华为云码道 CodeArts 代码智能体

用户指南（码道 CLI）

文档版本 01

发布日期 2026-07-31

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

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，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为云计算技术有限公司

地址： 贵州省贵安新区黔中大道交兴功路华为云数据中心 邮编：550029

网址： https://www.huaweicloud.com/

目 录

1 什么是码道 CLI 1
2 安装和升级 3
2.1 前提条件 3
2.2 安装 4
2.3 授权 5
2.4 升级 9
3 权限 10
4 自定义命令 13
5 沙箱 17
6 智能体 26
7 技能 32
8 Hooks 35
9 代码库索引 41
10 LSP 46
11 定时任务 48
12 自定义模型 57
13 MCP 63
13.1 默认 MCP 配置 63
13.2 Claude Code 格式 67
14 TUI 72
14.1 斜杠命令 72
14.2 快捷键 74
14.3 SDD 75
15 CLI 80
15.1 命令 80
15.2 MCP 命令 89

1 什么是码道 CLI

产品概述

华为云码道CodeArts代码智能体CLI（CodeArts Agent CLI，码道CLI）是面向开发者的智能编码助手，具备代码编写、代码分析、代码优化等全链路智能化能力。

产品提供终端交互界面（TUI）与命令行模式（CLI）使用形态，可无缝融入各类研发工作流，适配多样化开发场景。无桌面环境下，TUI可视化操作便捷编写调试代码，CLI靠命令快速完成编译。

表 1-1 TUI 与 CLI 区别

特性	TUI（终端交互界面）	CLI（命令行模式）
交互方式	对话框输入，可视化操作	直接传参数，一行命令执行
适用场景	交互式开发、代码编写与调试	脚本自动化、批量任务
推荐人群	新手用户、日常开发	有命令行经验的开发者

支持的操作系统

- Windows：推荐使用Windows 11 (x64)；如果使用Windows 10 (x64)，系统版本需为2019年及以上，建议升级至最新稳定版本。
- macOS：macOS 11及以上版本，兼容ARM64（Apple Silicon）架构。
- Linux：Huawei Cloud EulerOS 2.0、SUSE Linux Enterprise Server 12 SP5、Ubuntu 18.04/20.04/22.04/24.04 LTS、Debian 10/11/12、CentOS 8、RHEL 8/9。

核心功能

- 命令覆盖：提供models（模型配置）、sessions（管理会话）等核心命令，支持完整生命周期管控。
- 脚本化集成：支持通过命令参数直接传递任务，也可以将命令写入脚本文件（如Shell脚本、Python脚本）自动执行，实现AI任务的自动化与批量处理。

产品优势

- 双交互形态覆盖：同时提供终端交互界面（TUI）与命令行模式（CLI），兼顾交互式可视化操作与自动化脚本调用，适配不同使用习惯与业务场景。
- 智能可扩展代理体系：内置Build、Plan等核心智能代理，同时支持自定义代理编排，灵活适配复杂项目开发与任务拆解需求。
- 丰富生态能力集成：原生支持MCP服务器、技能、自定义智能体等扩展能力，可深度联动各类开发工具与业务能力。
- 灵活规范化配置：兼容JSON、JSONC格式配置文件，配置结构清晰、可读性强，便于本地调试、团队同步与工程化托管。

2 安装和升级

2.1 前提条件

支持的操作系统

为确保最佳兼容性与性能表现，推荐使用以下系统配置：

- Windows操作系统：**推荐使用Windows 11 (x64)**；如果使用Windows 10 (x64)，系统版本需为2019年及以上，建议升级至最新稳定版本。
- macOS操作系统：macOS 11及以上版本，兼容ARM64（Apple Silicon）架构。
- Linux：Huawei Cloud EulerOS 2.0、SUSE Linux Enterprise Server 12 SP5、Ubuntu 18.04/20.04/22.04/24.04 LTS、Debian 10/11/12、CentOS 8、RHEL 8/9。

购买套餐

步骤1 在购买码道CLI前，需要参考如下内容，**申请华为账号**。

1. 进入**华为云官网**，单击页面右上角的“注册”。
2. 参考**注册华为账号并开通华为云**中操作，完成注册。

步骤2 **如需开通基础版或专业版套餐，请在购买套餐前完成实名认证。**

进入**华为云码道CodeArts代码智能体购买页**，开通套餐。

- 开通套餐的账号拥有企业管理员权限，可继续**安装**码道CLI，开启使用。
- 若您的账号为华为账号，需IAM用户使用码道CLI，请先使用**华为账号创建IAM用户**，**企业管理员**再将其**添加为企业成员并启用席位**。

----结束

添加企业成员并启用席位

步骤1 使用**华为账号**登录**华为云码道代码智能体**页面。

步骤2 在左侧导航栏选择“企业管理 > 成员及团队管理”，进入“成员及团队管理”页面。

步骤3 单击右上角的“添加企业成员”，弹出“添加企业成员”对话框。

步骤4 参考表2-1配置企业成员信息。

添加企业成员时，若有剩余席位，系统会自动为其启用席位。批量添加企业成员时，若剩余席位不足，系统将按成员名称字母顺序启用席位。未启用席位的成员，可在“席位管理”页面重新分配席位，或变更套餐开通更多席位，拥有席位的成员方可使用华为云码道代码智能体服务。

图 2-1 添加已有 IAM 用户为企业成员

表 2-1 添加已有 IAM 用户为企业成员

参数	说明
角色	配置IAM用户在华为云码道代码智能体中的角色。 • 成员：普通成员权限，可执行查看个人用量、查看知识空间等操作。 • 企业管理员：拥有最高管理权限，可执行添加企业成员、创建团队、配置席位等操作。
选择IAM用户	从IAM用户列表中选择待添加为企业成员的用户。支持一次勾选多个IAM用户。

步骤5 单击“确定”，完成企业成员的添加。

在“企业全部成员”页面，查看新添加的成员。

----结束

2.2 安装

本文以首次下载安装为例，操作截图以Linux操作系统演示。

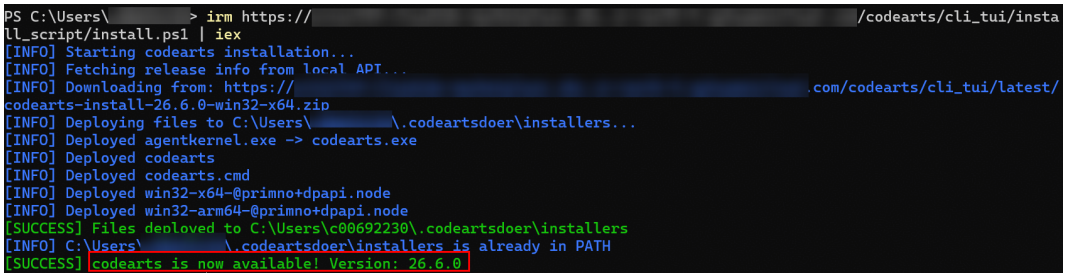
Windows

步骤1 **Windows**仅支持在**PowerShell**执行安装命令。请在PowerShell执行如下命令：

```
irm https://cnnorth4-cloudide-marketplace.obs.cn-north-4.myhuaweicloud.com/codearts/cli_tui/install_script/install.ps1 | iex
```

例如下图所示，如果正常回显码道CLI的版本号，代表成功安装码道CLI。

图 2-2 Windows 安装成功回显版本号



步骤2 安装成功后，请继续参考**授权**章节完成授权。

步骤3 （可选）如果您处于公司内网，请参考文档**配置代理**，完成代理的配置。

----结束

macOS/Linux

本文截图以Linux操作系统为例。

步骤1 （可选）如果您处于公司内网，请参考文档**配置代理**，完成代理的配置。

步骤2 请在终端执行如下命令，下载并执行码道CLI的安装脚本：

```
sh -c "$(curl -L https://cnnorth4-cloudide-marketplace.obs.cn-north-4.myhuaweicloud.com/codearts/cli_tui/install_script/install.sh)" && export PATH=~/.codeartsdoer/installers:$PATH
```

例如下图所示，如果正常回显码道CLI的版本号，代表成功安装码道CLI。

图 2-3 安装成功后回显版本号

步骤3 安装成功后，请继续参考**授权**章节完成授权。

----结束

2.3 授权

在码道CLI中，您可以通过**进程级配置**或**系统环境变量配置**来完成授权操作。其中，进程级配置适用于临时授权，而系统环境变量配置则用于设置永久授权。

前置准备

步骤1 参考**访问密钥**新建访问密钥。

步骤2 在新建访问密钥页面，勾选协议，单击“下一步”，填写密钥“描述”后，单击“确定”，完成授权。

创建成功后，请单击“立即下载”，保存密钥信息，否则弹窗关闭后将无法再次获取该密钥信息。

步骤3 根据您的操作系统，参考下述章节并完成环境变量的配置。其中，Access Key Id和Secret Access Key的值，通过您下载保存的credentials.csv文件中获取。

----结束

Windows

下面为您介绍在Windows系统中，如何通过进程级配置和系统环境变量配置完成授权操作。

配置进程级变量

通过配置进程级变量，可对当前窗口完成临时授权。

- 如果您使用CMD，请执行下述命令进行配置。
  ```
  set CODEARTS_CLI_AK=Access Key Id
  set CODEARTS_CLI_SK=Secret Access Key
  ```
  执行完成后，在当前窗口执行如下命令，检验是否生效。
  ```
  echo %CODEARTS_CLI_AK%
  echo %CODEARTS_CLI_SK%
  ```
  如果显示您刚设置的内容，则配置生效，如果显示“%CODEARTS_CLI_AK%”或者“%CODEARTS_CLI_SK%”，表示配置未生效。
- 如果您使用PowerShell，请执行下述命令进行配置。
  ```
  $env:CODEARTS_CLI_AK="Access Key Id"
  $env:CODEARTS_CLI_SK="Secret Access Key"
  ```
  执行完成后，您可以执行下述命令验证是否设置成功。无异常回显则表示配置生效。
  ```
  echo $env:CODEARTS_CLI_AK
  echo $env:CODEARTS_CLI_SK
  ```

配置系统环境变量

通过配置系统环境变量，可实现配置永久生效。

步骤1 打开“系统属性”，单击“高级”页签下的“环境变量”。

步骤2 修改“用户变量”或者“系统变量”，创建CODEARTS_CLI_AK和CODEARTS_CLI_SK。

其中，CODEARTS_CLI_AK的值为“Access Key Id”，CODEARTS_CLI_SK的值为“Secret Access Key”。

步骤3 配置完成后，您可进入想要启动的项目，鼠标右键“在终端中打开”，输入“codearts”并回车。

进入TUI开发环境，表示永久性配置生效。

----结束

macOS

下面为您介绍在macOS系统中，如何通过进程级配置和系统环境变量配置完成授权操作。

配置进程级变量

步骤1 打开终端并执行下述命令，可对当前窗口完成临时授权。

```
export CODEARTS_CLI_AK="Access Key Id"
export CODEARTS_CLI_SK="Secret Access Key"
```

步骤2 执行完成后，在当前窗口执行如下命令，检验是否生效。

```
echo $CODEARTS_CLI_AK
echo $CODEARTS_CLI_SK
```

如果输出密钥内容，表示配置生效，如果没有输出或者空行，表示配置没有生效。

----结束

配置系统环境变量

步骤1 执行如下命令，把环境变量配置追加写入到“~/.zshrc”文件。

```
echo 'export CODEARTS_CLI_AK="Access Key Id"' >> ~/.zshrc
echo 'export CODEARTS_CLI_SK="Secret Access Key"' >> ~/.zshrc
```

步骤2 执行如下命令，将两个变量永久保存。

```
source ~/.zshrc
```

步骤3 执行如下命令，验证是否生效。

```
echo $CODEARTS_CLI_AK
echo $CODEARTS_CLI_SK
```

如果输出密钥内容，表示配置生效，如果没有输出或者空行，表示配置没有生效。

----结束

Linux

您可以根据您的需要，选择授权方式：

表 2-2 不同授权方式对比

授权方式	生效范围	生命周期	适用Shell	权限要求	典型使用场景
临时进程授权	仅当前终端窗口	关闭终端立即失效，重启服务器清空	bash、zsh、sh	普通用户即可	临时调试、一次性测试、临时执行脚本，不想留存密钥
Bash用户永久授权	当前登录用户，所有新终端	永久保存，重启终端或者服务器仍存在	Linux默认Bash（CentOS、Ubuntu或者麒麟等）	普通用户即可	个人服务器、日常高频使用，本机仅自己操作

临时进程授权

仅当前终端窗口生效，关闭窗口后配置自动失效，适合一次性调试、临时执行命令，不持久存储密钥。

步骤1 配置临时环境变量（AK/SK）。

在终端中执行以下命令，为当前会话窗口注入临时访问凭证：

```
export CODEARTS_CLI_AK="Access Key Id"
export CODEARTS_CLI_SK="Secret Access Key"
```

其中，*Access Key Id*和*Secret Access Key*请替换为您实际获取的密钥值，获取方式请参考**前置准备**。

步骤2 验证授权是否成功。

例如，在终端中执行**codearts models**命令，测试权限是否生效。

- 如果输出类似如下回显信息，表示授权成功。

```
root@szvphis32149187:~# codearts models
model_id                                  model_name
------------------------------------------------------------
huaweicloud-maas/deepseek-v3.2            DeepSeek-V3.2
huaweicloud-maas/GLM-4.7-SFT-Harmony          GLM-4.7-ArkTS-SPARK
huaweicloud-maas/Glm-5-internal           GLM-5
huaweicloud-maas/GLM-5.1                  GLM-5.1
huaweicloud-maas/GLM-5.2                  GLM-5.2
```

- 如果返回认证失败或错误提示，请执行如下命令，检查配置的AK/SK是否正确。

```
echo $CODEARTS_CLI_AK
echo $CODEARTS_CLI_SK
```

----结束

Bash 用户永久授权

如果当前登录用户需要长期使用，推荐此方式进行永久授权。新开终端、重启服务器配置均保留，无需管理员权限。

步骤1 将环境变量追加写入Bash配置文件。

执行以下命令，将CODEARTS_CLI_AK和CODEARTS_CLI_SK环境变量追加到当前用户的“.bashrc”文件中：

```
echo 'export CODEARTS_CLI_AK="Access Key Id"' >> ~/.bashrc
echo 'export CODEARTS_CLI_SK="Secret Access Key"' >> ~/.bashrc
```

其中，*Access Key Id*和*Secret Access Key*请替换为您实际获取的密钥值，获取方式请参考**前置准备**。

步骤2 重新加载Bash配置，使添加的环境变量立即生效。

```
source ~/.bashrc
```

步骤3 验证授权是否永久生效。

关闭当前终端窗口，打开一个新的终端会话，执行**codearts models**命令，测试授权是否生效。

- 如果回显模型信息，说明永久授权已生效。

```
root@szvphis32149187:~# codearts models
model_id                                  model_name
------------------------------------------------------------
huaweicloud-maas/deepseek-v3.2            DeepSeek-V3.2
huaweicloud-maas/GLM-4.7-SFT-Harmony          GLM-4.7-ArkTS-SPARK
huaweicloud-maas/Glm-5-internal           GLM-5
```

```
huaweicloud-maas/GLM-5.1                GLM-5.1
huaweicloud-maas/GLM-5.2                GLM-5.2
```

- 如果返回认证失败或错误提示，请执行如下命令，检查配置的AK/SK是否正确。

```
echo $CODEARTS_CLI_AK
echo $CODEARTS_CLI_SK
```

----结束

2.4 升级

当前码道CLI支持两种升级方式：自动升级和手动升级。

自动升级

当您启动码道CLI，当前版本不是最新版本，码道CLI会提示您是否自动升级。

本文截图均以CMD界面为示例。

步骤1 系统检测到新版本时，会提示您是否升级。

步骤2 选择“Yes”，启动升级操作。

提示“Installation Successful”，表示成功升级码道CLI。

----结束

手动升级

如下图所示，您执行“codearts upgrade”命令，回显中出现Done，表示您成功完成手动升级。

图 2-4 手动升级

3 权限

码道CLI可通过配置“permission”文件，管控码道CLI的执行权限，支持自动运行、提示用户审批以及禁止运行。

配置路径

权限文件的配置路径：~/.codeartsdoer/cli-data/storage/permission/global.json

配置格式

权限配置格式如下所示。

```
[
  {
    "permission": "权限名称",
    "pattern": "*",
    "action": "权限策略"
  }
]
```

其中，“权限名称”请参考表3-1，“权限策略”请参考表3-2。

权限策略配置参数

表 3-1 权限名称

权限	说明
edit	所有文件修改，包括AI新建、修改文件或者批量修改代码时触发
write	写入文件
deleteFile	删除文件
bash	运行Shell命令，匹配解析后的完整命令。例如AI执行“git status”、“npm install”、“python manage.py runserver”等命令时触发 bash命令的权限需要与“bash_mode”结合，表示沙箱模式下的白名单命令。“bash_mode”也需要在“~\.codeartsdoer\cli-data\storage\permission\config.json”文件中配置，“bash_mode”支持的配置模式请参考表3-3。

权限	说明
webfetch	获取URL，AI访问外部API、下载文件或者获取网页内容时触发
external_directory_read	读取项目目录外的路径，AI读取“/etc/hosts”、“~/.ssh/id_rsa”这类不在项目工作目录内的文件时触发
external_directory_write	修改项目目录外的路径，AI写入或者修改“/etc”、“~/.bashrc”这类系统或用户配置文件时触发
dotfile	修改点文件（以“.”开头的配置文件），AI修改“.gitignore”、“.env”、“.vscode”或者“settings.json”等隐藏配置文件时触发
doom_loop	工具调用重复循环保护，当同一工具，例如“read”以完全相同的输入被连续调用3次时触发，防止死循环

表 3-2 权限策略

配置项	说明
allow	直接运行
ask	需要审批，当您配置ask时，将出现以下三个选项供选择： ● Allow once，仅允许本次执行，下次遇到相同命令会再次询问。 ● Allow always，本次会话中，此类命令将不再询问，直接执行。 ● Reject，不执行此命令，继续对话。适合您认为有风险或不必要的操作。
deny	禁止执行

说明

- 在TUI开发模式下，默认在沙箱模式下运行，将返回“Allow once”、“Allow always”和“Reject”三个选项。
- 在CLI开发模式下，自动禁止执行，如果需要配置为运行模式，请配置为“allow”。

表 3-3 “bash_mode”支持的模式

模式	含义
sandbox	在沙箱中执行（默认，最安全）
always_ask	所有命令都询问
always_allow	所有命令直接执行

白名单配置示例

示例如下：

```
[
  {
    "permission": "bash",
    "pattern": "git *",
    "action": "allow"
  },
  {
    "permission": "bash",
    "pattern": "npm *",
    "action": "allow"
  },
  {
    "permission": "bash",
    "pattern": "ls *",
    "action": "allow"
  }
]
```

该示例包含3个JSON对象，支持AI自动执行git、npm、ls开头的所有命令。

4 自定义命令

除了内置的斜杠命令、文件引用与Bash命令，还支持通过**在.jsonc文件自定义命令**或者**自定义.md文件**来添加自定义命令。

在.jsonc 文件自定义命令

码道CLI支持通过在“codearts_cli.json”或者“codearts_cli.jsonc”（带注释的JSON文件）配置自定义命令。

支持配置个人级和项目级路径，详细路径信息可查看**表4-1**。

⚠ 注意

使用前需您自行手动创建文件并编辑配置内容，完成后再运行码道CLI。

表 4-1 配置路径

配置级别	配置说明	位置
个人级	个人级配置为全局默认配置，您可以用于设置个人偏好（例如主题或快捷键），对所有项目生效。	“~/.codeartsdoer/codearts_cli.json”或“~/.codeartsdoer/codearts_cli.jsonc” “~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/
项目级	项目级配置为项目专属配置，**优先级高于个人级**，可覆盖个人级配置，用于定义项目统一规则。	“项目根目录/.codeartsdoer/codearts_cli.json”或“项目根目录/.codeartsdoer/codearts_cli.jsonc”

配置示例

以配置自定义命令“test”、“check”为例。

步骤1 本示例以项目级配置和Windows系统为例。

在项目根目录的“./.codeartsdoer”下，创建“codearts_cli.jsonc”文件，并配置为如下内容：

其中，自定义的参数命令名字和“template”字段为必填。

```
{
  //Command配置
  "command": {
    //所有自定义命令都放在这里
    "test": {
      //命令名字=test，您可以根据习惯自定义为run、fix等
      "template": "当您设置后这里是发送给AI的提示词。可以是一段话，也可以是复杂的指令。",
      "description": "这个命令的简短说明，会在TUI的命令列表中显示。",
      "agent": "执行此命令的AI代理名称 (可选)",
      "model": "为此命令指定的模型ID (可选，会覆盖默认模型)",
      "subtask": false // 可选的布尔属性，用于控制此命令是否以子智能体模式执行。
    },
    //第二个自定义命令示例check
    "check": {
      "template":"这里填写run命令专属AI提示词，用于运行项目、检查代码。",
      "description": "一键检查代码并运行当前项目",
      "agent": "",
      "model": "",
      "subtask": false
    }
  }
}
```

步骤2 检查上述自定义命令的配置是否生效。

- 在TUI开发模式下，输入“/test”验证自定义命令test。您也可以输入“/check”查看自定义命令check。

  图 4-1 TUI 开发模式示例图

- 在CMD、PowerShell或者终端，执行“codearts run check”，即可关联出自定义的命令“check”。您也可以执行“codearts run test”，即可关联出自定义命令“test”。

  说明

  回显可能存在差异，最终效果请以实际输出为准。

----结束

自定义.md 文件

优先级：同名情况下，自定义.md文件 > 在.json文件内自定义命令

用户可在commands目录下自定义Markdown文件，以此定义自定义命令，**文件名称即命令名称**。

表 4-2 commands 文件夹路径

级别	路径
个人级	~/.codeartsdoer/commands “~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/
项目级	项目根目录/.codeartsdoer/commands

配置示例

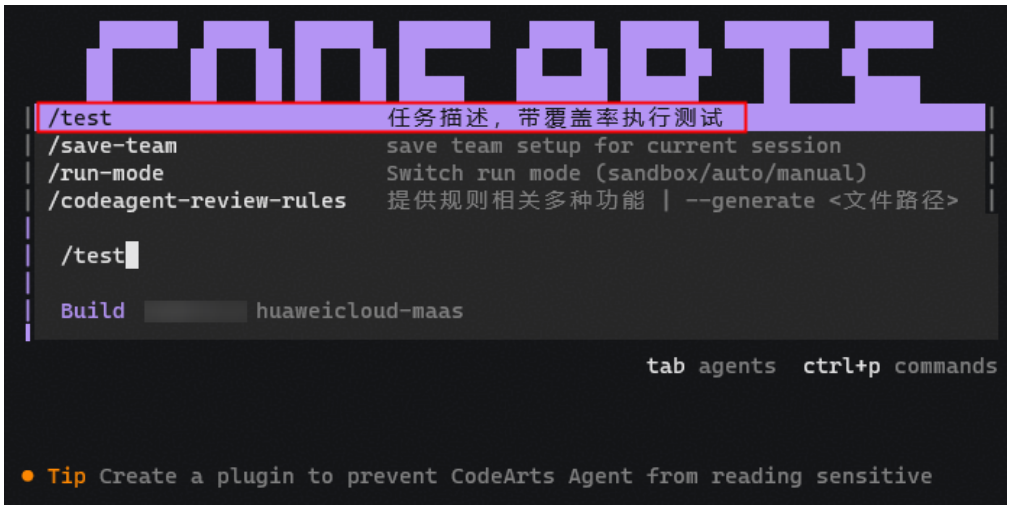
步骤1 本示例以Windows系统为例，创建个人级自定义命令。在目录“~\.codeartsdoer\commands”下创建命令文件“test.md”。

```
---
description: 任务描述，带覆盖率执行测试
agent: build
model: huaweicloud-maas/GLM-5.1
---
执行全量测试并生成覆盖率报告，排查报错用例并给出修复建议。
```

步骤2 重启码道CLI。

- 在TUI开发模式下，输入“/test”即可关联出自定义的命令“test”。

  图 4-2 TUI 开发模式示例图

- 在CMD、PowerShell或者终端，执行“codearts run test”，即可关联出自定义的命令“test”。

说明

回显可能存在差异，最终效果请以实际输出为准。

----结束

5 沙箱

沙箱（Sandbox）为智能体生成的指令提供**隔离且受限的执行空间**。通过严格的权限管控，确保命令在安全隔离的环境内运行，有效阻断对未经授权资源（文件、网络等）的访问或者高风险命令（权限修改等）。对于被拦截的命令，智能体会在对话流中发出风险提示，经由用户二次确认后，命令才会在沙箱外被执行。

约束与限制

表 5-1 约束与操作限制

限制类别	具体说明
操作系统	• Windows操作系统：**推荐使用Windows 11 (x64)**；如果使用Windows 10 (x64)，系统版本需为2019年及以上，建议升级至最新稳定版本。 • macOS操作系统：macOS 11及以上版本，兼容ARM64（Apple Silicon）架构。 • Linux：Huawei Cloud EulerOS 2.0、SUSE Linux Enterprise Server 12 SP5、Ubuntu 18.04/20.04/22.04/24.04 LTS、Debian 10/11/12、CentOS 8、RHEL 8/9。
管控对象	• Windows：只管控Bash命令和PowerShell命令。 • macOS：只管控Shell命令。 • Linux：只管控Shell命令。

文件访问控制

启用沙箱后，码道CLI对文件目录的访问权限配置如下。您也可以根据实际业务需求，自定义码道CLI对文件目录的访问权限，详细介绍请参见**启用沙箱**。

表 5-2 文件访问控制

操作系统	权限类型	目录类型	目录列表
Windows	只读	-	除**Windows系统关键目录**及**用户敏感目录**外，其余所有目录均开放读权限。
	读写	项目目录及其子目录	-
	不可读不可写	Windows系统关键目录	• C:\Windows\System32\config • C:\Windows\System32\drivers\etc • C:\Windows\SysWOW64\config • C:\ProgramData\Microsoft\Crypto • C:\Windows\System32\GroupPolicy • C:\Windows\System32\GroupPolicyUsers • C:\ProgramData\Microsoft\Windows\WER
		用户敏感目录	• C:\Users*\AppData\Local\Microsoft\Credentials • C:\Users*\AppData\Roaming\Microsoft\Credentials • C:\Users*\AppData\Local\Microsoft\Protect • C:\Users*\NTUSER.DAT • C:\Users*\ntuser.dat.LOG
macOS	只读	-	除**系统敏感目录**外，其余所有目录均开放读权限。
	读写	工作空间和额外设置的读写目录	• 临时目录：/tmp、/var/folders、TMPDIR环境变量路径 • 缓存目录：~/Library/Caches、~/.cache • 通用工具依赖目录：~/.local/lib、~/.local/bin、~/.local/share • 常用开发语言（Go、Java、Python、Node.js、Rust、Ruby）的工具链及其依赖目录

操作系统	权限类型	目录类型	目录列表
	不可读不可写	文档/桌面/下载（隐私）	• ~/Desktop • ~/Documents • ~/Downloads • ~/Pictures • ~/Movies • ~/.ssh • ~/.zsh_history
		密码/钱包/钥匙链相关	~/Library/Keychains
		系统级敏感配置	• /etc/passwd • /private/etc/passwd • /etc/group • /private/etc/group • /etc/hosts • /private/etc/hosts • /etc/resolv.conf • /private/etc/resolv.conf • /etc/pam.d • /private/etc/pam.d
Linux	只读	-	除**系统敏感目录**外，其余所有目录均开放读权限。
	读写	工作空间和额外设置的读写目录	• 临时目录：/tmp、TMPDIR环境变量路径 • 缓存目录：~/.cache、XDG_CACHE_HOME环境变量路径 • 通用工具依赖目录：~/.local/lib、~/.local/bin、~/.local/share • 常用开发语言（Go、Java、Python、Node.js、Rust、Ruby）的工具链及其依赖目录
	不可读不可写	Linux系统关键目录	• /etc/shadow • /etc/passwd • /etc/group • /etc/gshadow • /etc/resolv.conf • /etc/sudoers

启用沙箱

下面以在TUI开发环境中启用沙箱为例。在CLI开发环境中，请直接执行“codearts run "输入的指令" --sandbox”命令即可。

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 启用沙箱运行模式。

1. 在TUI对话框中，输入“/run-mode”并回车，进入“选择运行模式”页面。
2. 单击“Sandbox沙箱”，进入“沙箱配置”页面。

图 5-1 选择 SandBox 沙箱

步骤3 配置命令白名单。

根据实际需求将特定命令的前缀添加到白名单。被加入白名单的命令将绕过沙箱机制，直接在沙箱外执行。

步骤4 选择网络访问策略。

表 5-3 网络访问策略说明

参数	说明
全网访问	允许访问所有内部及外部网络资源。
本地网络访问	仅允许访问本地服务网络（局域网/内网），禁止访问外部互联网。
禁止网络访问	阻断所有网络连接，禁止访问任何内部或外部资源。
网络自定义访问	通过修改策略配置Json文件，自定义当前项目沙箱环境中进程的文件与网络访问范围。具体操作，请参见步骤5。

步骤5 自定义网络访问策略。

1. 在“网络访问策略”中，选择“网络自定义访问”，单击“确认”，进入“自定义策略”页面。
2. 在策略配置Json文件中，根据您的需求修改配置。

该文件的初始结构如下所示：

```
{
    "filesystem": {
        "readWrite": [],
        "readOnly": []
    },
    "network": {
        "default": "Allow",
        "allow": [],
        "deny": []
    },
    "resources": {
        "cpu": 50,
        "memory": 8
    }
}
```

表 5-4 策略配置 Json 文件参数说明

参数	是否必选	参数类型	描述
filesystem	否	**filesystem** Object	用于精确控制沙箱对本地文件系统的访问权限。 如果未设置（filesystem字段为空或不存在），则采用沙箱内置的文件系统安全策略。
network	否	**network** Object	用于控制沙箱内进程的网络访问策略，支持配置允许或阻止对特定网络资源的访问。 如果未设置，默认允许网络访问
resources	否	**resources** Object	用于定义沙箱运行时的计算资源上限，确保业务稳定并防止资源滥用。 如果未设置，系统将共享宿主机资源。

表 5-5 filesystem 字段参数说明

参数	参数类型	默认值	支持路径格式	优先级规则	描述
readWrite	Array	[ ]	- 绝对路径：如/home/user/project、C:\Projects - 相对路径：如./src、./config - 环境变量：$HOME（Linux/Mac）、%USERPROFILE%（Windows） - Home目录简写：~	readOnly > readWrite > 系统默认策略 如果某个路径同时匹配readOnly和readWrite，则以readOnly为准。	可读写路径列表。
readOnly	Array	[ ]			只读路径列表。

表 5-6 network 字段参数说明

参数	参数类型	默认值	优先级规则	描述
default	String	Allow	deny > allow > default 如果allow和deny同时配置同一规则时，则以deny为准。	默认的网络访问策略。 - Allow：允许访问 - Deny：禁止访问 **说明** 该字段支持“域名:端口”和“IP:端口”两种配置格式，其中域名部分支持通配符，IP地址支持CIDR表示法。多个端口可通过逗号分隔进行配置。若未指定具体端口，则默认对所有端口生效。
allow	Array	[ ]		允许访问的网络规则列表。
deny	Array	[ ]		拒绝访问的网络规则列表。

表 5-7 resources 字段参数说明

参数	参数类型	默认值	最小值	描述
cpu	Integer	50	20	CPU占比，单位为%。
memory	Integer	8	1	内存大小，单位为GB。

策略配置Json文件配置示例如下：

```
{
  "filesystem": {
    "readWrite": [
      "/home/user/project/output",
      "~/workspace/temp"
    ],
    "readOnly": [
      "/etc/systemd",
      "%USERPROFILE%/.ssh"
    ]
  },
  "network": {
    "default": "Allow",
    "deny": [
      "10.0.0.0/8",
      "192.168.0.0/16"
    ]
  },
  "resources": {
    "cpu": 50,
    "memory": 8
  }
}
```

3. 单击“确认”，退出当前设置界面，完成沙箱模式的启用。

----结束

高风险命令执行策略

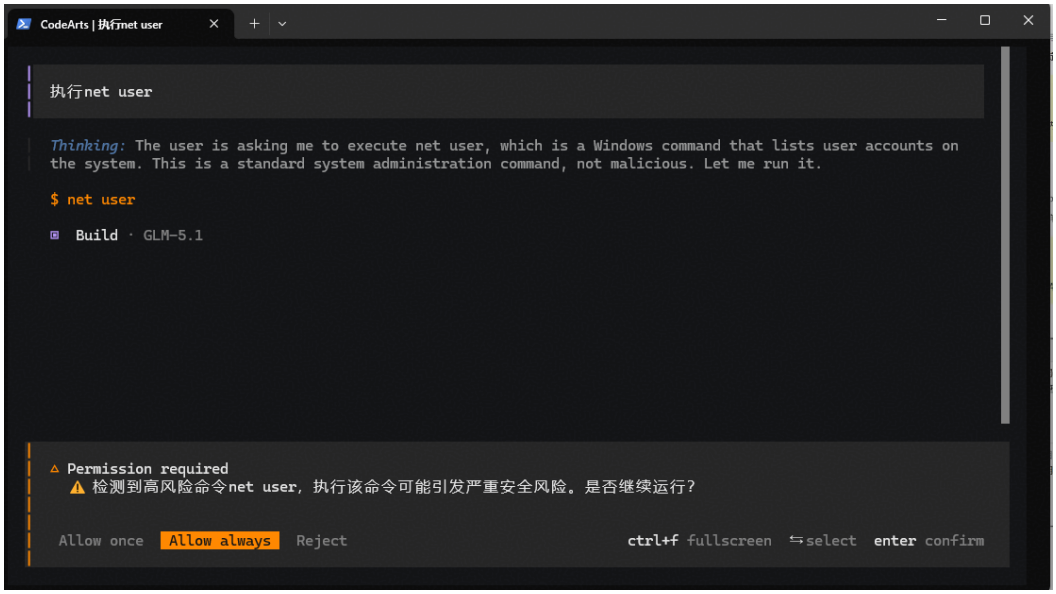
当智能体检测到高风险命令时，AI将会在对话流中发出提示（**TUI开发环境会弹框是否被拦截，CLI开发环境直接被拦截**）。请评估风险后，按需选择执行方式。

- Allow once，仅允许本次执行，下次遇到相同命令会再次询问。
- Allow always，本次会话中，此类命令将不再询问，直接执行。
- Reject，不执行此命令，继续对话。适合您认为有风险或不必要的操作。

说明

如果需要关闭沙箱功能，请参考**关闭沙箱模式**。

图 5-2 高危命令被拦截示例



关闭沙箱模式

以在TUI开发环境中关闭沙箱为例。在CLI开发环境中，请直接执行“codearts run "输入的指令" --auto”命令即可。

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

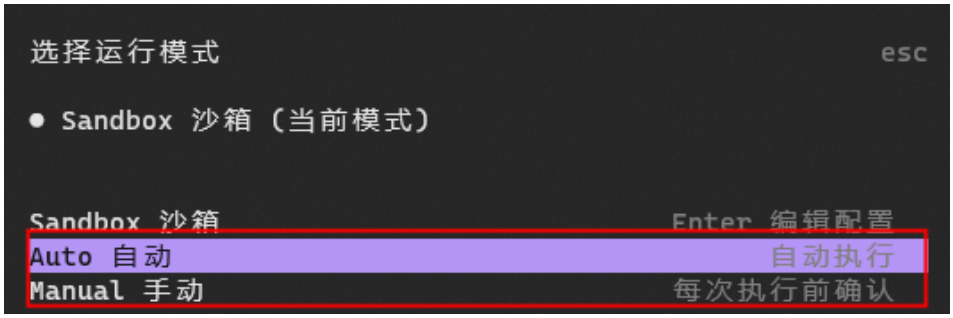
步骤2 关闭沙箱运行模式。

关闭沙箱模式就是将运行模式设置为手动或者自动。

1. 在TUI对话框中，输入“/run-mode”并回车，进入“选择运行模式”页面。
2. 在“Auto 自动”或“Manual 手动”上，单击Enter键，即可退出沙箱运行模式。
   - **Auto 自动**：智能体直接执行所有命令，无需征得您的同意。

     为了确保安全，建议只在必要时启用“自动运行”模式。此模式会绕过所有安全检查，可能在未经提醒的情况下执行高风险操作。
   - **Manual 手动**：智能体执行任何命令前，都会向用户发送确认提示，用户需手动确认后，命令才会继续执行。

图 5-3 切换沙箱运行模式



----结束

6 智能体

智能体（Agent）是面向多样化开发场景的编程助手。码道CLI内置了多款开箱即用的智能体，这些智能体无需手动创建，即拿即用，旨在为多种开发场景提供高效的编程辅助。除了内置智能体，还支持自定义智能体功能。

内置智能体

内置智能体分为**主智能体**和**子智能体**。主智能体是您直接交互的主要助手，可以使用**Tab键**来循环切换，这些主智能体处理您的主要对话。子代理是主代理可以调用来执行特定任务的专业助手，可以通过在消息中**@**子智能体来手动调用。

表 6-1 内置智能体列表

分类	创建者	说明
主智能体	系统内置	码道CLI内置了两个核心主智能体Build和Plan，分别承担开发与规划职责。 ● Build主智能体 默认的开发执行智能体，拥有完整工具访问权限，适用于需要实际修改代码、执行命令或创建文件等真实开发操作场景。 典型使用场景： – 编写、修改或删除代码文件 – 执行构建、测试、部署等终端命令 – 创建新功能模块或修复Bug – 对项目进行任何需落地变更的实际操作 ● Plan主智能体 受约束的只读型智能体，专注于分析代码结构、设计解决方案与制定实施计划，**不会对代码库做任何实质性修改**。 典型使用场景： – 分析代码结构和逻辑 – 设计功能方案或重构策略 – Bug定位和修复方案规划 – 架构评审和可行性分析 – 任何需要“先想清楚再做”的场景
子智能体	系统内置	子智能体是主智能体可以调用来执行特定任务的专业助手，您可以通过在消息中@子智能体来手动调用。 **说明** 您可以通过如下命令，查看系统内置的子智能体： ● 在CLI环境中：codearts run "自然语言描述"，自然语言描述中必须要带**subagent**关键词。 ● 在TUI环境中：在输入框中，直接输入@即可查看。 ● developer-test-agent：专门用于单元测试生成、修复、覆盖率优化，审查真实代码库。 ● explore：代码库探索智能体，专门用于快速查找文件、搜索代码关键词、回答代码库相关问题。 ● general：通用智能体，用于研究复杂问题和执行多步骤任务。 ● rule-generator：规则生成器。 ● spec-design-agent：生成综合技术设计，将需求（做什么）转化为架构（如何做）。 ● spec-requirement-agent：基于项目描述和上下文生成EARS格式需求。 ● spec-task-agent：根据需求和设计生成实现任务。

自定义智能体

除了系统内置的智能体外，码道CLI还支持自定义智能体功能。**存在同名的项目级和个人级智能体时，项目级智能体优先级高于个人级智能体**。

表 6-2 自定义智能体分类

分类	作用域	说明
自定义主智能体 自定义子智能体	项目级	仅针对当前项目的智能体，随代码库分发，存于本地。 存储位置：项目根目录/.codeartsdoer/agents
	个人级	个人习惯或特定偏好的智能体，仅对本用户下的所有项目生效。 存储位置：~/.codeartsdoer/agents “~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/

通过 Markdown 文件创建智能体

码道CLI支持通过新建Markdown文件，配置本地项目级或本地个人级智能体。存在同名的项目级和个人级智能体时，项目级智能体优先级高于个人级智能体。

自定义智能体文件格式

自定义智能体采用Markdown文件格式编写，文件头部配置YAML Frontmatter元信息，主体承载提示词角色指令。

```
---
description: Agent desc  # 智能体功能描述
mode: subagent           # 智能体描述信息
model: provider/model     #如果参照此格式配置模型，将覆盖默认模型
tools:                   # 工具权限配置
  tool1: true            # 允许使用tool1
  tool2: false           # 禁止使用tool2
---

# 提示词
这里是智能体的系统提示词内容
```

上述自定义智能体各参数填写规范如下表所示：

表 6-3 Markdown 文件参数说明

参数	是否必选	参数类型	默认值	描述
description	是	字符串	不涉及	自定义智能体的描述。

参数	是否必选	参数类型	默认值	描述
mode	是	字符串	subagent	设置智能体的角色。 • subagent：只能当作子智能体被调用。 • primary：只能当主智能体，不能被调用。 • all：既能当主智能体，也能当子智能体被调用。
model	否	字符串	不涉及	格式为：provider/model
tools	否	对象	不涉及	控制智能体可使用的工具权限，码道CLI支持的工具如表6-5，您也可以使用“*”匹配所有工具。 示例： tools: "*":false #表示禁止所有工具
hidden	否	布尔值	不涉及	可见性控制。 设为“true”时，从@自动补全菜单中隐藏。适用于仅供编程调用的内部代理。
disable	否	布尔值	不涉及	开关控制。 • true：禁用该智能体 • false：启用该智能体
提示词	是	字符串	不涉及	自定义智能体的系统提示词。

表 6-4 tools 对象取值说明

参数	说明
true	允许使用该工具
false	禁止使用该工具

表 6-5 码道 CLI 支持的工具

工具	说明
read	读取文件
write	写入文件
edit	编辑文件
bash	执行命令

工具	说明
glob	文件模式匹配
grep	内容搜索
task	子任务
webfetch	获取网页
question	询问用户

自定义智能体示例

本示例以Windows系统下配置个人级子智能体为例。

步骤1 进入“~/.codeartsdoer/agents”目录下，创建“test.md”文件，写入如下内容。

“~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/

```
---
# ========== 必填项 ==========
description: 智能体的功能及使用场景描述（必填，主智能体据此判断调用时机）
mode: subagent            # 枚举值：subagent子智能体 / primary主智能体 / all两者通用

# ========== 模型配置 ==========
model: provider/model-id  # 自定义模型，格式规范：provider/model-id，执行“/models”可查看支持的模型

# ========== 行为控制 ==========
hidden: false             # true：隐藏@联想菜单，仅程序内部调用
disable: false            # true：停用当前代理

# ========== 工具权限 ==========
tools:
  write: false            # 文件写入权限
  edit: false             # 文件编辑权限
  bash: false             # 系统命令执行权限
  read: false             # 文件读取权限

# 系统提示词（支持多行Markdown格式）
你是一名资深代码审查专家。请重点关注：

-代码质量与最佳实践
-潜在Bug和边界情况
-性能影响
-安全隐患

提供建设性反馈，不要直接修改代码。
```

步骤2 配置后，进入任意代码项目下，执行如下命令。

```
@test 分析代码
```

下发命令后，AI会指定自定义子智能体进行代码分析。

图 6-1 调用 test 子智能体分析代码

----结束

7 技能

配置目录

技能按作用范围可分为项目级和个人级。

类型	作用域	存放位置
项目级	仅针对当前项目生效，存储在本地	项目根目录/.codeartsdoer/skills
个人级	当前用户下的所有项目	~/.codeartsdoer/skills “~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/

若存在名称相同的技能，技能调用优先级为：项目级 > 个人级。

技能的目录结构

技能是一个包含**必需的SKILL.md文件**和其他**可选的捆绑资源**的结构化目录，打包完成特定任务所需的知识和工具。

```
skill-name/
├── SKILL.md (必填)
│   ├── YAML frontmatter 元数据 (必填)
│   │   ├── name: (必填)
│   │   └── description: (必填)
│   └── Markdown 指令 (必填)
└── Bundled Resources/ 捆绑资源 (选填)
    ├── scripts/      - 可执行代码 (Python/Bash 等)
    ├── references/   - 旨在根据需要加载到上下文中的文档
    └── assets/       - 输出中使用的文件 (模板、图标、字体等)
```

表 7-1 技能参数说明

参数名称	说明
skill-name	整个技能的根目录，skill-name需要替换成实际的技能名称（比如data-analysis-skill）。技能名称由小写字母、数字和连字符（-）组成，开头和结尾不能是连字符，且连字符不可连续使用，长度1~64字符。
SKILL.md	**SKILL.md文件名不可修改。**它是技能的核心描述文件，相当于技能的“说明书”，必须包含以下两部分： • **YAML frontmatter元数据**：放在SKILL.md文件的最顶部，以“---”首尾包裹，是AI可解析的结构化信息。包含name和description两个字段。 – **name**：技能名称，由小写字母、数字和连字符（-）组成，开头和结尾不能是连字符，且连字符不可连续使用，长度1~64字符。**name必须与skill-name保持一致**。 – **description**：技能描述，长度不超过1024字符。该字段对华为云码道代码智能体识别技能的使用场景至关重要，需清晰描述技能功能及适用触发条件。 • **Markdown指令**：YAML元数据下方的Markdown内容，用于详细说明技能的使用方法、参数、示例和依赖等。
Bundled Resources（可选）	存放技能配套资源的目录，非必填，但能让技能更完整。 • **scripts/**：存放技能的可执行脚本，例如Python脚本、Shell脚本等，实现**确定性操作**。 • **references/**：存放技能依赖的参考文档，例如API文档、领域知识、公司政策等，是给智能体的**知识库**，这些文档会在技能执行时按需加载到上下文。 • **assets/**：存放资源文件，例如字体资源、公司Logo、PPT模板等，直接在智能体生成的输出中使用，不需要被加载到上下文窗口。

示例

本文以Windows系统为例，创建个人级技能，在“.codeartsdoer/skills”下创建“test”文件夹，在该文件夹下创建“SKILL.md”文件，写入如下内容。

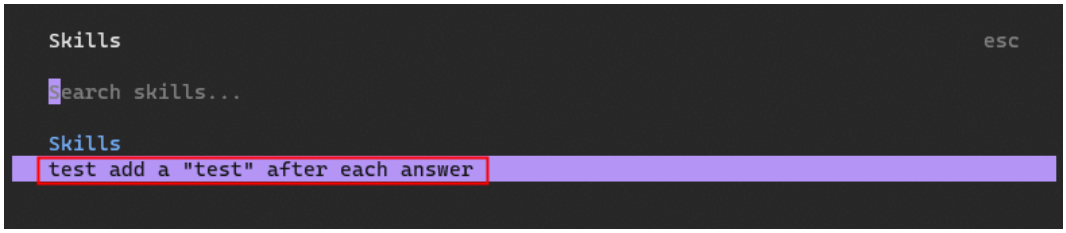
```
---
name: test
description: add a "test" after each answer
---
#回答增加后缀
在每个回答结束的时候加上“测试”
```

编写示例后，可在**TUI下启动**或者**CLI下启动**进行验证。

TUI 下启动

步骤1 编写示例后，进入TUI开发模式，输入“/skills”可查看技能列表。

图 7-1 查看技能



步骤2 选中“test”技能后，输入指令“测试一下”并回车，“test”技能生效。

----结束

CLI 下启动

步骤1 打开CMD，输入如下指令：

```
codearts run add a "test" after each answer
```

步骤2 回显如下信息，表明test技能已生效。

```
C:\Users\用户名>codearts run add a "test" after each answer

> build · GLM-5.1

→ Skill "test"

已加载该技能。现在起，我会在每个回答末尾加上“测试”。
```

----结束

8 Hooks

Hooks（钩子）是码道CLI插件系统的核心机制，允许开发者在工具执行、消息处理、权限控制等关键节点注入自定义逻辑。插件通过返回一个Hooks对象来声明自己关注的生命周期事件，当对应事件触发时，插件的自定义函数将被调用。

插件及依赖文件存放路径

插件是一个JavaScript/TypeScript模块，通过导出插件函数来工作。每个函数接收一个上下文对象，并返回对应的Hook对象。如果插件文件中需要使用外部包，必须在配置目录中创建package.json文件并配置所需的依赖项。华为云码道代码智能体在启动时会自动安装这些依赖项。

表 8-1 插件文件加载路径

插件类型	插件文件存放路径	插件依赖文件存放路径	说明
项目级	当前项目根目录./.codeartsdoer/plugin或./.codeartsdoer/plugins	当前项目根目录./.codeartsdoer/	仅对当前项目有效。
个人级	本地%USERPROFILE%/.codeartsdoer/plugin或 %USERPROFILE%/.codeartsdoer/plugins	本地%USERPROFILE%/.codeartsdoer/	对当前用户下所有项目均有效。

支持的 Hook 事件

码道CLI支持的Hook事件请参考**表8-2**，各Hook事件示例详见文档**Hook事件参考**。

表 8-2 支持的 Hook 事件

事件分类	事件名称	触发时机	可阻断	典型使用场景	说明
聊天消息	chat.message	用户发送消息后、进入处理流程前	是	聊天消息处理	新消息到达时自动触发，支持对消息内容及Parts进行编辑
聊天参数	chat.params	每次调用LLM前	是	聊天参数修改	支持动态调整大模型推理参数，包含temperature、topP、topK等，按需优化模型输出结果
聊天请求头	chat.headers	每次调用LLM前	是	聊天请求头修改	支持自定义修改HTTP请求头字段，适配网络代理、接口鉴权等复杂场景需求
聊天响应	chat.response	LLM返回响应后	否	聊天响应记录	自动捕获模型响应的Tokens、Cost、Duration等核心指标，实现精准计费统计、高效故障排查与完整日志留存
聊天错误	chat.error	LLM调用发生错误时	否	聊天错误记录	记录LLM调用失败时的错误日志
聊天压缩	chat.compression	上下文压缩时	否	聊天压缩记录	记录上下文压缩前后的Token数量，辅助存储与性能调优
聊天结束	chat.finished	聊天会话结束时	否	聊天结束记录	记录会话结束事件，包含本次运行的Token消耗和运行信息

事件分类	事件名称	触发时机	可阻断	典型使用场景	说明
用户审批	user.approval	需要用户确认/审批操作时	是	用户审批记录	拦截高风险操作，发起人工确认审批，管控敏感操作执行权限
Turn结束	turn.end	每个对话轮次结束时	否	Turn结束记录	记录每轮对话结束日志，包含处理耗时、终止原因等关键信息
命令执行	command.execute.before	命令执行前	是	命令执行前置处理	支持对Parts进行自定义配置，实现命令的前置处理与灵活适配
工具执行	tool.execute.before	工具执行前	是	工具执行前置参数调整	支持在工具执行前自定义修改参数args，实现校验、动态改写与业务适配
	tool.execute.after	工具执行后	是	工具执行后置结果处理	支持自定义修改返回结果，包括title、output及metadata字段
工具定义	tool.definition	工具定义发送给LLM前	是	工具定义修改	支持在工具推送模型前自定义配置描述与参数，确保精准适配业务场景
Shell环境	shell.env	获取Shell环境变量时	是	Shell环境变量配置管理	获取Shell环境变量时，支持添加或修改环境变量
权限请求	permission.ask	需要权限验证时	是	权限请求处理	处理权限校验请求，支持配置allow（允许）、deny（拒绝）、ask（询问）三种策略

事件分类	事件名称	触发时机	可阻断	典型使用场景	说明
消息转换	experimental.chat.messages.transform	消息列表发送给LLM前	是	消息转换	支持在模型上报前预处理聊天消息列表
系统消息转换	experimental.chat.system.transform	系统提示词发送前	是	系统消息转换	支持在系统提示词发送前优化指令内容
会话压缩	experimental.session.compacting	上下文压缩开始前	是	会话压缩	支持在执行上下文压缩前，自定义压缩提示词
文本补全	experimental.text.complete	文本补全完成时触发	是	文本补全结果定制	文本补全完成后，支持对输出内容进行修改
配置加载	config	应用启动配置加载时	否	配置加载	在加载自定义配置时触发，用于执行配置初始化及动态参数注入
通用事件	event	订阅所有Bus事件时触发	否	事件处理	通用事件处理

编写示例

本文以创建个人级别的Hook为例。

以下示例创建一个JsonFormatPlugin插件，在读取.json文件后，自动将挤成一行的压缩JSON，整理成2个空格缩进的格式化形式。

步骤1 进入“~\.codeartsdoer\plugin”，创建并编写文件“JsonFormatPlugin.ts”。

```
// 导入Plugin类型，用于定义插件
import type { Plugin } from "@opencode-ai/plugin"
// 导入文件系统模块，用于读写磁盘文件
import { readFileSync, writeFileSync } from "fs"

// 导出插件实现，Plugin是一个异步函数，接收插件输入参数并返回Hooks对象
export const JsonFormatPlugin: Plugin = async ({}) => {
  return {
    // 监听工具执行后的钩子，在工具执行完成后触发
    "tool.execute.after": async (
      // 输入参数：包含工具名称、会话ID、调用ID、参数等信息
      input: { tool: string; sessionID: string; callID: string; args: any },
      // 输出参数：包含工具执行结果的标题、输出内容、元数据
      output: { title: string; output: string; metadata: any },
    ) => {
      if (input.tool !== "read") return
```

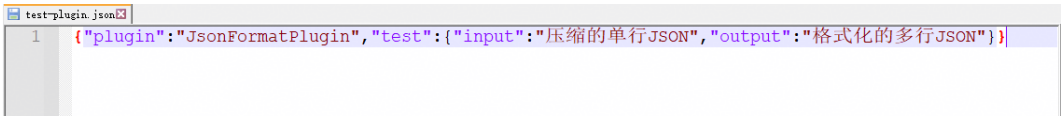


```
    // 获取文件路径，只处理.json文件
    const filePath = input.args?.filePath || input.args?.[0] || ""
    if (!filePath.endsWith(".json")) return

    try {
      const raw = readFileSync(filePath, "utf-8").trim()
      const parsed = JSON.parse(raw)
      const formatted = JSON.stringify(parsed, null, 2) + "\n"
      if (formatted === raw) return

      // 将格式化后的内容写回磁盘文件
      writeFileSync(filePath, formatted, "utf-8")
    } catch {}
  },
 }
}
```

步骤2 进入项目文件夹目录，创建并编写测试文件“test-plugin.json”，内容为压缩格式的JSON。

```
{"plugin":"JsonFormatPlugin","test":{"input":"压缩的单行JSON","output":"格式化的多行JSON"}}
```

图 8-1 格式化前



步骤3 根据不同的开发模式，执行命令：

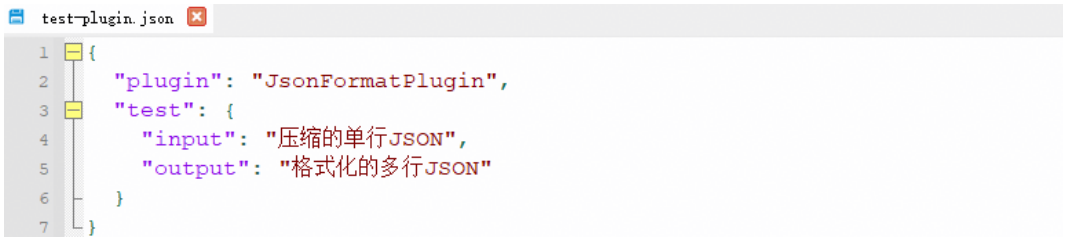
- 如果是TUI开发模式，在TUI对话框中输入如下指令并回车：
  ```
  读取 test-plugin.json
  ```
- 如果是CLI开发模式，请执行如下命令：
  ```
  codearts run "读取 test-plugin.json"
  ```

图 8-2 CLI 开发模式的命令截图

步骤4 执行完成后，“test-plugin.json”文件将被自动格式化为多行缩进形式。再次读取同一文件时，由于已是格式化状态，不会重复处理。

打开目标JSON文件，可发现该JSON文件已被格式化。

图 8-3 格式化后的效果图



----结束

9 代码库索引

华为云码道代码智能体提供了代码库索引功能，帮助您快速检索并获取代码相关信息。该功能支持创建本地个人索引与云端共享索引，灵活适配不同场景需求。

约束与限制

表 9-1 约束与限制

限制类别	具体限制
语言	支持Java、JavaScript、TypeScript和Go。
代码文件总数	• 体验版：最多支持检索**5千**个代码文件。 • 基础版：最多支持检索**5万**个代码文件。 • 专业版：最多支持检索**10万**个代码文件。
功能限制	仅**基础版**和**专业版**支持创建云端**团队级**和**企业级**代码库索引。

本地个人索引和云端索引的差异

表 9-2 本地个人索引和云端索引的差异

对比维度	本地个人索引	云端索引
使用者	当前用户	• 企业：所有企业成员都可用，由企业管理员、团队管理员、团队成员或企业成员创建。 • 团队：仅本团队成员可用，由企业管理员、团队管理员或团队成员创建。 • 个人：仅成员本人可用。
仓库类型	不限制	支持Git、Repo和Github三种仓库。

代码库索引命令

支持以下命令，用于管理代码库索引生命周期。

表 9-3 代码库索引命令

命令	功能说明
codearts codebase --init	创建代码库索引。 首次使用时，在项目目录下运行此命令，即可新建代码库索引。
codearts codebase --detail	查询代码库索引进度
codearts codebase --update	增量更新代码库索引
codearts codebase --delete	删除代码库索引

创建代码库索引

以下介绍创建本地个人索引和创建云端索引的具体步骤。

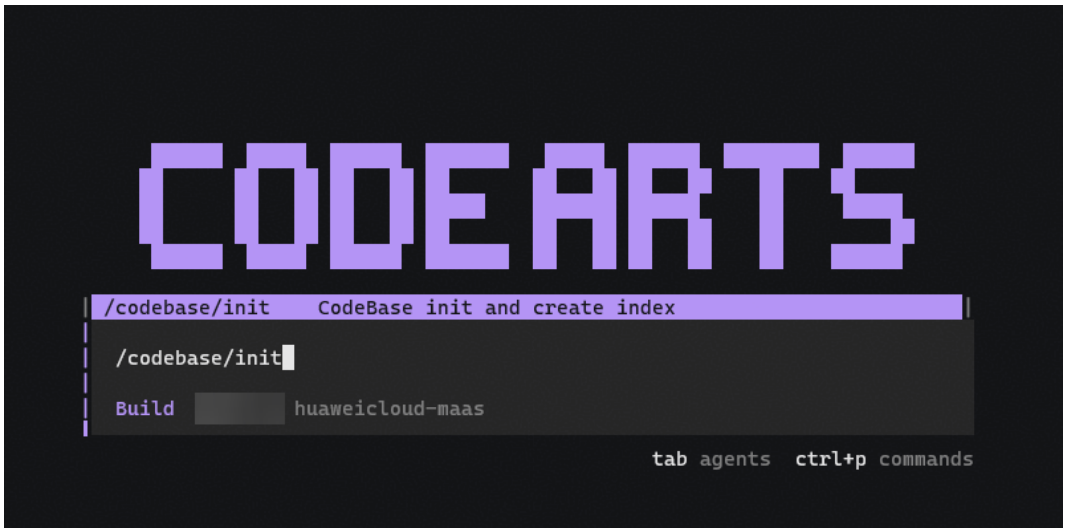
创建本地个人索引

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 **创建代码库索引**。在对话框中输入“/codebase/init”并回车，即可开始创建索引。

图 9-1 输入索引创建命令



步骤3 **查看索引创建进度**。在对话框中输入“/codebase/detail”并回车，即可查看索引创建进度。

图 9-2 查看索引创建进度

步骤4 **增量更新代码库索引**。在对话框中输入“/codebase/update”并回车，即可对新增或修改的内容进行索引。

系统会对比当前状态与上一次增量更新的结果，仅对新增或修改过的代码文件进行索引，跳过未变更文件以节省时间。

步骤5 **删除代码库索引。**如果需要清除当前所有索引数据，可在对话框中输入“/codebase/delete”并回车。

----结束

创建云端索引

步骤1 进入**码道代码智能体控制台**。

步骤2 在左侧导航栏选择“智能体设置 > 代码库索引”，进入代码库索引页面。

步骤3 单击“新建索引”，进入新建代码库索引页面。

步骤4 参考**表9-4**，设置索引基本信息，单击“下一步”。

表 9-4 代码库索引参数说明

参数	说明
可见范围	定义代码库索引的可见范围。**仅华为云码道代码智能体基础版和专业版，支持创建云端团队和企业代码库索引。** • 个人：仅成员本人可用。 • 团队：仅本团队成员可用，由企业管理员、团队管理员或团队成员创建。 • 企业：所有企业成员都可用，由企业管理员、团队管理员、团队成员或企业成员创建。

参数	说明
可用团队	“可见范围”设置为“团队”时，才会显示该配置项。 在下拉列表中选择团队，仅展示当前账号所属团队。
代码源	选择您想要索引的代码仓库来源。 ● Repo：代码托管（CodeArts Repo）中的仓库。 ● Git：通用Git仓库。 ● Github：GitHub平台仓库，支持公开仓库及经授权的私有仓库。
选择项目	“代码源”设置为“Repo”时，才会显示该配置项。 从已关联的代码源中，选择具体要索引的项目。
获取授权	选择一种方式来授权系统访问您的代码源。 ● 通过服务授权：在下拉框中选择服务扩展点，该扩展点对应Repo、Github或Git仓库的连接信息。 若还未创建服务扩展点，单击“添加”，参考表9-5创建服务扩展点。 ● 通过个人访问令牌：使用用户个人账户生成的访问令牌进行认证。
默认分支	“代码源”设置为“Git”时，才会显示该配置项。 输入Git仓库的分支名称，分支名称不能超过200个字符。

图 9-3 新建服务扩展点

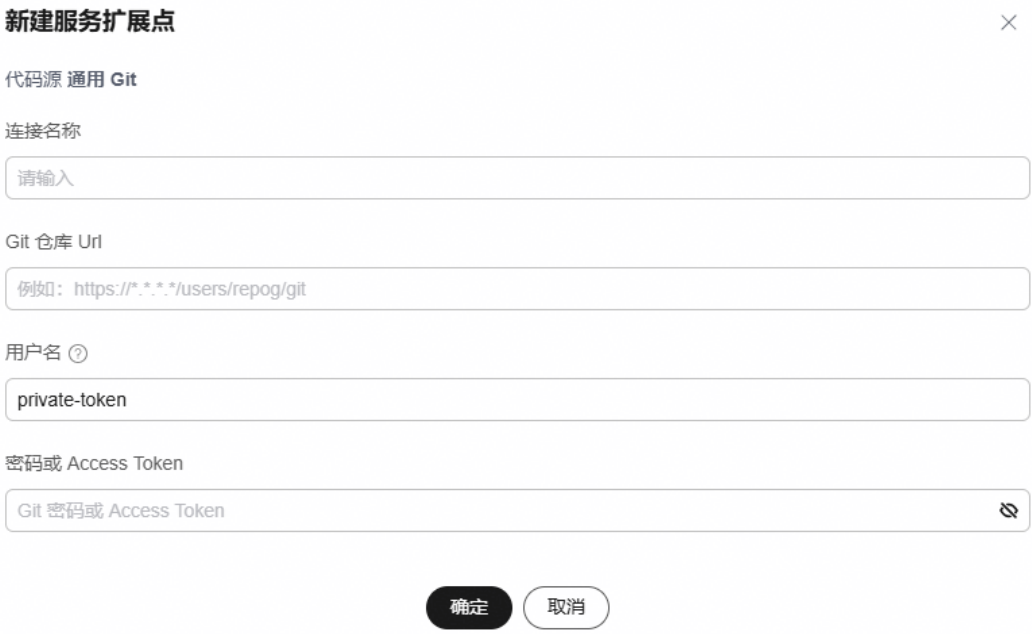


表 9-5 服务扩展点参数说明

参数	说明
连接名称	自定义连接仓库的名称。 命名规范：由中文、英文、数字、下划线和中划线组成，不能以下划线开头。
Git仓库Url	“代码源”设置为“Repo”或“Git”时，才显示该配置项。 Repo/Git仓库的HTTPS访问地址。
用户名	“代码源”设置为“Repo”或“Git”时，才显示该配置项。 Repo/Git仓库HTTPS协议的认证用户名。
密码或Access Token	“代码源”设置为“Git”时，才显示该配置项。 Git仓库HTTPS协议的认证密码或Access Token。
Access Token	“代码源”设置为“Repo”或“Github”时，才显示该配置项。 Repo/Github仓库HTTPS协议的Access Token。

步骤5 选择代码库和语言，单击“确定”，完成代码库索引的创建。

在代码库仓索引页面，查看新创建的代码库索引。当该代码库索引的状态由“解析中”变为“解析完成”时，表示解析代码库索引成功。

步骤6 查看索引创建进度。

1. 在Git/Repo/Github仓库项目的根目录下，鼠标右键单击空白处，选择“在终端中打开”。
2. 在终端中输入“/codearts”并回车，进入TUI开发环境，系统将自动从云端下载团队代码库索引。
3. 在TUI对话框中，输入“/codebase/detail”并回车，即可查看索引创建进度。

----结束

10 LSP

语言服务器协议（Language Server Protocol，简称LSP）是一种用于提供语言服务的标准化协议。LSP将语言服务与IDE进行了解耦，由独立的语言服务器进程来提供代码分析、快速导航及符号查询等核心能力。

华为云码道代码智能体已预置本地LSP功能。开启该功能后，智能体可基于语言服务提供代码分析结果、代码跳转及符号信息，实现更精准的代码理解和代码修改操作。

支持的主流语言 LSP 服务器

表 10-1 支持的主流语言 LSP 服务器

LSP服务器	扩展名	要求
jdtls	.java	请确保已安装Java SDK 21或更高版本。
typescript-language-server	.ts、.tsx、.js、.jsx、.mjs、.cjs、.mts、.cts	请确保已安装TypeScript。
pyright	.py、.pyi	依赖pyright，请确保已安装该工具。
gopls	.go	依赖Go语言环境，请确保go命令已安装并配置到系统环境变量中。

开启本地 LSP

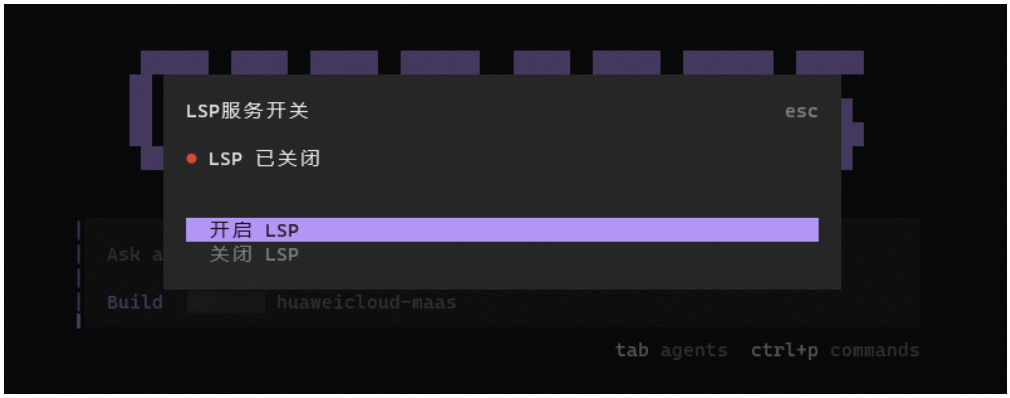
步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 开启本地LSP能力。

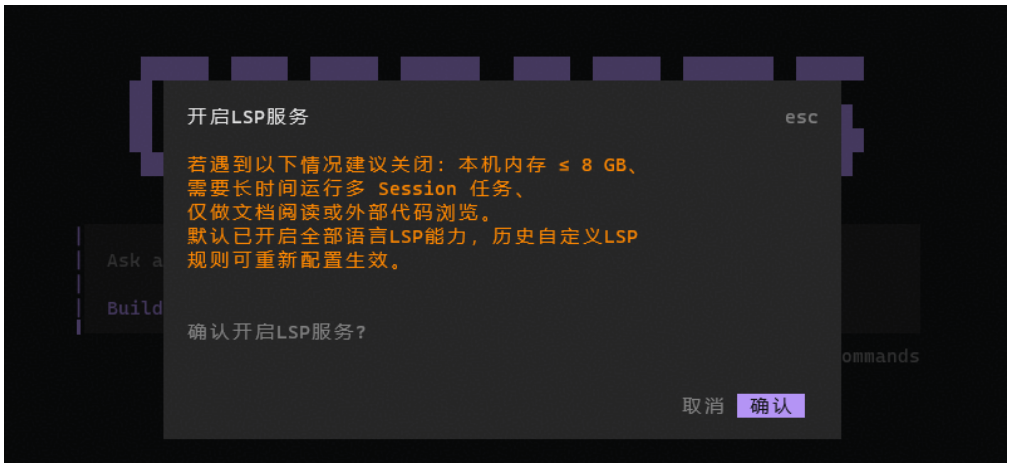
1. 在TUI对话框中，输入“/lsp-toggle”并回车，进入LSP服务开关页面。

图 10-1 开启 LSP 页面



2. 按回车，然后单击“确认”，开启LSP服务。

图 10-2 开启 LSP 服务



步骤3 重启码道CLI，使修改生效。

----结束

相关文档

启用本地LSP服务后，若在智能体发起会话时收到“LSP服务不可用”的提示，可参考**LSP服务不可用**中的操作进行排查和处理。

11 定时任务

通过自然语言创建和管理定时任务，包括周期性任务（如每日代码检查）和一次性任务（如明日提交提醒），实现灵活的任务调度与控制。

定时任务命令

表 11-1 定时任务命令

开发环境	命令	功能说明
TUI	/schedule 自然语言提示词	在TUI开发环境中，您可以通过输入/schedule命令，或直接使用自然语言描述来创建定时任务。
CLI	codearts run --command schedule "*提示词*" codearts run "提示词"	在CLI开发环境中，您可以通过输入/schedule命令，或直接使用自然语言描述来创建定时任务。

⚠ 注意

通过命令创建的定时任务会立即执行，使用自然语言描述创建的定时任务则不会立即执行。

定时任务配置文件

在码道CLI中运行定时任务依赖以下文件，其中的数据均由系统自动生成。

表 11-2 定时任务依赖文件

文件类型	文件路径	说明
任务数据文件（配置文件）	~/.codeartsdoer/cli-data/cron/scheduled_tasks.json	存储所有定时任务的定义，包括间隔、提示词、状态等。数据文件包含的具体字段，如表11-3。
执行日志文件（执行记录）	~/.codeartsdoer/cli-data/cron/cron_log/task_*{TASK_ID}*.json	记录每个任务的每次执行结果，包括状态、耗时、结果摘要。

表 11-3 数据文件字段说明

字段	取值范围 / 格式	说明	示例
id	8字符随机ID（系统自动生成）	任务的唯一标识。	10e366d0
cron	5段Cron表达式：minute hour day month weekday 支持：*、*/N、N、N-M、N,M,O 字段范围：分钟（0–59）、小时（0–23）、日（1–31）、月（1–12）、星期（0-6，0为周日）	执行时间计划（本地时间），用于定时触发任务。	*/5 * * * *（每5分钟）、0 9 * * *（每天9点）
prompt	任意文本	任务触发时执行的AI提示词（指令内容）。	检查本地C盘剩余存储空间
projectPath	绝对路径	所属项目路径（系统自动绑定当前项目上下文）。	D:\tmp\test_hc_cli
status	• active：运行中 • paused：暂停 • expired：已过期 • deleted：已删除	任务当前状态（系统自动管理，初始值为active）。	active
recurring	• true：循环任务，按cron间隔反复执行 • false：单次任务，执行一次后自动结束	是否启用循环执行机制。	true

字段	取值范围 / 格式	说明	示例
durable	• true：持久化定时任务。持久化任务跨重启保留，退出后下次启动会自动恢复并继续执行。只有当用户明确要求“跨重启保留”、“持续每天运行”、“永久保留”时，才会创建持久化任务。 • false：会话级定时任务。默认创建的任务为会话级，退出程序后任务不再保留，下次启动不会恢复。适合临时提醒、短期轮询等场景。	是否为持久化任务，默认为false。	false
createdAt	Unix时间戳（毫秒）	任务创建时间，由系统自动生成。	1783996637621
humanSchedule	中文描述	人类可读的调度描述，系统根据cron表达式自动生成。	每分钟、每天09:00、每5分钟
sessionID	字符串	会话标识符。定时任务会新建会话，并随任务刷新而更新。	ses_0a16ceba1ffeXQHnpNW0jh20J8
agentID	内置Agent名称	执行任务所使用的Agent，由系统自动获取。	build、code
modelID	已配置的模型ID	执行任务使用的模型，由系统自动获取。	deepseek-v3.2
providerID	已配置的提供商ID	执行任务使用的模型提供商，由系统自动获取。	huaweicloud-maas
isCustom	• true：自定义Agent • false：内置Agent	是否使用自定义Agent，由系统自动判断。	false
sessionType	内核会话（kernel）、其他会话类型	会话类型，由系统自动获取。	kernel
category	• code：代码类 • office：办公类	任务分类，由系统自动获取。	code

在 TUI 中创建定时任务

在TUI开发环境中，您可以使用命令或自然语言两种方式来创建定时任务。需要注意的是，通过命令创建的定时任务会立即执行，而使用自然语言描述创建的定时任务则不会立即执行。

使用/schedule 命令创建定时任务

命令语法：

```
/schedule [时间间隔] <提示词>
```

表 11-4 /schedule 命令参数说明

参数	描述
时间间隔	可选参数，设置定时任务每隔多久执行一次。省略时默认每隔10分钟执行一次。 格式：数字 + 单位字母。单位字母包含s（秒）、m（分钟）、h（小时）和d（天）四种。
提示词	必选参数，您希望AI执行的指令。

注意

首次提交定时任务指令后，AI将立即执行一次指定的提示词，并随后设置一个定时任务持续运行。后续的定时任务将在一个独立的会话（session）中生成，您可以通过“/sessions”命令查看所有会话的情况。例如，下发一个“/schedule 5m 检查部署状态”指令，AI会立即检查一次，然后每5分钟自动检查部署状态。

使用示例：

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 通过“/schedule”命令，创建定时任务。

1. 在TUI对话框中，输入“/”，选择“/schedule”。
2. 在“/schedule”命令后，输入如下指令，并按Enter键。
   ```
   1m 按照后缀递增创建一个txt文件，并写入当前系统日期与时间
   ```
3. AI会创建定时任务并立即执行一次定时任务，同时生成表11-5中两个文件。

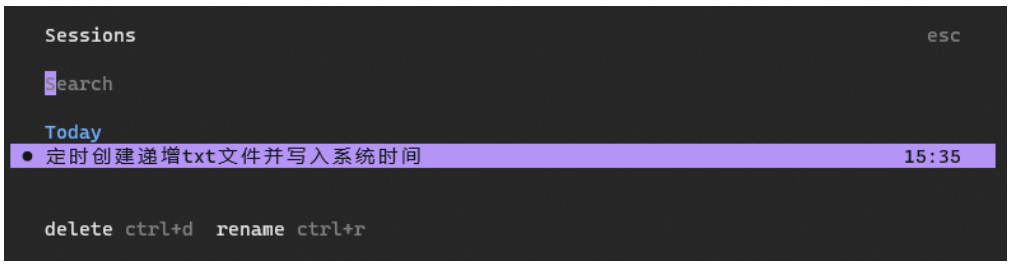
图 11-1 创建会话级定时任务

表 11-5 生成文件

文件名	存储路径	说明
scheduled_tasks.json	%USERPROFILE%/.codeartsdoer/cli-data/cron/	记录当前任务信息。
task_*{TASK_ID}*.json	%USERPROFILE%/.codeartsdoer/cli-data/cron/cron_log/	记录当前任务的执行记录。

4. 通过“/sessions”命令，查看定时任务新建的会话。

图 11-2 查看定时任务新建的会话



----结束

使用自然语言创建定时任务

除了通过“/schedule”命令创建定时任务，还支持通过自然语言描述来创建定时任务。**通过自然语言创建的定时任务，不会立即执行。**

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 在TUI对话框中，输入您希望AI执行的操作，并按Enter键。

- **通过自然语言创建循环任务示例**

  表 11-6 通过自然语言创建循环任务

提示词	AI会做的事
每30分钟检查一下生产环境的错误日志	创建循环任务，每30分钟执行一次
每天早上9点运行单元测试	创建循环任务，每天9:00执行一次
每小时扫描一次安全漏洞	创建循环任务，每1小时执行一次

- **通过自然语言创建单次任务示例**

  表 11-7 通过自然语言创建单次任务

提示词	AI会做的事
明天上午9点提醒我review PR#42	创建单次任务，明天9:00执行一次后自动结束
3分钟后提醒我确认数据库迁移结果	创建单次任务，3分钟后执行一次
下周一10点提醒我提交周报	创建单次任务，下周一10:00执行一次

----结束

在 CLI 中创建定时任务

在CLI开发环境中，您可以使用命令或自然语言两种方式来创建定时任务。需要注意的是，通过命令创建的定时任务会立即执行，而使用自然语言描述创建的定时任务则不会立即执行。

⚠ 注意

CLI对话是一次性的，因此CLI场景下创建的定时任务仅会写入到scheduled_tasks.json配置文件中，待启动TUI或者其他码道CLI进程时才会执行定时任务。请保持码道CLI持续运行，确保任务能够按时、稳定地执行。

使用命令创建定时任务

命令语法：

```
codearts run --command schedule "提示词"
```

表 11-8 命令参数说明

参数	描述
提示词	必选参数，您希望AI执行的操作。

使用示例：

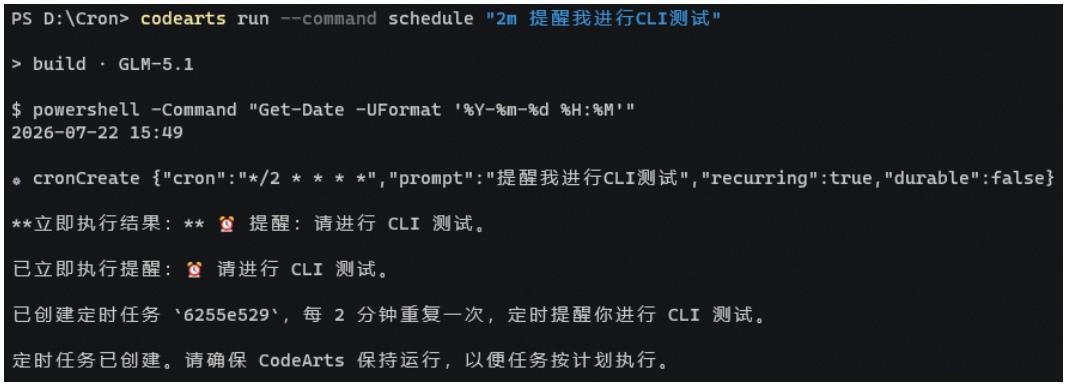
步骤1 在CLI开发环境中，输入创建定时任务命令并回车。

例如，创建一个每两分钟提醒我进行CLI测试的定时任务，您可使用如下提示词。**如果指令中没有明确时间，则创建默认每10分钟执行一次的定时任务**（如codearts run --command schedule "列出所有定时任务"）。

```
codearts run --command schedule "2m 提醒我进行CLI测试"
```

步骤2 AI会根据您输入的指令，创建定时任务并立即执行一次定时任务。

图 11-3 创建并执行定时任务



----结束

使用自然语言创建定时任务

步骤1 在CLI开发环境中，输入创建定时任务命令并回车。

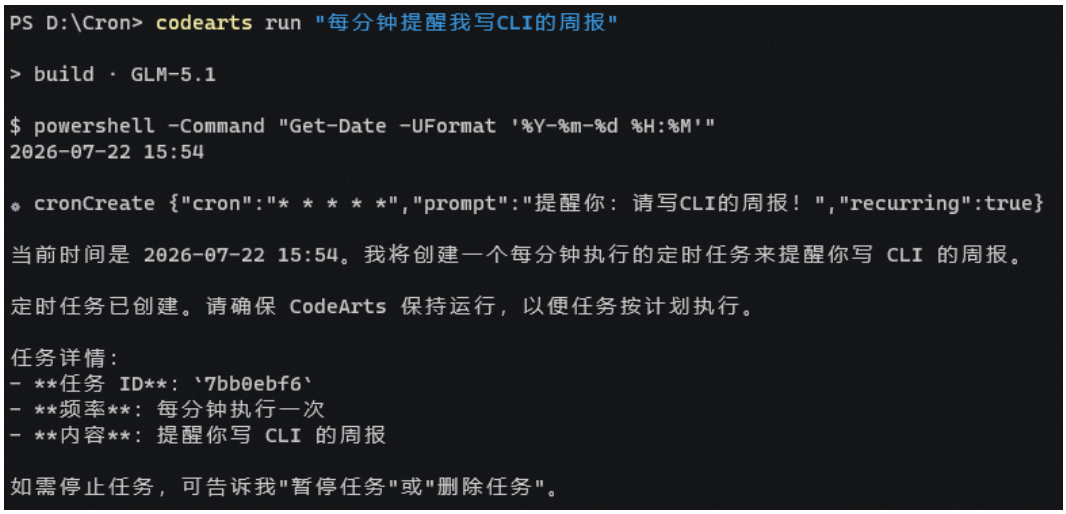
```
codearts run "提示词"
```

例如，创建一个每分钟提醒我写CLI周报的定时任务，您可使用如下提示词。

```
codearts run "每分钟提醒我写CLI的周报"
```

步骤2 AI会根据您输入的指令，创建定时任务。

图 11-4 创建定时任务



----结束

管理定时任务

不支持通过命令管理定时任务，仅执行通过自然语言描述来管理已创建的定时任务。

表 11-9 管理定时任务

输入自然语言提示词	执行结果
• TUI：列出所有的定时任务 • CLI：codearts run "列出所有的定时任务"	展示全部定时任务列表。
• TUI：查看暂停的定时任务 • CLI：codearts run "列出所有的定时任务"	仅展示处于暂停状态的任务。
• TUI：查看*任务ID*任务详情 • CLI：codearts run "查看*任务ID*任务详情"	展示该任务的详情，包括执行历史和下次运行时间
• TUI：暂停*任务ID* • CLI：codearts run "暂停*任务ID*"	任务停止执行，暂停不会丢失任何配置，恢复后任务继续按原间隔执行。
• TUI：恢复*任务ID* • CLI：codearts run "恢复*任务ID*"	将暂停的任务重新激活，按原计划继续运行。
• TUI：删除*任务ID* • CLI：codearts run "删除*任务ID*"	永久移除该任务，不可恢复。

输入自然语言提示词	执行结果
• TUI：立即执行定时任务*任务ID* • CLI：codearts run "立即执行定时任务*任务ID*"	马上触发一次执行，不影响原有的定时计划。例如，可以立即执行处于暂停状态的定时任务。

12 自定义模型

码道CLI新增对接第三方大语言模型的能力。若您已购买华为云码道代码智能体专业版套餐，同时支持在控制台添加第三方大语言模型，具体请参见**配置企业自定义模型**。

约束与限制

- 接入第三方企业自定义模型仅支持**OpenAI规范的Chat Completions API格式**和**Anthropic规范的Messages API格式**。
- 在控制台新建的自定义模型所有智能体均可调用，在客户端新建的自定义模型仅系统内置智能体、AgentTeam和自定义智能体可调用，智能问答无法调用。
- 相同模型提供商和模型ID的模型仅允许添加一次。

前提条件

- 已将账号添加至**企业成员**中，并**启用席位**。
- 若您已购买华为云码道代码智能体专业版套餐，请确保企业管理员已在控制台开启成员自定义模型权限，具体操作请参考**配置企业自定义模型**。

自定义模型配置文件

在码道CLI中，通过配置文件对自定义模型进行管理。在CLI/TUI模式下，不支持使用命令创建、修改或删除自定义模型，所有配置均需手动编辑文件。

自定义模型配置文件及其存放路径：“~/.codeartsdoer/codearts_cli.json”，“~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于“/home/用户名/”。

配置文件以provider为顶层键，每个提供商ID对应一个配置块。自定模型配置文件完整的结构如下：

```
{
  "provider": {
    "<提供商ID>": {
      "name": "<提供商显示名称>",
      "npm": "<SDK包名>",
      "api": "<提供商API地址>",
      "env": ["<环境变量名>", ...],
      "whitelist": ["<模型ID>", ...],
      "blacklist": ["<模型ID>", ...],
      "options": {
        "apiKey": "<API密钥>",
        "baseURL": "<API基础地址>",
```

```
    "enterpriseUrl": "<企业版URL>",
    "setCacheKey": true,
    "timeout": 300000,
    "chunkTimeout": 60000
   },
   "models": {
    "<模型ID>": {
     "id": "<模型API标识>",
     "name": "<显示名称>",
     "family": "<模型家族>",
     "reasoning": true,
     "attachment": true,
     "temperature": true,
     "tool_call": true,
     "limit": { "context": 128000, "input": 120000, "output": 8000 },
     "cost": {
      "input": 3.0,
      "output": 15.0,
      "cache_read": 0.5,
      "cache_write": 2.0,
      "context_over_200k": {
       "input": 6.0,
       "output": 30.0,
       "cache_read": 1.0,
       "cache_write": 4.0
      }
     },
     "modalities": {
      "input": ["text", "image", "pdf"],
      "output": ["text"]
     },
     "headers": { "X-Custom-Header": "value" },
     "status": "active",
     "provider": { "npm": "<SDK包名>", "api": "<API地址>" },
     "variants": { ... }
    }
   }
  }
 }
}
```

说明

apiKey是您的核心资产，请不要轻易外泄。当您配置apiKey后，重启TUI/CLI，系统会自动对apiKey进行加密，后续配置文件中展示的就是密文。

提供商ID是配置文件中的顶层键名，用于标识模型来源。常见提供商ID，如**表12-1**所示。您也可以使用任意自定义字符串作为提供商ID（如my-server），只要对应模型支持OpenAI兼容的API格式即可。

表 12-1 常见提供商 ID

提供商ID	说明
openai	OpenAI
deepseek	DeepSeek
glm	glm
kimi	kimi

表 12-2 自定义模型配置文件参数说明

字段名	类型	必填	说明
provider.<提供商ID>.name	字符串	否	提供商的显示名称，缺省时使用提供商ID。
provider.<提供商ID>.npm	字符串	否	SDK包名，缺省时默认为@ai-sdk/openai-compatible。
provider.<提供商ID>.api	字符串	否	提供商的API URL，与options.baseURL功能类似。
provider.<提供商ID>.env	字符串数组	否	环境变量名列表，码道CLI会依次读取这些环境变量作为API密钥的回退来源。
provider.<提供商ID>.whitelist	字符串数组	否	模型白名单，仅列表中的模型会被加载（对内置提供商可用来过滤不需要的模型）。
provider.<提供商ID>.blacklist	字符串数组	否	模型黑名单，列表中的模型不会被加载。
provider.<提供商ID>.options.apiKey	字符串	是	API密钥。首次加载时明文密钥会自动加密为enc:v1:格式并写回配置文件。
provider.<提供商ID>.options.baseURL	字符串	是	API基础地址，如https://your-server.com/v1。
provider.<提供商ID>.options.enterpriseUrl	字符串	否	GitHub Enterprise URL，仅用于Copilot提供商的企业版认证。
provider.<提供商ID>.options.setCacheKey	布尔值	否	是否启用promptCacheKey，默认为false。
provider.<提供商ID>.options.timeout	数字或false	否	请求超时时间，单位为毫秒，默认为300000（5分钟）。设置为false，可禁用超时。
provider.<提供商ID>.options.chunkTimeout	数字	否	SSE流式响应的块间超时时间，单位为毫秒。超过该时间未收到新数据块则中断请求。

字段名	类型	必填	说明
provider.<提供商ID>.options.*	任意类型	否	options还支持任意额外字段（catchall），提供商的SDK实现可能读取这些自定义选项。
provider.<提供商ID>.models.<模型ID>.id	字符串	否	模型的API标识（实际发送给提供商的模型ID），缺省时使用配置键名。
provider.<提供商ID>.models.<模型ID>.name	字符串	否	码道CLI客户端显示的模型名称，缺省时使用配置键名。
provider.<提供商ID>.models.<模型ID>.family	字符串	否	模型家族名称（如gpt、claude），用于模型分组。
provider.<提供商ID>.models.<模型ID>.reasoning	布尔值	否	是否支持推理/思考能力。
provider.<提供商ID>.models.<模型ID>.attachment	布尔值	否	是否支持附件上传（图片、PDF等）。
provider.<提供商ID>.models.<模型ID>.temperature	布尔值	否	是否支持温度参数调节。
provider.<提供商ID>.models.<模型ID>.tool_call	布尔值	否	是否支持工具/函数调用，默认为true。
provider.<提供商ID>.models.<模型ID>.limit.context	数字	否	最大上下文窗口大小（Token）。
provider.<提供商ID>.models.<模型ID>.limit.input	数字	否	最大输入Token数。
provider.<提供商ID>.models.<模型ID>.limit.output	数字	否	最大输出Token数。
provider.<提供商ID>.models.<模型ID>.cost.input	数字	否	输入Token单价（每百万Token美元价格）。
provider.<提供商ID>.models.<模型ID>.cost.output	数字	否	输出Token单价。

字段名	类型	必填	说明
provider.<提供商ID>.models.<模型ID>.cost.cache_read	数字	否	缓存读取单价。
provider.<提供商ID>.models.<模型ID>.cost.cache_write	数字	否	缓存写入单价。
provider.<提供商ID>.models.<模型ID>.cost.context_over_200k	对象	否	超过200K上下文时的替代定价，包含input、output、cache_read和cache_write。
provider.<提供商ID>.models.<模型ID>.modalities.input	字符串数组	否	输入支持的模态，包括text、audio、image、video、pdf。
provider.<提供商ID>.models.<模型ID>.modalities.output	字符串数组	否	输出支持的模态，包括text、audio、image、video、pdf。
provider.<提供商ID>.models.<模型ID>.headers	字符串键值对象	否	自定义HTTP请求头，随模型请求发送。
provider.<提供商ID>.models.<模型ID>.status	字符串	否	模型状态标记，包括active、beta、alpha、deprecated。
provider.<提供商ID>.models.<模型ID>.provider.npm	字符串	否	该模型使用的SDK包名，覆盖提供商级默认。
provider.<提供商ID>.models.<模型ID>.provider.api	字符串	否	该模型的API URL，覆盖提供商级默认。
provider.<提供商ID>.models.<模型ID>.variants	键值对象	否	变体配置。

自定义模型配置示例

以最小化配置为例，演示如何通过设置**提供商ID**、**API密钥**及**模型**来自定义模型。以Windows操作系统为例。

步骤1 进入%USERPROFILE%/.codeartsdoer路径，找到并打开codearts_cli.json配置文件。

步骤2 在codearts_cli.json配置文件中，增加provider内容。

```
{
  "$schema": "https://opencode.ai/config.json",
  "provider": {
    "deepseek": {
      "options": {
```



```
      "apiKey": "sk-***",
      "baseURL": "https://api.deepseek.com/v1"
    },
    "models": {
      "deepseek-chat": {
        "name": "DeepSeek Chat"
      },
      "deepseek-reasoner": {
        "name": "DeepSeek Reasoner",
        "reasoning": true
      }
    }
  }
 },
 "mcp": {},
 "lsp": false
}
```

其中，apiKey需要您到DeepSeek官网购买后填写。

步骤3 保存文件并退出。

步骤4 查看自定义模型是否添加成功。

- 在CLI环境中执行以下命令，从回显的模型列表中检查是否包含自定义模型。
  ```
  codearts models
  ```
- 在TUI环境中输入如下斜杠命令，进入模型选择页面确认是否存在自定义模型。
  ```
  /models
  ```

----结束

指定运行模型

在码道CLI中，您可以通过指定特定模型来运行您的任务。如下命令中，*provider*为模型的供应商，*model*为模型的名称。

- TUI开发环境
  - 在**启动**TUI开发模式时，可以通过执行如下命令，指定运行的模型。
    ```
    codearts --model provider/model
    或使用短参数
    codearts -m provider/model
    ```
  - 在TUI开发环境中，只需输入“/models”命令即可快速切换模型。
- CLI开发环境
  ```
  codearts run --model provider/model "您的问题"
  ```


13 MCP

码道CLI的MCP（Model Context Protocol）服务器支持两种运行模式：

- 本地模式：在用户当前电脑上运行的进程，码道CLI会直接在本地启动它并通信。
- 远程模式：运行在其他机器或者服务器上的进程，码道CLI通过网络（HTTP/WS）和它通信。

配置MCP文件后，您可以在TUI中可以执行“/mcps”命令查看。

13.1 默认 MCP 配置

支持配置项目级与个人级的MCP，配置路径如**表1 MCP可配置路径表**所示。

表 13-1 MCP 配置路径

级别	作用范围	配置路径
项目级	仅当前项目 优先级：项目级 > 个人级	“项目根目录/.codeartsdoer/codearts_cli.jsonc”或者“项目根目录/.codeartsdoer/codearts_cli.json”
个人级	当前用户的所有项目	“~/.codeartsdoer/codearts_cli.jsonc”或者“~/.codeartsdoer/codearts_cli.json” “~”表示当前用户的主目录，Windows下等同于“C:\Users\用户名\”，macOS下等同于“/Users/用户名/”，Linux下等同于/home/用户名/

本地 MCP

以下介绍如何配置本地MCP。

参数配置

表 13-2 本地 MCP 配置参数表

参数名称	参数类型	说明
type	字符串	必填，值必须为“local”
command	字符串数组	必填，表示启动MCP服务的执行命令及入参
environment	对象	运行服务器时设置的环境变量
enabled	布尔值	控制MCP服务器启用状态
timeout	数字	获取MCP服务工具响应超时时间，单位毫秒，默认100000毫秒

配置示例

本文以Windows系统为例，演示个人环境下本地MCP服务器的配置方法。

步骤1 在“C:/Users/用户名/.codeartsdoer/”路径下，创建文件“codearts_cli.jsonc”，**写入如下配置并重启码道CLI**：

```
{
  "mcp": {
    // 自定义的MCP服务名称：字符分析工具
    "mcp-character-tools" : {
      //必填项，启动命令：使用npx启动本地MCP服务
      "command" : [ "npx", "mcp-character-tools" ],
      //必填项，服务类型：表示本地运行的MCP服务器
      "type": "local"
    }
  }
}
```

步骤2 进入TUI开发环境。

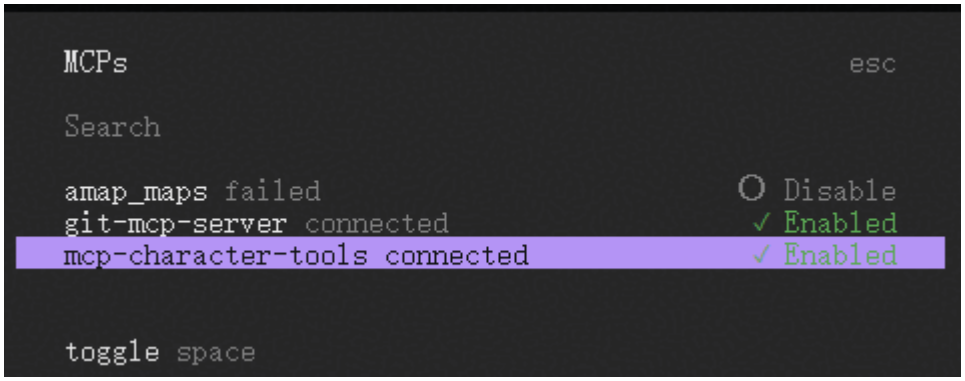
1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤3 输入如下命令，查看此配置是否生效。

```
/mcps
```

MCP服务器“mcp-character-tools”的状态为“Enabled”，表示此MCP服务器已启用。

图 13-1 启用状态



步骤4 在TUI对话框发送如下指令。

```
mcp-character-tools该如何使用?
```

返回功能介绍和使用方法，表示此本地MCP服务器配置已生效。

----结束

远程 MCP

以下介绍如何配置远程MCP。

参数配置

表 13-3 远程 MCP 配置参数表

参数名称	参数类型	说明
type	字符串	必填，值必须为“remote”
url	字符串	必填，远程MCP服务器的URL
headers	对象	随请求发送的请求头
environment	对象	运行服务器时设置的环境变量
oauth	对象	OAuth身份验证配置
enabled	布尔值	控制MCP服务器启用状态
timeout	数字	获取MCP服务工具响应超时时间，单位毫秒，默认5000毫秒

配置示例

本文以Windows系统为例，演示个人环境下远程MCP服务器的配置方法。

步骤1 在“C:/Users/用户名/.codeartsdoer/”路径下，创建文件“codearts_cli.jsonc”，**写入如下配置并重启码道CLI**：

```
{
  "mcp": {
```

```
    // EdgeOne Pages Deploy MCP 是一项专用服务，能够将 Web 应用快速部署到 EdgeOne Pages 并生成公开访问链接。这使您能够立即预览和分享 AI 生成的网页内容。
    "edgeone-pages": {
      "type": "remote",
      "url": "https://mcp-on-edge.edgeone.site/mcp-server",
      "enabled": true,
      "timeout": 45000,
      "headers": {
        // 接口请求鉴权头，非必填
        "Authorization": "Bearer edgeone_token_your_secret"
      }
    }
  }
}
```

步骤2 进入TUI开发环境。

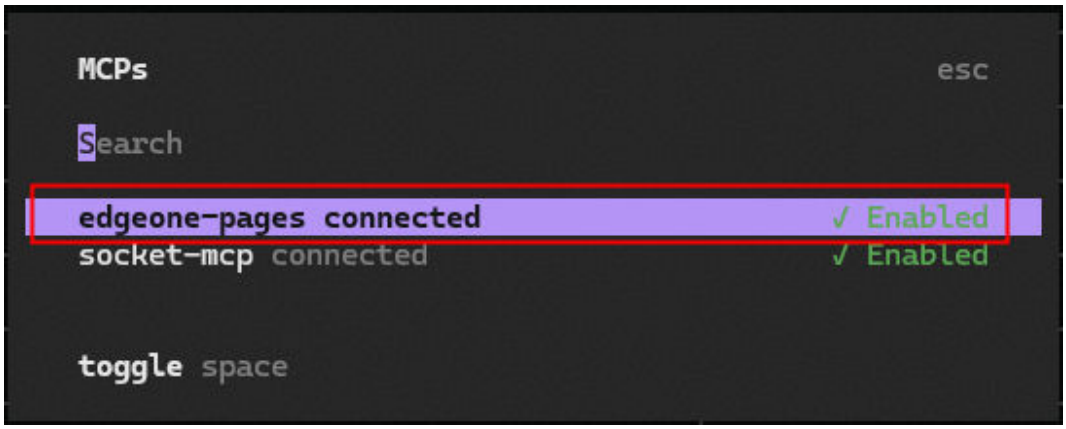
1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤3 输入如下命令，查看此配置是否生效。

```
/mcps
```

MCP服务器“edgeone-pages”的状态为“Enabled”，表示此MCP服务器已启用。

图 13-2 启用状态

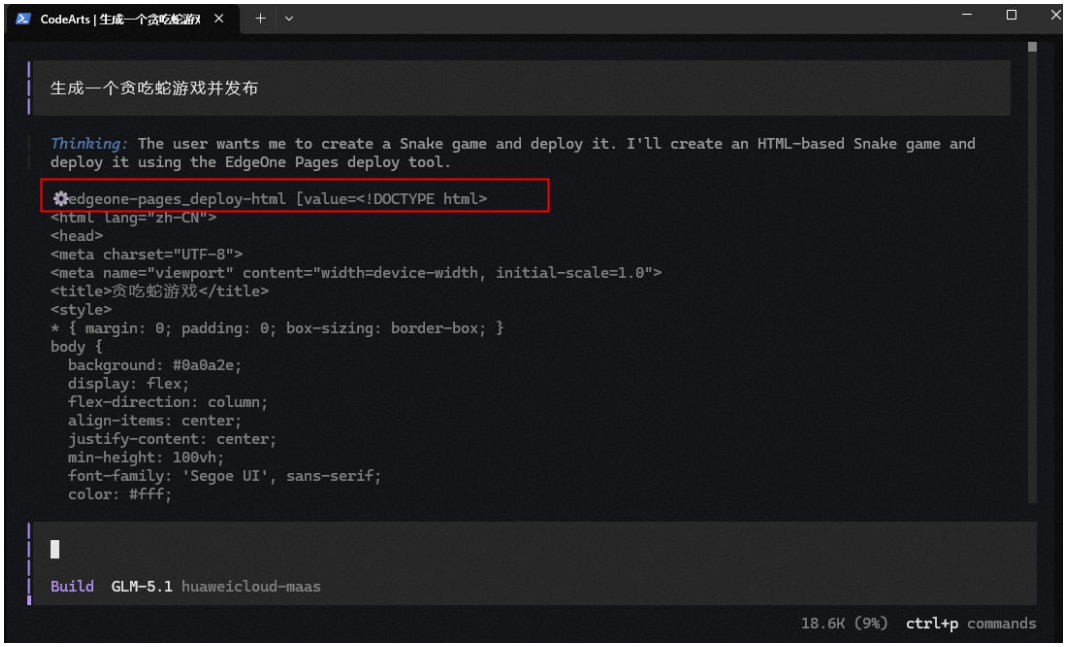


步骤4 在TUI开发模式输入如下指令并回车。

```
生成一个贪吃蛇游戏并发布
```

AI会调用MCP服务器，并完成贪吃蛇游戏的创建。

图 13-3 MCP 服务器调用成功



----结束

13.2 Claude Code 格式

除码道CLI原生配置格式外，码道CLI还兼容Claude Code的MCP配置格式。如果您已有Claude Code的MCP配置文件，可直接在码道CLI中使用，无需重新编写。两种格式功能等价，选择任一即可。

支持配置项目级的MCP，配置路径如**表1 MCP可配置路径表**所示。

表 13-4 MCP 配置路径

级别	作用范围	配置路径
项目级	仅当前项目	项目根目录/.codeartsdoer/mcp/mcp_settings.json

本地 MCP

采用stdio传输模式，以拉起本地进程的方式对接MCP服务端。

参数配置

表 13-5 本地 MCP 配置选项表

参数名称	参数类型	说明
command	字符串	必填，启动MCP服务器进程的命令，如npx、uvx、node、python
args	字符串数组	启动命令的参数列表
env	对象	环境变量，用于传递API密钥等配置
disabled	布尔值	控制MCP服务器启用状态
timeout	数值	连接超时时间，单位毫秒

配置示例

本文以Windows系统为例，演示本地MCP服务器的配置方法。

步骤1 在“D:/tmp/test_hc_cli/.codeartsdoer/”路径下，创建路径文件“mcp/mcp_settings.json”，**写入如下配置并重启码道CLI**：

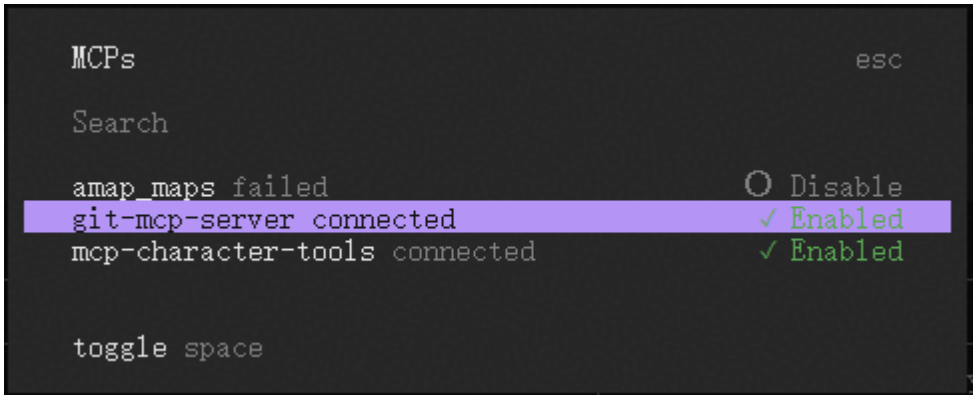
```
{
  "mcpServers": {
     "git-mcp-server": {
     "transportType": "stdio",
     "disabled": false,
     "timeout": 100000,
     "args": [
       "@cyanheads/git-mcp-server"
     ],
     "command": "npx",
     "env": {
       "GIT_SIGN_COMMITS": "false",
       "MCP_LOG_LEVEL": "info"
     }
    }
  }
}
```

步骤2 输入如下命令，查看此配置是否生效。

```
/mcps
```

MCP服务器“git-mcp-server”的状态为“Enabled”，表示此MCP服务器已启用。

图 13-4 启用状态



步骤3 在TUI对话框发送如下指令。

```
git-mcp-server能做什么?
```

返回功能介绍和使用方法，表示本地MCP服务器配置已生效。

----结束

远程 MCP

采用StreamableHttp传输模式，经由HTTP协议对接远程MCP服务端。

参数配置

表 13-6 远程 MCP 配置选项表

参数名称	参数类型	说明
url	字符串	必填，MCP服务器的HTTP地址
headers	对象	HTTP请求头，用于认证
disabled	布尔值	控制MCP服务器启用状态
timeout	数值	连接超时时间，单位毫秒
autoApprove	字符串数组	自动批准的工具列表

配置示例

本示例以Windows系统为例。

步骤1 在“D:/tmp/test_hc_cli/.codeartsdoer/”路径下，创建路径文件“mcp/mcp_settings.json”，**写入如下配置并重启码道CLI**：

```
{
  "mcpServers": {
    "socket-mcp": {
      "url": "https://mcp.socket.dev/",
      "headers": {
        "Content-Type": "application/json"
```

```
    },
    "timeout": 30000,
    "disabled": false
  }
 }
}
```

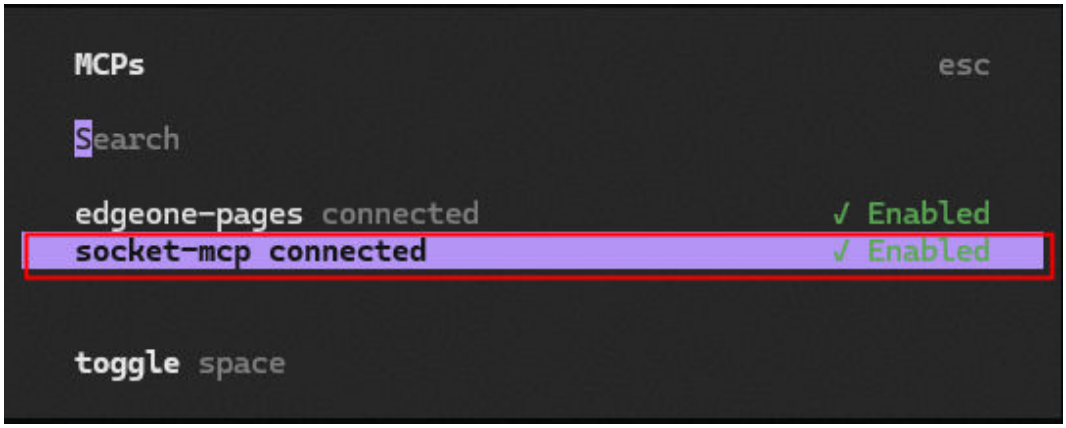
该MCP服务器是一个依赖项扫描服务，用于评估软件包的质量和安全性。

步骤2 输入如下命令，查看此配置是否生效。

```
/mcps
```

MCP服务器“socket-mcp”的状态为“Enabled”，表示此MCP服务器已启用。

图 13-5 启用状态

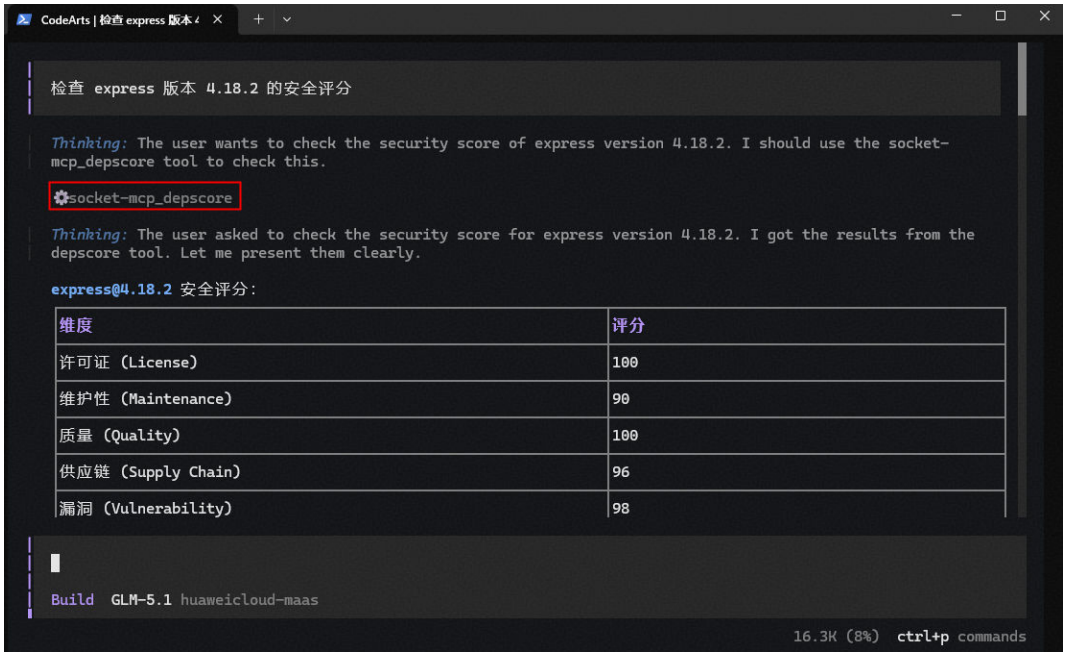


步骤3 在TUI对话框输入如下指令并回车。

```
检查express版本4.18.2的安全评分
```

返回安全评分结果，表示此远程MCP服务器配置已生效。

图 13-6 MCP 服务器配置生效图



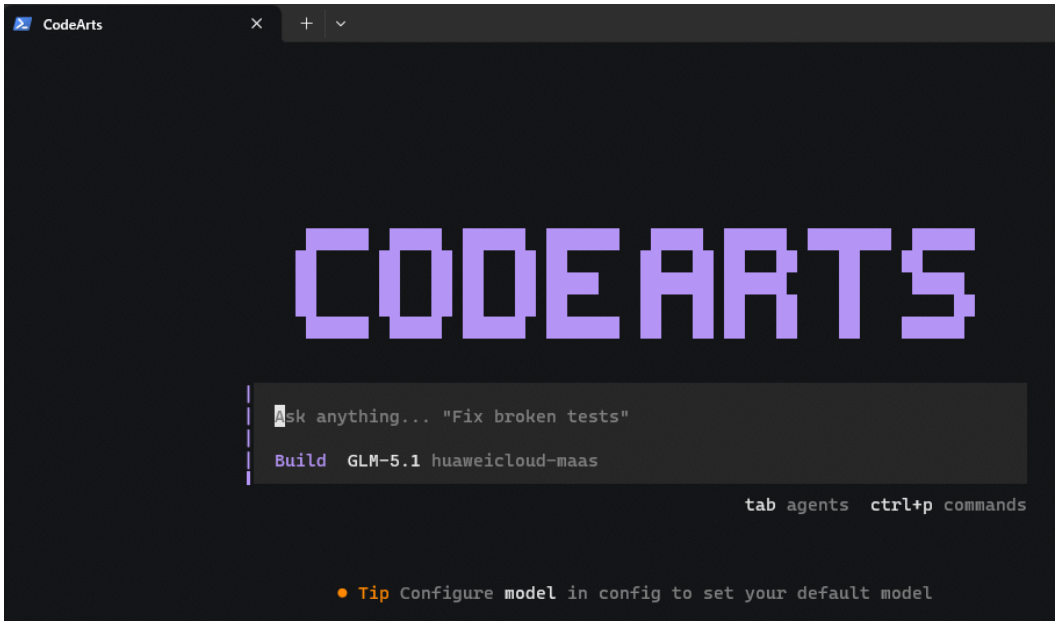
----结束

14 TUI

14.1 斜杠命令

您可在CMD、PowerShell或者终端，输入“codearts”并回车，进入TUI开发模式。

图 14-1 TUI 开发模式



在TUI开发模式下，您可以输入“/”后跟命令名称来快速执行操作，当前支持如下命令。

表 14-1 TUI 支持的斜杠命令

命令	说明
/help	显示帮助对话框 **说明** 弹出对话框后，单击“ok”、“esc”或者“Enter”，即可继续使用快捷键查看快捷键操作提示。
/mcps	展示所有MCP列表。您可以通过空格键切换MCP的“Enabled”与“Disabled”。 **说明** 如果您的MCP状态为“Disabled”，表示当前的MCP不能被LLM使用。
/compact	压缩当前会话
/exit	退出TUI，回到PowerShell或者CMD界面
/q	
/quit	
/logout	
/init	该命令会扫描您当前项目，了解项目的用途，并据此生成一个AGENTS.md **说明** 如果您当前项目无源代码、无配置、无构建文件等，TUI会建议您在有实际代码的项目中创建AGENTS.md，或者在您的工作盘下创建一个AGENTS.md
/models	查看并切换模型
/new	开始新的会话
/undo	执行该命令后将撤销对话中的最后一条消息，您可重新编辑任务内容，或直接取消任务 **注意** 所做的任何文件更改也将被还原
/redo	仅支持使用“/undo”后可用，表示重新执行撤销的消息 **注意** 所有文件更改也将被恢复
/sessions	列出会话并在会话之间切换。 如果您选择需要操作的会话后，可继续执行如下操作： ● “Ctrl+D”表示删除会话，二次确认后将删除选择的会话。 ● “Ctrl+R”表示重命名会话，修改名字后，按下Enter，完成会话重命名
/resume	
/continue	
/themes	列出可用主题

命令	说明
/thinking	切换对话推理过程的显示状态，默认开启。开启后可查看具备深度思考能力模型的完整推理细节。 **说明** 该命令仅控制思考块的显示或隐藏，不会开启/关闭模型本身的推理能力
/privacy	查看隐私政策
/docs	查看帮助文档
/lsp-toggle	切换或者查看lsp服务开关状态
/sdd-new	创建SDD需求规格
/sdd-design	创建SDD技术设计
/sdd-tasks	创建SDD编码任务
/sdd-apply	执行SDD编码实现
/run-mode	切换或者查看沙箱的运行模式
/bugfix	根据用户提交的问题单描述，自动调用修复专家智能体进行代码问题修复。
/schedule	创建定时任务。 详细介绍，请参见**在TUI中创建定时任务**。
/agents	查看并切换智能体。
/codebase/init	创建代码库索引。 首次使用时，在项目目录下运行此命令，即可新建代码库索引。
/codebase/detail	查询代码库索引进度
/codebase/update	增量更新代码库索引
/codebase/delete	删除代码库索引
/editor	打开文本编辑器
/review	审查代码变更
/skills	查看技能列表
/status	查看状态

14.2 快捷键

上一章节列举了一些斜杠命令的对应的快捷键。除此之外，TUI开发环境中还支持提示词输入快捷键，这些快捷键是内置功能，支持对对话框的输入信息进行便捷操作。

表 14-2 TUI 对话框提示词输入快捷键

快捷键	操作
Ctrl+A	移动到当前行的开头 **说明** 此快捷键仅适用于Windows、Linux和macOS操作系统。
Ctrl+E	移动到当前行的末尾
Ctrl+B	光标向左移动一个字符
Ctrl+F	光标向右移动一个字符 **说明** 此快捷键仅适用于Windows、Linux和macOS操作系统。
Windows/Linux：Alt+B macOS：Option+B	光标向左移动一个单词 **说明** 此快捷键仅适用于Windows、Linux和macOS操作系统。
Windows/Linux：Alt+F macOS：Option+F	光标向右移动一个单词 **说明** 此快捷键仅适用于Windows、Linux和macOS操作系统。
Ctrl+D	删除光标所在位置的字符
Ctrl+K	删除从光标到行尾的内容
Ctrl+U	删除从光标到行首的内容
Ctrl+W	删除前一个单词
Windows/Linux：Alt+D macOS：Option+D	删除后一个单词 **说明** 此快捷键仅适用于Windows、Linux和macOS操作系统。
Ctrl+G	跳转到第一轮会话

14.3 SDD

SDD（Specification-Driven Development）是一种规格驱动开发模式，严格通过“需求规格、技术设计、任务规划、编码实现”四个阶段推进工作。这种结构化流程旨在保障每一步开发均有据可依，实现过程可控与历史可查。

SDD 命令概述表

表 14-3 SDD 命令概述表

命令	输入说明	输出产物路径	功能说明
/sdd-new	/sdd-new + *需求描述*，必选参数	*{项目根路径}*/.codeartsdoer/specs/*产物文件夹名*/spec.md	根据需求描述生成spec.md文档，即定义“做什么”，如业务行为、需求约束等。
/sdd-design	/sdd-design + *产物文件夹名*，可选参数，系统会从对话上下文自动推断上一阶段的产物文件夹名 **注意** 必须存在目录（*{项目根路径}*/.codeartsdoer/specs/*产物文件夹名*）和生成的spec.md。	*{项目根路径}*/.codeartsdoer/specs/*产物文件夹名*/design.md	基于spec.md生成design.md文档，即定义“怎么做”，如技术架构、实现方案等。
/sdd-tasks	/sdd-tasks + *产物文件夹名*，可选参数，系统会从对话上下文自动推断上一阶段的产物文件夹名 **注意** 必须同时存在spec.md和design.md。	*{项目根路径}*/.codeartsdoer/specs/*产物文件夹名*/tasks.md	基于spec.md和design.md生成任务清单tasks.md
/sdd-apply	/sdd-apply + *产物文件夹名*，可选参数，系统会从对话上下文自动推断上一阶段的产物文件夹名，**直接在主智能体中执行** **注意** 必须同时存在spec.md、design.md和tasks.md。	代码 + 更新tasks	按tasks.md逐项执行编码任务

> **说明**
>
> *产物文件夹名*由模型自主生成。受生成随机性影响，即使输入相同的需求描述，输出的*产物文件夹名*也可能存在差异。

使用 SDD 生成代码

步骤1 进入TUI开发环境。

1. 打开目标项目的根目录。
2. 鼠标右键单击空白处，选择“在终端中打开”。
3. 在终端中输入“/codearts”并回车，即可进入TUI开发环境。

步骤2 **创建需求规格**。在TUI对话框中，输入“/sdd-new *需求描述*”并回车，生成spec.md文件。

图 14-2 创建需求规格

需求规格创建完成后，进入*{项目根路径}*/.codeartsdoer/specs/*产物文件名*目录下，可查看到生成的spec.md。如果生成的spec.md不符合您的预期，可以通过手动或智能体进行修改。

步骤3 **创建技术设计**。在TUI对话框中，输入“/sdd-design”并回车，基于spec.md生成design.md文档。

图 14-3 创建技术设计

技术文档设计完成后，进入*{项目根路径}*/.codeartsdoer/specs/*产物文件名*目录下，可查看到生成的design.md。如果生成的design.md不符合您的预期，可以通过手动或智能体进行修改。

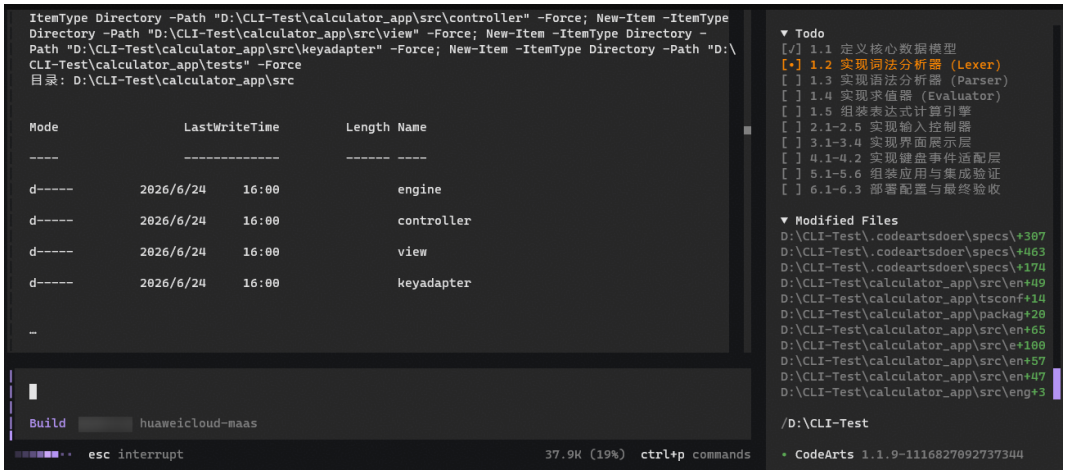
步骤4 **创建编码任务**。在TUI对话框中，输入“/sdd-tasks”并回车，基于spec.md和design.md生成任务清单tasks.md。

图 14-4 生成编码任务

编码任务规划文档设计完成后，进入*{项目根路径}*/.codeartsdoer/specs/*产物文件名*目录下，可查看到生成的tasks.md。如果生成的tasks.md不符合您的预期，可以通过手动或智能体进行修改。

步骤5 **执行编码实现**。在TUI对话框中，输入“/sdd-apply”并回车，智能体会按tasks.md逐项执行编码任务。

图 14-5 执行编码任务



----结束

15 CLI

15.1 命令

打开CMD、Powershell或者终端，执行命令“codearts [command]”，即可进入码道代码智能体的CLI开发模式。

表 15-1 CLI 命令表

命令	描述	子命令或子参数
codearts help	查看命令帮助	无
codearts completion	生成Shell自动补全脚本	无
codearts mcp	管理MCP服务器	详细子命令请参考**MCP命令**
codearts [project]	启动指定项目的TUI开发环境，其中[project]是可选参数，表示目标项目名称 例如，执行以下命令，进入TUI开发模式并加载Test项目： `codearts D:\Test`	无
codearts run [message]	非交互模式运行码道CLI，直接传入提示词。 例如，执行以下命令，读取当前目录下的文件： `codearts run 读取文件`	**codearts run**命令支持的参数列表请参考**表15-2**
codearts agent	管理智能体	• create，创建新智能体 • list，列出所有可用智能体

命令	描述	子命令或子参数
codearts models [provider]	列出可用模型 • 例如，执行如下命令，可获取所有供应商的可用模型 codearts models • 例如，执行如下命令，可获取指定供应商“huaweicloud-maas”的可用模型 codearts models huaweicloud-maas	无
codearts stats	显示Token使用量和成本统计信息	执行如下命令，查看所有统计数据 codearts stats 其余子命令请查看**表15-3**
codearts session	管理会话	• list，列出所有会话。 • delete <sessionID>。其中，执行**codearts session list**可查询sessionID，sessionID是必填参数。执行命令后，选中需删除的会话，按下回车即可
codearts export [sessionID]	以JSON格式打印出指定会话ID的数据。其中，执行**codearts session list**可查询sessionID • 例如，执行如下，选中需导出的会话，按下回车，即可打印出此会话的数据 codearts export • 例如，执行如下命令，即可打印出此sessionID的数据 codearts export [sessionID]	无
codearts import <file>	从本地JSON文件导入会话数据，<file>为必填参数 例如，执行如下命令，将把指定文件的数据导入，多一条sessionID数据 codearts import session.json 使用场景：生成这条sessionID数据后，您可以基于该ID继续会话	无
codearts upgrade	将码道CLI升级到最新版本	无
codearts uninstall	卸载码道CLI并删除所有相关文件	无

命令	描述	子命令或子参数
codearts docs	获取码道CLI的帮助文档链接	无
codearts privacy	查看码道CLI的隐私政策	无
codearts debug	调试与故障排查工具	涉及子命令请查看**表15-7**
codearts --version	打印版本号	无
codearts --print-logs	将日志输出到stderr	无
codearts --log-level <日志级别>	设置日志级别（DEBUG、INFO、WARN、ERROR）	无
codearts codebase	您可通过codearts codebase命令管理代码库索引	此命令支持的参数请查看**表9-3**

codearts serve	启动无界面的服务器。
	在执行serve相关命令前，请先在PowerShell、CMD或者终端执行如下命令，为服务器配置用户名和密码，否则将返回如下报错：“Error: CODEARTS_SERVER_PASSWORD is not set; please configure the password to start the server.”
	`export CODEARTS_SERVER_USERNAME="服务器登录账号"`
	`export CODEARTS_SERVER_PASSWORD="服务器登录密码"`
	其中，服务器登录账号与登录密码由用户自行设置。
		此命令支持的参数请查看**表15-10**

codearts attach <url>	将本地命令行绑定到远端码道CLI服务实例，使用用户名和密码登录，支持两种传账号密码的方式：环境变量、命令行--password参数。	此命令支持的参数列表请参考**表15-11**

表 15-2 codearts run 支持的参数列表

标志	描述

codearts run --command	• 运行命令，使用--command <命令>。
	例如，执行如下命令，完成初始化：
	`codearts run --command init`
	例如，执行如下命令，可运行test命令
	`codearts run --command test`
	• 使用SDD命令生成代码。格式为--command <SDD命令> <参数>。
	SDD命令支持以下四种取值：sdd-new、sdd-design、sdd-tasks和sdd-apply。参数为必填项，具体含义如下：
	– 当命令为sdd-new时，参数代表您需要执行的任务描述。
	– 当命令为sdd-design、sdd-tasks或sdd-apply时，参数代表产物文件夹名。
	在CLI中使用SDD命令时，无需在命令前添加斜杠“/”。其余功能与TUI界面保持一致。如需了解详细信息，请参见**SDD**。
	• 创建定时任务命令，使用--command schedule "提示词"，详细介绍请参见**在CLI中创建定时任务**。

codearts run --continue	继续上一个会话，可缩写为“-c” 例如，执行如下命令，继续上一个会话 `codearts run --continue`
codearts run "待执行指令" --sessionID <sessionID>	指定一个会话ID继续进行会话，可缩写为“-s” 例如，执行如下命令，在指定的sessionID中继续执行会话任务。 `codearts run "今天星期几" --sessionid ses_07885a8e7ffe8ESvH4zV1BdFJw` 其中，ses_07885a8e7ffe8ESvH4zV1BdFJw为会话ID，可通过**codearts session list**命令获取。
codearts run --fork	继续时分叉会话
codearts run --model	要使用的模型，可缩写为“-m”
codearts run --agent	要使用的代理
codearts run --file	附加到消息的文件，可缩写为“-f”
codearts run --format	格式：default或json
codearts run --title	会话标题
codearts run --thinking	显示思考块

标志	描述
codearts run --attach	用于连接到一个正在运行的服务器。 例如，执行如下命令，连接到正在运行的codearts服务器“http://localhost:4096”，执行问答： codearts run --attach *http://localhost:4096* --password *1***5 "hi"*
codearts run --auto	在当前会话中，以自动运行模式执行Bash工具。
codearts run --sandbox	在当前会话中，以沙箱运行模式执行Bash工具。

表 15-3 codearts stats 支持的参数列表

命令	描述
codearts stats --days	显示最近N天的统计信息 • 例如，执行如下命令，查看全部天数 codearts stats --days • 执行如下命令，查看近2天 codearts stats --days 2
codearts stats --tools	显示工具维度的统计数据
codearts stats --models	显示所有模型用量明细
codearts stats --project	显示项目的统计数据 进入需要查询的**Git仓**根目录下，右键选择“在终端中打开”，执行如下命令，查询当前项目的统计数据。 codearts stats --project **说明** 若目标代码仓无Git提交记录，系统会展示全部非Git仓项目数据。

表 15-4 codearts [project]支持标志

标志	简写	描述
--continue	-c	进入TUI开发模式，继续上一个会话 例如，执行如下命令，即可进入“D:\Test”项目下，进入TUI开发模式继续对话。 codearts D:\Test -c
--session	-s	要继续的sessionID。其中，sessionID通过执行“codearts session list”获取。 codearts D:\Test -s <sessionID>

标志	简写	描述
--fork	-	基于指定会话创建分叉会话。子命令如下： • **codearts --fork -c**，进入TUI开发模式，继续上一个会话 • **codearts --fork -s sessionID**，基于指定的会话ID，创建分叉会话
--model	-m	要使用的模型
--agent	-	要使用的代理

表 15-5 codearts session list 支持的参数

标志	简写	描述
--max-count	-n	限制为最近N个会话
--format	-	输出格式：table或json

表 15-6 codearts uninstall 命令标志

标志	简写	描述
--keep-config	-c	保留配置文件
--keep-data	-d	保留会话数据
--dry-run	-	显示将被删除的内容
--force	-f	跳过确认提示

表 15-7 codearts debug 子命令

命令	描述
codearts debug config	含义：显示解析后的完整配置 用途：查看当前生效的所有配置（包括合并了默认值、环境变量后的最终状态），用于排查配置加载问题
codearts debug rg	含义：ripgrep（代码搜索）调试工具，涉及子命令请查看表15-8 用途：专门调试ripgrep相关的行为，比如搜索规则、忽略文件、路径匹配等问题
codearts debug file	含义：文件系统调试工具，涉及子命令请查看表15-9 用途：用于调试文件读取/写入、路径解析、权限、符号链接等文件相关的问题

命令	描述
codearts debug scrap	含义：列出所有已知项目 用途：主要用于调试，帮助开发者了解码道CLI当前识别的项目
codearts debug skill	含义：列出所有可用技能 用途：输出当前环境中加载的所有Skill，包括它们的名称、描述、位置及内容
codearts debug agent <name>	含义：查看指定智能体的配置详情，<name>是必填参数 用途：输出某个智能体（Agent）的完整配置，包括权限、工具、模型参数、提示词等，用于调试Agent行为异常
codearts debug paths	含义：显示全局路径信息 用途：输出核心目录路径，包括数据目录、配置目录、缓存目录或者状态目录，用于排查文件找不到、权限问题
codearts debug wait	含义：无限等待（用于调试），如需退出，按快捷键（Windows（Ctrl+C）/Mac（⌘+C）） 用途：启动后不退出，保持进程运行，方便调试器附加到进程中进行调试

表 15-8 codearts debug rg 的子命令

命令	描述
codearts debug rg tree	含义：表示使用ripgrep展示文件目录树 用途： ● 查看ripgrep实际“看到”的文件结构，验证哪些目录/文件被包含或排除 ● 排查.gitignore、.ignore等规则是否生效，为什么某些文件搜不到 典型场景：怀疑项目文件被意外忽略，想确认ripgrep的扫描范围
codearts debug rg files	含义：使用ripgrep列出所有文件 用途： ● 列出ripgrep扫描范围内的所有文件（受忽略规则影响） ● 快速对比实际文件和ripgrep可见文件的差异，定位“文件未被索引”问题 典型场景：某个文件在码道CLI里搜不到，想确认是否被ripgrep忽略

命令	描述
codearts debug rg search <pattern>	含义：使用ripgrep搜索文件内容 用途： ● 直接调用ripgrep搜索文件内容，验证码道CLI搜索功能的底层行为 ● 对比码道CLI界面搜索结果和原生ripgrep结果是否一致，判断问题出在前端还是底层 典型场景：搜索结果缺失、匹配规则异常，用原生ripgrep做对比验证

表 15-9 codearts debug file 的子命令

命令	描述
codearts debug file read <path>	含义：以JSON格式读取并输出指定文件的内容 用途： ● 直接查看码道CLI实际读到的文件内容（尤其是配置文件、状态文件） ● 排查“文件明明存在，但码道CLI读取不到、读取内容为空或者读取格式不对的问题 ● 验证文件编码、JSON格式是否正确
codearts debug file status	含义：显示文件系统状态信息 用途： ● 查看当前目录或者工作区的文件状态、权限、是否存在异常（如符号链接、只读权限） ● 排查“文件权限不足”“路径不存在”等问题 ● 检查文件是否被系统锁定、占用
codearts debug file list <path>	含义：列出当前项目下指定目录下的所有文件和子目录 用途：主要用于列举项目下的目录下的文件和子目录
codearts debug file search <query>	含义：按条件模糊搜索当前项目下的文件 用途：用于快速查找跟条件相似的文件
codearts debug file tree [dir]	含义：显示指定的目录下非空目录的树结构 用途：以树形视图展示指定目录的完整结构，直观查看目录层级

表 15-10 codearts serve 支持的参数列表

参数	描述
--port	指定服务器监听端口，默认端口为4096 例如，执行如下命令，指定监听端口4096： `codearts serve --port 4096` 服务成功启动后，可打开新窗口，通过codearts attach命令连接服务器，详细操作指导可参考**表15-1**。
--hostname	服务器监听地址，默认值为127.0.0.1 例如，执行如下命令，指定服务器监听地址127.0.0.1： `codearts serve --hostname 127.0.0.1` 服务成功启动后，可打开新窗口，通过codearts attach命令连接服务器，详细操作指导可参考**表15-1**。
--mdns	启用mDNS局域网零配置服务发现，同一局域网内设备可自动扫描并访问当码道CLI，无需手动填写本机IP地址。 例如，执行如下命令，将在本机4096端口启动码道CLI，并启用mDNS局域网服务发现；同一局域网内设备可自动扫描当前服务，无需手动填写主机IP，通过.local域名即可访问Web管理页面与MCP接口。 `codearts serve --port 4096 --mdns`

--mdns-domain	此参数需要配合--mdns使用，表示自定义mDNS局域网主机名前缀，默认值为codearts.local：如果用户不写--mdns-domain，系统会自动使用codearts作为前缀，拼接mDNS固定后缀.local，完整局域网域名就是codearts.local。
	例如，执行如下命令，表示采用默认域名：codearts.local，访问地址为：http://codearts.local:4096
	`codearts serve --port 4096 --mdns`
	例如，执行如下命令，表示手动指定--mdns-domain，覆盖默认值，域名为：dev-server.local，访问地址为：http://dev-server.local:4096
	`codearts serve --port 4096 --mdns --mdns-domain dev-server`

--cors <origin>	表示给外部网页开权限，配置允许跨域访问的浏览器页面来源地址，支持多次配置多个来源。传入*表示放行所有网页，用于本地开发调试。未配置时仅服务自身页面可调用接口，外部网页会触发浏览器跨域拦截。
	例如，执行如下命令，表示运行在http://localhost:5173的本地前端页面，可以正常请求4096端口的接口
	`codearts serve --port 4096 --cors http://localhost:5173`
	例如，执行如下命令，允许访问多个来源：
	`codearts serve --port 4096 \`
	`--cors http://localhost:5173 \`
	`--cors http://127.0.0.1:3000`
	例如，执行如下命令，表示允许所有源，表示仅本地调试用，**生产环境不建议，存在安全风险**。
	`codearts serve --port 4096 --cors *`

表 15-11 codearts attach 支持的参数列表

参数	描述
<url>	执行该命令，可以连接到运行的服务端。 您可以按照如下步骤连接正在运行的服务器： 1. 使用该命令连接服务器前，您需要配置临时环境变量，指定连接服务器的用户名和密码： ● Windows系统的配置命令如下： set CODEARTS_SERVER_USERNAME=服务器用户名 set CODEARTS_SERVER_PASSWORD=服务器密码 ● macOS或者Linux系统的配置命令如下： export CODEARTS_SERVER_USERNAME="服务器用户名" export CODEARTS_SERVER_PASSWORD="服务器密码" 2. 执行如下命令，在一台终端启动服务： codearts serve 出现如下回显，表示成功启动服务。 本服务为本地开发环境，非标准云服务，不提供企业级安全加固与 SLA 保障。 请勿将服务地址暴露至公网或分享给他人，可能存在未授权访问、代码泄露等风险。 继续使用即视为您已了解并自愿承担上述风险。 建议通过配置反向代理等安全访问方案，配置方式可参考： https://support.huaweicloud.com/codeartsagent_faq/codeartsagent_faq_0054.html codearts server listening on http://127.0.0.1:4096 3. 在一台终端启动服务后，您可以执行如下命令，在另一台终端连接指定服务器： codearts attach ${CODEARTS_SERVER_URL} --password ${CODEARTS_SERVER_PASSWORD} 其中，${CODEARTS_SERVER_URL}为要连接的服务器地址，${CODEARTS_SERVER_PASSWORD}为服务器的密码。 例如，执行如下命令，连接本机4096端口的远程开发服务器，并通过密码完成鉴权接入开发环境。 codearts attach http://localhost:4096 -p yourpassword
-p，--password	服务端密码
-c，--continue	继续上次会话
-s，--session	指定会话ID
--fork	分叉会话，与“--continue”或“--session”配合使用

15.2 MCP 命令

codearts mcp命令用于管理MCP服务器，支持添加、列出、认证和调试MCP服务器。

codearts mcp add

执行该命令可新增一个MCP服务器。

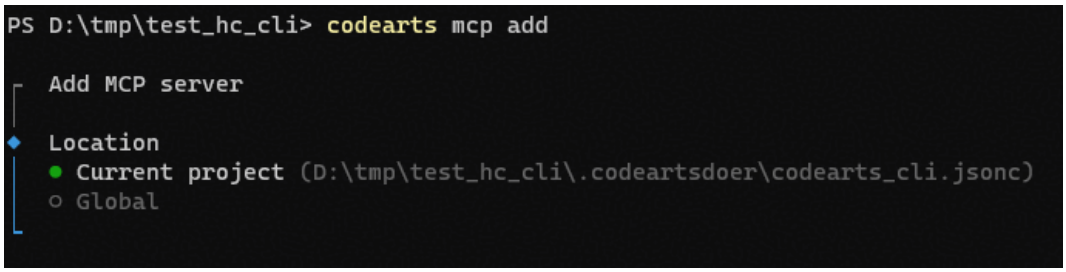
本示例以Windows系统的项目级配置为例：

步骤1 进入“D:/tmp/test_hc_cli”鼠标右键，选择“在终端中打开”，在打开的PowerShell界面上输入如下命令，根据提示进行选择。

```
codearts mcp add
```

步骤2 按提示配置“Location”参数，本示例选择“Location”。

图 15-1 配置“Location”参数



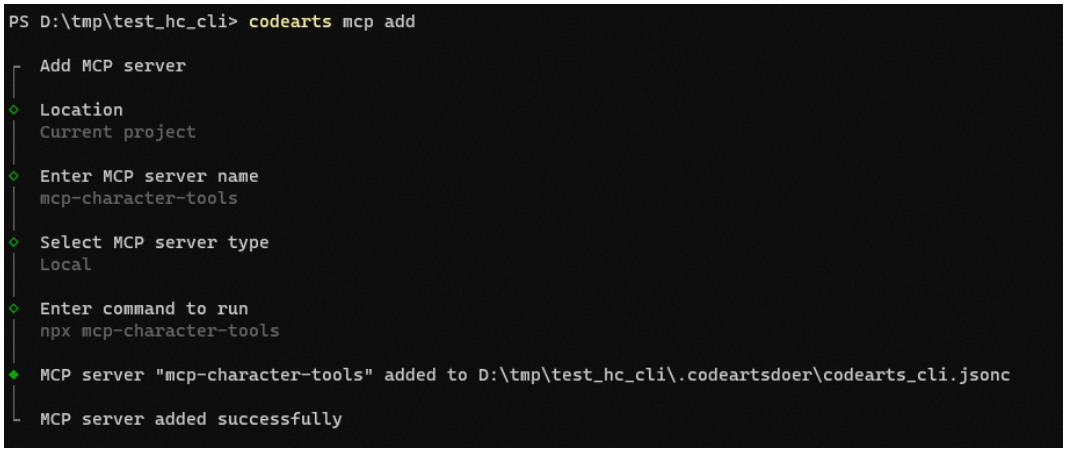
Location参数的填写指导请参考表15-12。

表 15-12 Location 参数说明

参数	解释
Current project	将MCP配置到当前项目下：项目根目录/.codeartsdoer/codearts_cli.jsonc
Global	将MCP配置到用户路径下：~/.codeartsdoer/codearts_cli.jsonc

步骤3 继续根据提示配置参数“Enter MCP server name”、“Select MCP server type”、“Enter command to run”。

图 15-2 配置示例



各参数的配置详情请参考表15-13。

表 15-13 其他 MCP 服务器配置参数

参数	解释
Enter MCP server name	输入自定义的MCP服务器名称
Select MCP server type	选择MCP服务器的运行模式： • Local，表示这个服务器通过本地执行一条命令启动，例如（npx、uvx或者python等），通信方式是stdio • Remote，表示配置远程MCP服务器，后续需要输入服务器的URL，码道CLI将通过网络（HTTP/SSE）连接
Enter command to run	运行此MCP服务器的命令

步骤4 如果您需要确认当前配置的MCP服务器是否可用，请执行**codearts mcp list**查看。

----结束

codearts mcp list

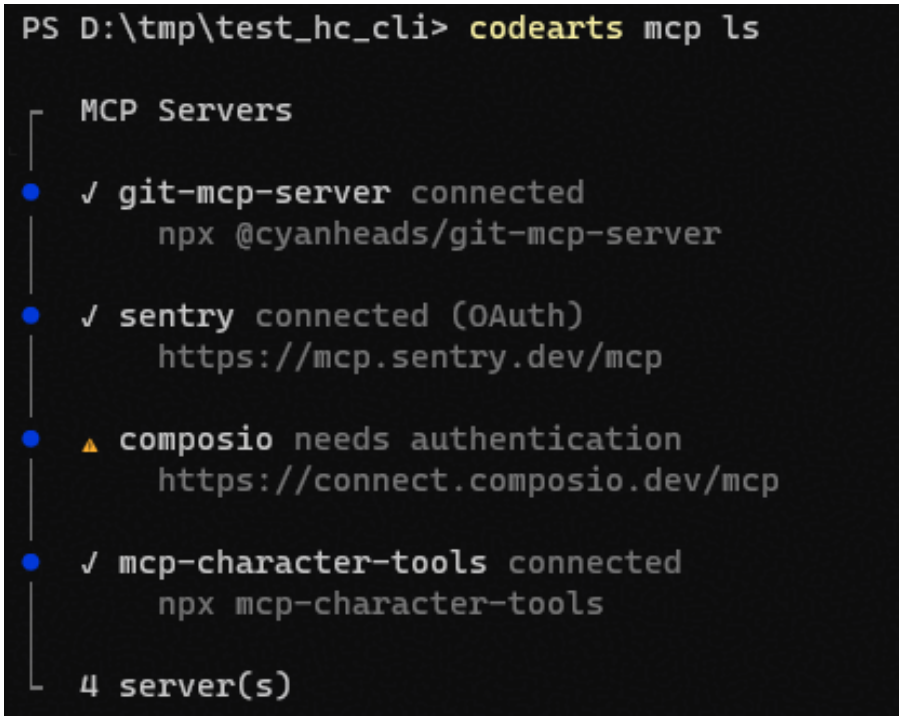
执行该命令将列出所有已配置的MCP服务器及其连接状态。

例如，执行如下命令，查看项目目录“D:/tmp/test_hc_cli”下已配置的MCP服务器及其连接状态。

```
codearts mcp list
```

当前目录下共4个MCP服务器，各MCP服务器状态为：git-mcp-server、sentry、mcp-character-tools已连接，composio待认证。

图 15-3 查询示例



MCP服务器的状态解释，请参考下表：

表 15-14 MCP 服务器状态

图标	状态
✓	已连接
□	待认证
○	未初始化或者已禁用
✗	失败

codearts mcp auth list

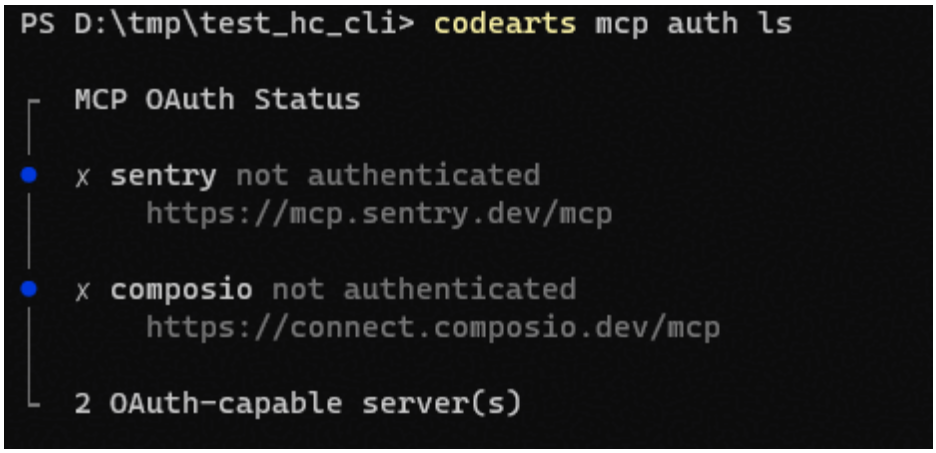
执行该命令，将列出支持认证的MCP服务器及其认证状态。

例如，在项目目录“D:/tmp/test_hc_cli”执行该命令，将列出该目录下支持认证的MCP服务器及状态。

```
codearts mcp auth list
```

例如下图所示，查询到当前项目下有2个MCP服务器待验证。

图 15-4 待认证的服务器



codearts mcp auth [name]

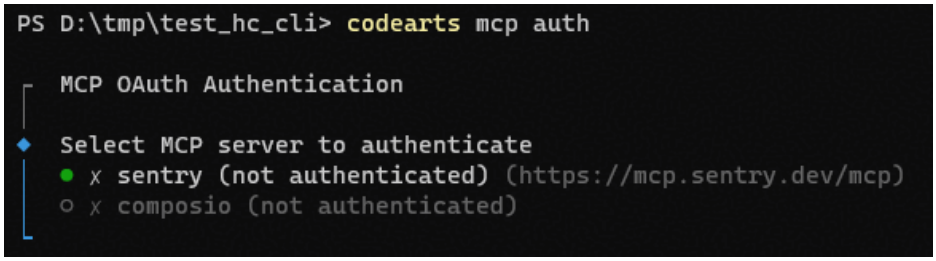
执行该命令，可对MCP服务器进行认证。

步骤1 执行如下命令，将列出没有认证过的MCP服务器，请根据回显选择需要认证的MCP服务器，回车即可启动认证。

```
codearts mcp auth
```

例如下图所示，查询到当前项目下有2个MCP服务器待验证，选择需要认证的MCP服务器，回车即可启动认证。

图 15-5 待验证的 MCP 服务器



您可参考如下示例配置进行认证：

```
"mcp": {
   "sentry": {
     "type": "remote",
     "url": "https://mcp.sentry.dev/mcp",
     "oauth": {}
   }
 }
```

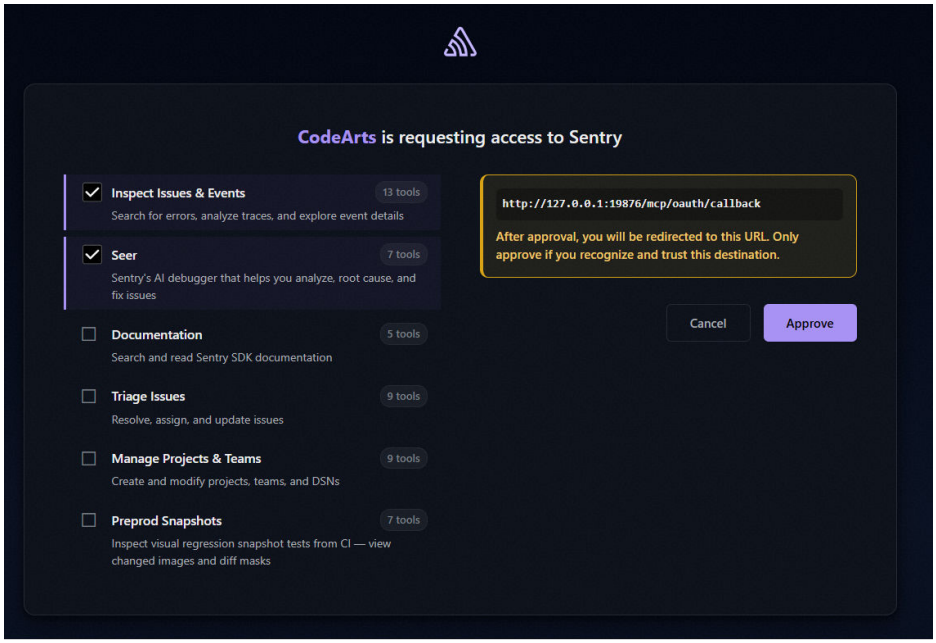
步骤2 配置完成后，执行如下命令，可打开浏览器窗口完成认证，将码道CLI连接到你的Sentry账户，命令中的“sentry”为上述步骤中配置的MCP名称。

认证完成后，“~/.codeartsdoer/cli-data/mcp-auth.json”文件会增加Sentry配置。

```
codearts mcp auth sentry
```

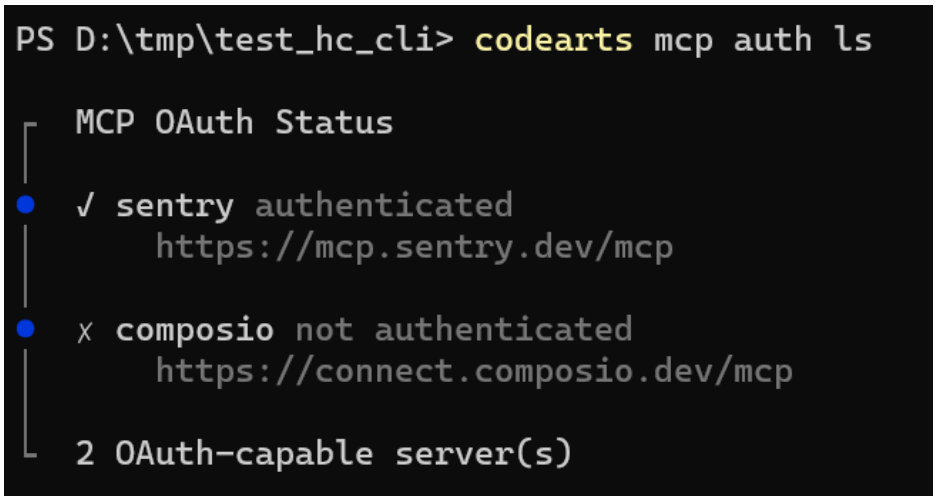
如下图所示，完成Sentry账户认证。

图 15-6 Sentry 账户认证



步骤3 授权成功后，再次执行**codearts mcp auth ls**，可以查看“sentry”已经被授权。

图 15-7 授权成功



----结束

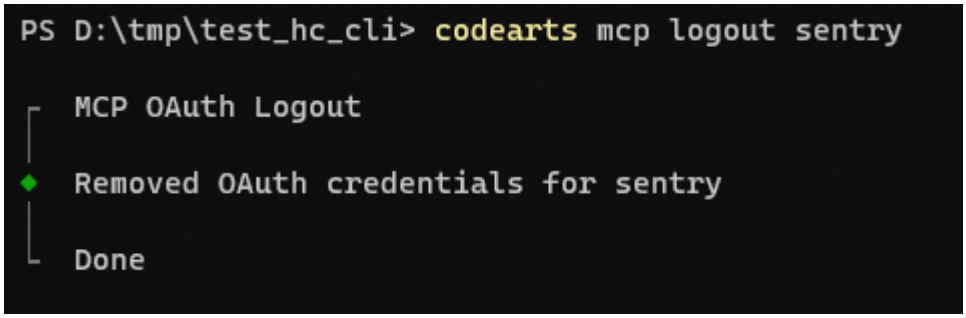
codearts mcp logout [name]

执行该命令，将移除MCP服务器的认证凭据。

例如，进入项目目录“D:/tmp/test_hc_cli”，执行如下命令，将移除该项目下的认证凭据。

此时，“~/.codeartsdoer/cli-data/mcp-auth.json”文件中对应的MCP配置被删除。

图 15-8 删除认证凭据



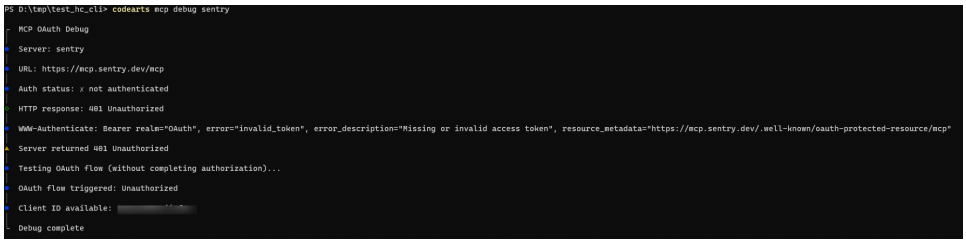
codearts mcp debug <name>

执行该命令，将显示认证信息和连接测试结果。

示例：

图示为调试OAuth登出的MCP服务器。

图 15-9 调试图



图示为调试OAuth授权成功的MCP服务器。

图 15-10 调试图