dialogue experience for more coherent context generation.
- Introduced one-click adoption support in Diff view mode.
- Fixed other known issues to improve user experience.

#### v2.10.0 (2024-07-17)

- 提升对话以及续写的速度。
- 修复其他已知问题，提升用户体验。
- Improved the speed of conversations and continuations.
- Fixed other known issues to enhance user experience.

#### v2.9.0 (2024-07-01)

- 使用能力更精准快捷，现在它不仅支持传统的关键字搜索，还增加了拼音搜索功能。
- 优化官方知识集展示形态。

- 修复其他已知问题，提升用户体验。
- Enhanced search capabilities are now more precise and efficient, supporting not only traditional keyword search but also pinyin search.
- Optimize the presentation form of the official knowledge base
- Fixed other known issues to enhance user experience.

#### v2.7.0 (2024-06-19)

- 新增【智能Debug】功能。当编译运行报错时，通过快捷按钮或勾选错误日志，一键定位错误并给出修复代码。
- 新增【代码安全】功能，企业版用户在管理员开启该功能后可在侧边栏中使用。插件会扫描代码中的安全漏洞，并提供修改建议。用户采纳后可一键修复安全漏洞。
- 支持混合云服务部署方案。
- 优化Go语言续写推荐效果。
- 优化用户个人账号信息的展示。
- 修复其他已知问题，提升用户体验。
- New AI Debug Feature: Easily locate and fix errors with a single click using shortcut buttons or by selecting error logs when compilation or runtime errors occur.
- New Code Security Feature: Available in the sidebar for enterprise users once enabled by the administrator. This plugin scans for security vulnerabilities in the code and provides suggestions for fixes. Users can apply these suggestions and resolve security issues with one click.
- Support for Hybrid Cloud Service Deployment.
- Enhanced Go Language Inline Suggestion.
- Improved Display of Personal Account Information.
- Fixed other known issues, improving user experience.

#### v2.6.0 (2024-06-04)

- 修改侧边栏启动样式：新增使用引导，帮助用户快速熟悉侧边栏功能。
- 续写推荐方式可配置：允许用户在设置中自定义续写推荐的方式，选择单行或多行优先。
- 注释生成代码交互优化：注释生成代码时，增加实时加载状态提示。
- 修复其他已知问题，提升用户体验。
- Improved the on-boarding experience of the chat interface: Added a few examples on the welcome page to help you quickly explore our latest features.
- Customize auto-completion setting: Allow users to customize how continuation recommendations are displayed in settings, choosing between single-line or multi-line prioritization.
- Optimize comment-to-code generation interaction: Provide real-time loading status when generating.
- Fixed other known issues, improving user experience.

#### v2.5.0 (2024-05-21)

- 搜索功能升级：输入「#」后选择「网络搜索」，可以轻松在侧边栏中进行网络搜索，获取更多相关信息。

- 历史提问便捷切换：通过键盘上的上下键，可以在侧边栏中快速切换历史提问内容，优化使用体验。
- 修复其他已知问题，提升用户体验。
- Improved Search: You can now search the web by typing "#" and selecting "Web Search" in the sidebar.
- Quickly Switch Historical Questions: Use the up and down arrow keys to switch between previous questions in the sidebar.
- Fixed Other Known Issues: We've resolved some known problems to enhance your experience.

#### v2.4.0 (2024-04-30)

- 在 Vue 文件中，JS/TS 代码中的函数支持通过快捷按钮，快速生成注释、生成单测等能力。
- 新增了对 JS/TS 代码中箭头函数的快捷按钮的支持。
- 在侧边栏中新增唤起「快捷指令」、「插件」和「知识」的按钮。
- 对调优建议指令进行了针对性优化，效果提升明显。
- 修复其他已知问题，提升用户体验。
- In Vue files, functions within JS/TS code now support quick generation of comments, unit tests etc through shortcut buttons.
- Added support for shortcut buttons in JS/TS code for arrow functions.
- New buttons for invoking "Commands", "Agents", and "Context" have been added to the sidebar.
- The optimization suggestion command has been specifically optimized, resulting in a significant improvement.
- Fixed other known issues, improving user experience.

#### v2.3.0 (2024-04-17)

- 对英文版做了进一步的支持，为全球用户提供智能编码服务：英文版个人官网、用户手册、更新日志上线；插件内支持英文使用；AutoWork高级生成能力支持英文对话。
- 在Comate中可以关联网页了，支持更个性化的问答：你可以通过在侧边栏中通过「#」关联你的URL链接，Comate可以帮你深度解读信息，并生成更贴合你需求背景的回答。
- 发布「工具箱」插件，支持Base64编解码、JSON和YAML互转等能力：你可以在侧边栏中通过「@」唤起「工具箱」插件，或者通过「/」直接调起快捷指令。
- 修复其他已知问题，提升用户体验。
- Further support has been added for the English version, providing intelligent coding services for global users: The English version of the personal profile, quick start, and change log are now available. English language is supported within the extension, and AutoWork's advanced generation capability now supports English chat.
- Now you can now link web pages, allowing for more personalized question and answer experiences in Baidu Comate: By using "#" in the sidebar, you can associate your URL links, enabling Comate to deeply interpret information and generate answers that are more tailored to your specific needs and background.
- We've released the "Toolbox" plugin, which supports Base64 encoding/decoding, JSON and YAML conversion, and more: You can summon the "Toolbox" plugin in the chat using "@" or directly activate shortcut commands using "/".
- Fixed other known issues, improving user experience.

#### v2.2.0 (2024-03-31)

- 现在，我们的应用已经支持英文，为更多全球用户提供便利。

- 对话区代码操作现在支持新建文件和插入到终端，代码采纳过程更加顺畅。
- 对话框现在支持 # 引用多个文件或者文件来进行提问，已获得更准确的回答。
- 对停止生成功能的交互进行了优化，使其变得更加容易操作。
- 修复其他已知问题，提升用户体验。
- Our application now supports English for more global users.
- Chat area now supports creating new files and inserting into terminals, making the code adoption process smoother.
- Chat now support referencing multiple files or directories with # for more accurate answers.
- Optimized the interaction for stopping generation, making it easier to operate.
- Fixed other known issues, improving user experience.

#### v2.1.0 (2024-03-20)

- 新增飞桨PaddlePaddle官方插件，支持代码生成，智能问答。
- 新增百度智能小程序官方插件，支持代码生成，智能问答。
- 现在，我们支持在消息流式输出过程中对代码进行操作，可以更快的采纳或复制代码。
- 对消息的渲染性能进行了优化，现在渲染长消息的速度提升了4倍，有效地解决了滚动卡顿的问题。
- 对采纳操作和Tab键的交互逻辑进行了智能化处理，让您在使用过程中享受到更流畅的用户体验。
- 我们对代码库相关检索逻辑进行了深度优化，提升了检索效率。
- 修复其他已知问题，提升用户体验。
- Added official plugin PaddlePaddle, supporting code generation and intelligent Q&A.
- Added official plugin 百度智能小程序, supporting code generation and intelligent Q&A.
- Now supports code manipulation during message stream output for faster adoption or copying of code.
- Optimized rendering performance of messages, resulting in a 4x speed increase in rendering long messages, effectively addressing scroll lag issues.
- Intelligently handled the interaction logic for adoption actions and Tab key, providing a smoother user experience.
- Deeply optimized code library related search logic, improving retrieval efficiency.
- Fixed other known issues, enhancing user experience.

#### v2.0.0 (2024-03-01)

- 我们正式发布了Comate+开放平台，现支持通过插件扩展Comate能力，提供强大的开放性与定制性，让你的编程现场从未有过的强大。
- ——发布多个插件，通过Git插件连接Github、Gitee、GitLab等多个常用工具，为你智能生成提交信息、创建Issue，快来通过【@】【/】【#】在Comate侧边栏中尝试最新的智能化开发吧。
- ——上线插件配置中心与知识中心。Comate的能力可以通过配置呈现更贴合你需求的效果；你还可以通过知识中心上传私域知识，让Comate更懂你，生成更个性化、更准确的内容与代码。
- ——通过丰富的开放基座，企业可以通过Comate的插件形式将自身服务、能力、知识等连接到编程现场，增强推荐效果、提供个性化能力，为组织量身打造自有智能代码助手。

- 上线了AutoWork，它能深度解读你的代码库，洞悉必要的背景知识，为你生成量身定制的代码。
- 新增对XML语言的智能补全提示，让您在编写XML文件时也能得到便利的帮助。
- 现在，当我们进行多行代码推荐时，您可以选择逐行采纳，让您根据需要更精准地把握每一行代码。
- 新增了插件内的小调研功能，让我们可以及时了解您对各项功能的满意度，便于我们进行针对性的优化。
- 导航栏新增了意见反馈入口，我们随时关注并解决用户遇到的问题。
- 我们对对话区的样式进行了全新改版，现在的页面信息呈现更加清晰，界面更加清爽。
- 代码检索策略进行了优化，提高您查找代码的速度和准确性，对检索和生成的界面进行了视觉效果升级，使用更加方便并拥有更好的视觉感受。
- 修复其他已知问题，提升用户体验。
- Officially launched the ComatePlus platform, now supporting extension of Comate capabilities through plugins, providing powerful openness and customization, making your programming scene more powerful than ever.
- Released multiple plugins, such as Git plugins connecting to Github, Gitee, GitLab, etc., intelligently generating commit messages, creating issues. Try out the latest intelligent development through `[@] [/] [#]` in the Chat panel.
- Launched Plugin Configuration Center and Docs Center. Comate's capabilities can be presented more fittingly through configuration; you can also upload domain-specific docs through the Docs Center, making Comate understand you better and generate more personalized, accurate content and code.
- Enterprises can connect their own services, capabilities, and knowledge to the programming scene through the rich open base of Comate in the form of plugins, enhancing recommendation effects, providing personalized capabilities, and tailoring intelligent code assistants for organizations.
- Launched AutoWork, which deeply interprets your codebase, understands necessary background docs, and generates customized code for you.
- Added intelligent autocompletion prompts for XML language, providing convenient assistance when writing XML files.
- Now, when performing multi-line code suggestions, you can choose to accept line by line, allowing you to grasp each line of code more accurately as needed.
- Added a mini survey feature within plugins to promptly understand your satisfaction with various features, facilitating targeted optimization.
- Added a feedback entry to the navigation bar, where we constantly monitor and address issues encountered by users.
- Completely redesigned the style of the Chat area for clearer page information and a fresher interface.
- Optimized code retrieval strategy, improving the speed and accuracy of code lookup, and upgraded the visual effects of retrieval and generation interfaces for easier use and better visual experience.
- Fixed other known issues, enhancing user experience.

#### v1.14.1 (2024-02-02)

- 迎新年，贺新春，龙年主题活动开启，参与活动有机会赢积分、兑京东卡！
- 对话区顶部新增“帮助文档”和“设置”两个入口，便于您更加方便地获取帮助和进行配置。
- 对插件的分类和标签进行更新，使您在搜索 Comate 插件时更加轻松
- 我们将文案“自然语言生成代码”更改为“开始对话”，使其更容易理解
- 我们增加了消息的前置判断，防止在未输入有效内容时发送了消息

- 对消息的渲染样式进行了优化，以呈现更好的视觉效果
- 修复其他已知问题
- Celebrate the new year with the Dragon Year theme event! Engage for an opportunity to earn credit scores or redeem JD cards!
- Added "Help" and "Settings" entries at the top of the Chat area for easier access to help and configuration.
- Updated plugin categorization and tags to make it easier for you to search for Comate plugins.
- Changed the phrase "Natural Language Code Generation" to "Start Chat" for easier understanding.
- Added pre-checking for messages to prevent sending messages when no valid content is entered.
- Optimized message rendering styles for better visual effects.
- Fixed other known issues.

#### **v1.13.11 (2024-01-19)**

- 优化代码生成异常时的提示信息
- 修复部分 C++ 语言的文件不显示函数功能按钮的问题
- 修复侧边栏流式输出问答偶现输出中断的问题
- Enhanced error messages for code generation exceptions.
- Fixed the issue where certain C++ language files were not displaying buttons above functions.
- Resolved intermittent interruptions in the Chat panel output.

#### **v1.13.1 (2024-01-02)**

- 新增数据看板，助力企业研发效率提升
- 购买会员权益后可直接扫码登录使用，无需配置 License
- 优化代码解释功能的上下文获取策略，提升解释准确率
- Introduced a data dashboard to boost enterprise development efficiency.
- VIP Members can now directly log in via QR code after purchasing membership benefits, eliminating the need for license configuration.
- Optimized context retrieval strategies for code interpretation, enhancing accuracy.

#### **v1.13.0 (2023-12-19)**

- 支持百度账号认证，扫码快速登录
- Jupyter Notebook 文档支持代码续写功能
- 优化 Comate 推荐和 IDE 自带推荐框冲突的问题
- 修复链接远端机器时不显示函数功能按钮的问题
- Added support for Baidu account authentication, enabling quick login via QR code scanning.
- Jupyter Notebook documents now support code suggestions.
- Resolved conflicts between Comate suggestions and built-in IDE suggestion boxes.

- Fixed the issue where buttons above functions were not displayed when linking to remote machines.

#### v1.12.1 (2023-12-05)

- 优化侧边栏流式输出问答时，自动向下滑动的体验问题
- 优化打开侧边栏后输入框自动聚焦的问题，提升问答体验
- 修复函数注释结果的行首空格问题
- 修复侧边栏没有历史消息时仍可以清空消息的问题
- 修复获取默认用户名失败时插件激活失败的问题
- Improved the user experience of automatic downward scrolling during Chat output streaming.
- Enhanced the focus behavior of the input box upon opening the Chat, improving the chat experience.
- Fixed the leading whitespace issue in function comment results.
- Fixed the ability to clear messages in the Chat even when there are no historical messages.
- Resolved the issue of plugin activation failure when the default username retrieval fails.

#### v1.2.0 (2023-11-21)

- 增加侧边栏会话中查看生成代码变更的功能
- 增加生成结果自定义反馈的功能
- 增加敏感信息过滤的功能
- 支持侧边栏的流式输出，内容输出更流畅
- 支持行中推荐
- Added functionality to view code changes in Chat.
- Introduced customizable feedback for generated results.
- Implemented sensitive code filtering capabilities.
- Enabled smooth streaming output in the Chat, enhancing content delivery.
- Supported suggestions in the middle of line.

#### v1.1.0 (2023-11-07)

- 优化存在推荐内容时删除字符不流畅的体验问题
- 优化取消本次推荐命令并默认绑定快捷键的设置问题
- 修复单测生成不支持 C++ 语言的问题
- 修复获取不到设备 ID 的问题
- 修复侧边栏生成结果部分语言没有语法高亮的问题
- Improved the user experience of deleting characters when suggestion exist.
- Streamlined the process of canceling suggestions and default shortcut key bindings.
- Fixed the issue where unit test generation did not support the C++ language.
- Resolved the problem of failing to retrieve device IDs.

- Fixed the lack of syntax highlighting for some languages in the Chat results.

#### v1.0.0 (2023-10-24)

- 行级/函数级实时续写：单行续写、多行续写、注释生成代码（函数注释）
- 代码生成：自然语言描述生成代码、增强生成
- 单元测试生成：单测生成、接口文档生成测试代码
- 代码优化：长函数拆分代码重构、自然语言优化代码
- 代码注释生成：函数注释、行间注释
- 智能 CLI
- 技术问答
- Real-time Suggestions at the line/function level: single-line suggestions, multi-line suggestions, code generation from comments (function comments).
- Code generation: code generation from natural language descriptions, enhanced generation.
- Unit test generation: unit test generation, API documentation generation for test code.
- Code optimization: refactoring of long functions, optimization of code through natural language.
- Docstring generation: function docstring, inline comments.
- Intelligent CLI.
- Technical Q&A.

## Xcode端插件更新日志

### Comate 4.0版本正式发布！

[下载：文心快码Comate](#)

#### 过去，你可能遇到过这些抓狂时刻

- 陷入死循环 —— “为什么它又陷入了死循环，重复同样一句话？”
- 工具成摆设 —— “明明项目代码都摆在那里，为什么都不搜一下、不会却硬答？”
- 鸡同鸭讲 —— “指令说得很清楚，为什么生成的却不是我想要的？”
- 选择性失忆 —— “任务多问几轮就忘了前文”
- 回答没有重点 —— “生成过程又慢又卡，满屏输出没有重点”
- 能力跟不上 —— “业内先进能力、先进模型，什么时候能用上？”

每一个反馈、每一条吐槽，我们都认真看过了

这些声音让我们意识到：如果一个工具不能在关键时刻让人放心，那它就还不够好

所以，我们做了一个决定：不搞小修小补，直接对 Comate 的底层逻辑进行系统级重构

#### 🔄 更新方式

- Comate AI IDE：打开Comate IDE → 点击更新提示 → 重启（最新版本 V1.0.0）

- VS Code：拓展 → 已安装 → Comate → 更新 → 重启（最新版本 V4.0.0）
- JetBrains 系列：设置 → Plugins → Installed → Comate → Update → 重启（最新版本 V4.0.0）
- 其他 IDE：Xcode、Comate CLI，紧张开发中，敬请期待

🔗 Comate 4.0版本正式发布！七大能力升级，一次看懂

🌟 版本升级公告

## Comate 4.0 Agent 核心能力升级

底层能力 × 交互体验 × 系统稳定性 = 全新协作标准

The infographic is divided into three main sections:

- 核心能力跃升 (Core Capability Leap):**
  - 推理增强: 复杂任务拆解 + 长对话稳定 + 多模式支持
  - 架构扩展: Skill生态 + MCP渐进注册
  - 交互升级: Web检索 + Slash快捷指令
  - 推理链路重构 & 长对话优化
  - Explore Subagent: 语义级代码检索
  - 多模式输入 (图文混合)
  - Skill机制 (全局/项目级, 热插拔)
  - MCP架构重构 (低Token、高稳定)
  - Web工具: 实时网页解析 + 检索
  - Slash指令体系 (/ 快速调用)
- 体验优化 (Experience Optimization):**
  - 输出更轻: 关键信息强化, 冗余内容精简
  - 展示更清: Subagent层级重构
  - 操作更快: 输入框 & 终端一体化
  - 等待更轻: 随机Loading文案
  - Agent输出: 高密度、低噪音
  - Subagent渲染: 视觉层级清晰
  - 输入体验: 模型选择直观 + 输入交互升级
  - 终端操作: bash 一键插入执行
  - Mermaid: 渲染后支持复制代码
  - Loading: 动态文案缓解焦虑
- 稳定性与性能提升 (Stability and Performance Improvement):**
  - 任务可控: 回滚 + 异常重建
  - 对话不中断: SSE超时优化
  - 效率提升: 查询/响应双提速
  - 数据可靠: 文件编辑即时落盘
  - 任务回滚 & subtask自动重建
  - 长对话SSE稳定性优化
  - 消息处理性能提升
  - 文件编辑持久化修复
  - 其他问题修复

### 01 全面重构推理与执行能力

“大脑” (System Prompt) → “手脚” (Tools) → “知识” (Context) 全面进化

**任务理解更准确**：重写 System Prompt

- 执行意图的把握更精准
- 强化任务规划和管理
- 优先技术真实性，不迎合、有创新

**工具调用更合理**：重新18+工具定义与工具调用逻辑

- 调用工具更精准
- 复杂任务的多工具协作更流畅
- 调用成功率大幅提升

**多轮长对话记忆增强**：重写上下文压缩逻辑

- 多轮对话衔接更自然
- 不丢失关键信息
- Token 消耗降低

### 02 全面重构 Subagent、MCP

**全新创建方式**

- 通过 /create-subagent 引导智能创建

- 支持个人级、项目级 Subagent，文件方式存储，同步与迁移都更便捷

#### 调度更智能

- 根据任务特征自动匹配最优 Subagent
- 支持更细粒度的任务委派，复杂任务自动拆解

#### 执行更稳定

- 调用链路全面加固，子任务失败率大幅降低
- 新增异常恢复机制，子任务断线后自动重建上下文

## 03 全面支持 Skills (Tag : New)

Comate 支持使用 Skill 固化常用能力

#### 创建更智能

- 通过 /create-skill 引导智能创建
- 支持个人级、项目级 Subagent，文件方式存储，同步与迁移都更便捷

#### 使用更灵活

- 根据任务特征自动匹配最优 Skill，也可手动选择所需 Skill
- 内置 5 个系统级 Skill，开箱即用
- 完全兼容社区 Skill 协议，支持 Claude Code、Cursor、Opencode等 20 多个工具

## 04 全新 Explore Subagent

Comate 内置的专注于检索的 Subagent

#### 代码库探索更深度

- 告别 grep/glob 简单匹配，强化语义检索，自主推理相关代码
- 自动识别代码深度理解场景、自动调起
- 分析代码架构、识别错误等典型场景下表现更佳

#### 效率更极致

- 秒级索引构建，支持大型代码库
- 编排多任务、多工具并行探索，最短时间最大覆盖
- 显著降低主 Agent 上下文负载、大幅降低 Token 消耗

## 05 全新 Slash

Comate 支持使用 「/」快速唤起系统级能力

#### 内置 5 个系统级 Skill

- /create-subagent秒建子 Agent

- /create-skill 自定义能力
- /create-rule 配置专属规则
- /figma2Code : 设计稿一键转代码
- /page-Builder : 创建美观现代化的网页

#### 内置 3 个系统级 Commands

- /code-review 基于变更评审代码并给出修复建议
- /debug 深度分析项目，解决项目报错
- /unit-test 有效检测 bug、提高单元测试覆盖率

#### 快速调起 Subagents

- 自定义的 Subagent 也可通过/直接调起

## 06 速度更快、Token更省

#### 效果更稳定

- 基于 Skill 「渐进式」原则重构 MCP
- 调用链路标准化、流程更稳定
- 新增调用失败自动重试，稳定性大幅提升
- 多端复用层统一实现，VSCode、JetBrains 体验一致

#### Token 更节省

- 按需加载 MCP 服务与对应工具，避免无效工具描述传入模型
- 精准的 Tool Result 格式控制，避免冗余信息挤占 Token

#### 生态更开放

- 内置百度专属 MCP 服务，开箱即用
- 支持 HTTP 传输方式，接入门槛更低，兼容更多服务

## 07 全新视觉：让复杂变简单

这次我们把界面聚焦到三件事：

- 任务流程
- 思考过程
- 执行结果

结构更清晰，逻辑更可追踪

[下载：文心快码Comate](#)