

安装Easyscan_v2.0

Easyscan_v2.0 被封装为可安装的 Python 软件包，可以通过 GitHub 仓库执行以下命令完成安装：

```
python3 -m pip install git+https://github.com/phyzhangyang/EasyScan_HEP.git
```

也可下载解压后直接在本地源码目录执行本地安装：

```
cd EasyScan_HEP-master/  
python3 -m pip install .
```

安装完成后，在任意工作目录下都可执行以下命令进入UI界面：

```
easyscan --ui
```

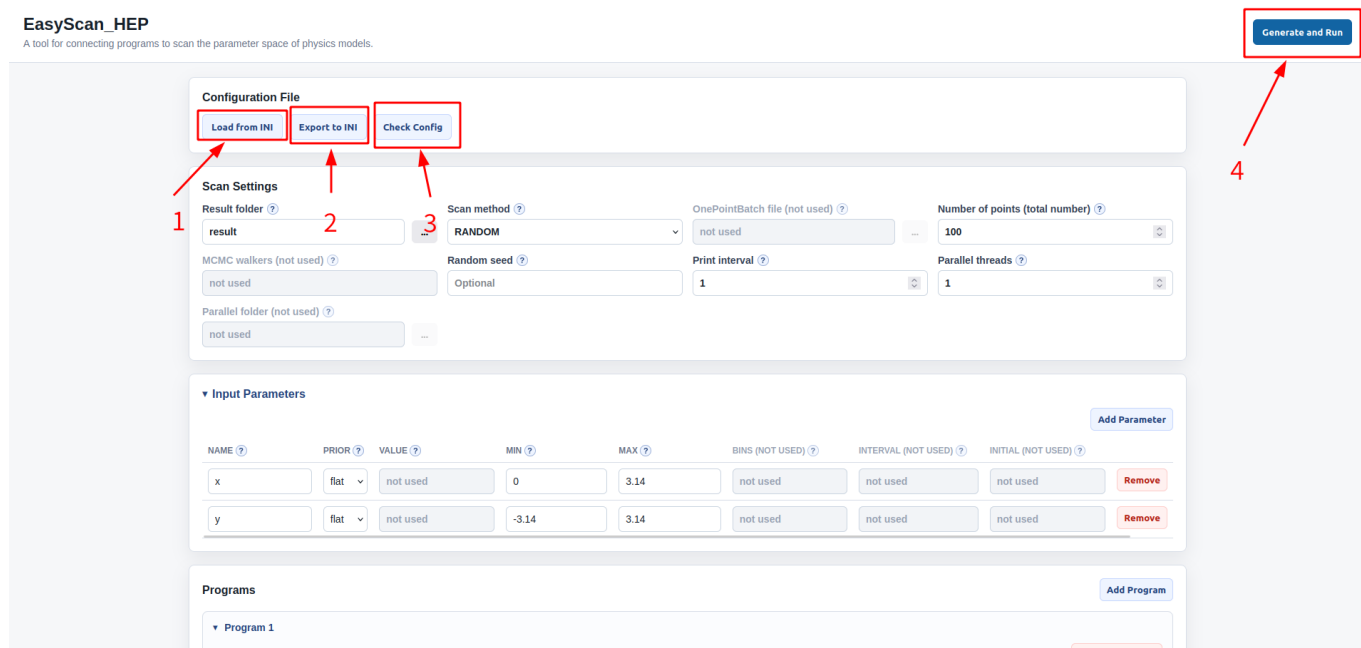
第一种安装方式：1、pip 在后台自动调用 git，临时把 GitHub 上最新的源码完整克隆到系统临时目录。2、读取源码里的 pyproject.toml（安装配置文件，相当于“安装说明书”），自动下载依赖、打包。3、把程序安装到当前 Python 环境，注册全局 easyscan 命令。4、安装完成后自动删除临时下载的源码文件。

第二种安装方式：在 Linux 终端的路径规则里，“.”就代表「当前终端所在的目录」。所以这条命令翻译过来就是：用 pip 安装「当前文件夹」里的这个 Python 包。背后的执行逻辑：1、pip 扫描当前目录，找到 pyproject.toml（旧项目是 setup.py）—— 这个文件定义了包名、版本、依赖、easyscan 命令入口等所有安装规则。2、按照配置文件，把当前文件夹里的代码打包、安装到 Python 系统目录。3、注册全局 easyscan 命令，和远程安装的效果完全一样。

Web UI界面操作

我们搭建一个测试文件夹，新建一个文件夹，命名为0000，放入以下文件：[/EasyScan_HEP-master/templates/example_random.ini](#)和[/EasyScan_HEP-master/utils](#)。

`easyscan --ui` 进入UI界面

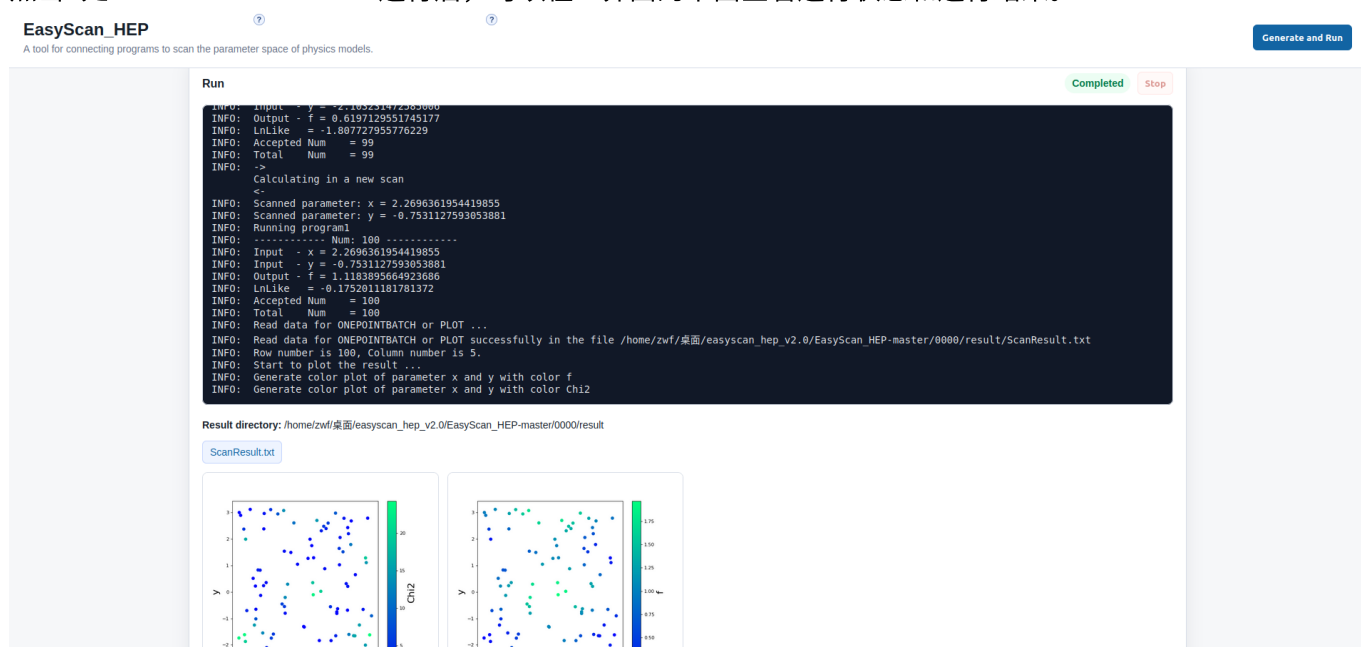


在UI界面中，1处是加载配置文件（.ini文件）；2处是将修改好的配置文件导出；3处是对修改的配置参数进行验证检查（看看有没有漏项 有没有语法错误）；4处是点击运行扫描程序。

也可以使用以下命令直接在UI界面中加载配置文件。

```
easyscan --ui example_random.ini
```

点击4处Generate and Run运行后，可以在UI界面的下面查看运行状态和运行结果。



使用easyscan_skills的两种方法

方法一：直接使用袁兴博老师推荐的 Claude Code加载easyscan_skills

从github上下载skills

```
mkdir -p ~/.codex/skills
git clone https://github.com/Contract-Mediated-Agent/easyscan-skill.git
~/.codex/skills/easyscan-hep
```

或者将 `files/easyscan-skill-main.zip` 解压到自己的工作目录例如 `work/.claude/skills`。

在 `claude code` 中启动 `easyscan_skills`: `/easyscan_hep` (可以使用 `table` 键补全)。

在对话框中输入以下内容进行测试: `Scan x from 0 to 3.14 and y from -3.14 to 3.14, randomly with 100 total points. Put results in example_random. Run ./TestFunction.py in utils/. Write x and y to utils/TestFunction_input.dat at row 1 columns 1 and 2. Read f from utils/TestFunction_output.dat at row 1 column 2. Add Gaussian f = 1 +/- 0.2. Make Color plots for x, y, f and x, y, Chi2.`

或

在 `x` 取值范围 `0` 到 `3.14`、`y` 取值范围 `-3.14` 到 `3.14` 的区域内随机采样, 共生成 `100` 个数据点。对每个点计算其到坐标原点 `(0, 0)` 的欧几里得距离, 将所有结果保存至名为 `distance_data.csv` 的 CSV 文件中。最后生成散点图, 对这些点的分布情况进行可视化展示。

方法二: 安装Codex, 替换为第三方API, 加载easyscan_skills

Codex CLI 基于 Node.js 开发, 要求 Node.js ≥ 18.0 。这里使用 `nvm` 管理 Node.js 版本, 避免污染系统环境。为此首先安装 `nvm`, 然后安装 Node.js

1、安装 nvm

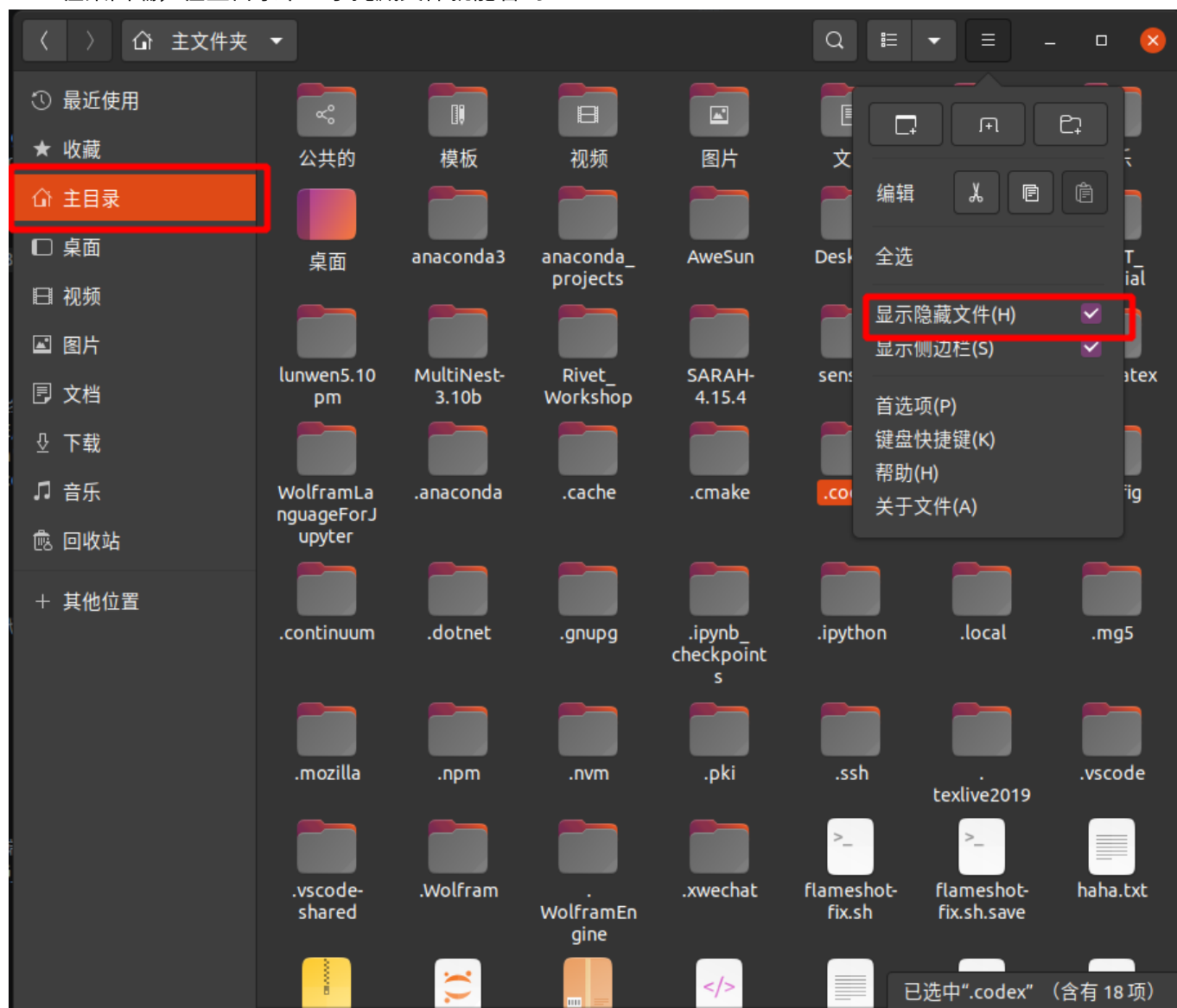
(1) 解压

```
cd files/
# 进入下载的aiagent文件夹下

tar -zxvf ./nvm-0.39.7.tar.gz -C ~/.
# 将nvm-0.39.7.tar.gz解压到home目录下。
#参数 -C 指定解压到的目录; ~/. 表示home目录

mv ~/.nvm-0.39.7 ~/.nvm
# 将解压到home目录下的 nvm-0.39.7 文件夹重命名为 .nvm
```

PS：在桌面端，在主目录下显示隐藏文件就能看到 .nvm



(2) 配置环境变量

```
# 写入 .bashrc 配置，直接在终端中运行就行
echo 'export NVM_DIR="$HOME/.nvm"' >> ~/.bashrc
echo '[ -s "$NVM_DIR/nvm.sh" ] && source "$NVM_DIR/nvm.sh"' >> ~/.bashrc
echo '[ -s "$NVM_DIR/bash_completion" ] && source "$NVM_DIR/bash_completion"' >> ~/.bashrc

# 加载生效
source ~/.bashrc

# 验证，会输出版本号 0.39.7 代表生效
nvm --version
```

(3) nvm 换源

定义一个 nvm 专属环境变量，告诉 nvm：下载 Node.js 二进制安装包时，不要去国外官方源 nodejs.org，改用清华源。

```
# 添加清华源到bashrc , 直接在终端中运行
echo 'export
NVM_NODEJS_ORG_MIRROR=https://mirrors.tuna.tsinghua.edu.cn/nodejs-release/'
>> ~/.bashrc

# 加载生效
source ~/.bashrc

# 验证 , 输出https://mirrors.tuna.tsinghua.edu.cn/nodejs-release/ 即代表生效。
echo $NVM_NODEJS_ORG_MIRROR
```

2、安装 Node.js

Codex CLI 要求 Node.js >=18.0, 这里我们下载20版本

```
#安装 20 系列最新 LTS (Node20 代号 iron)
nvm install lts/iron

# 设为系统默认版本, 新开终端自动生效
nvm alias default 20

# 当前终端立即切换到 20 LTS
nvm use 20

#验证安装结果。
#node -v: v20.19.2
node -v
#npm -v: 10.8.2
npm -v
```

3、安装 codex

建议使用手机热点进行安装 (确保网络畅通)

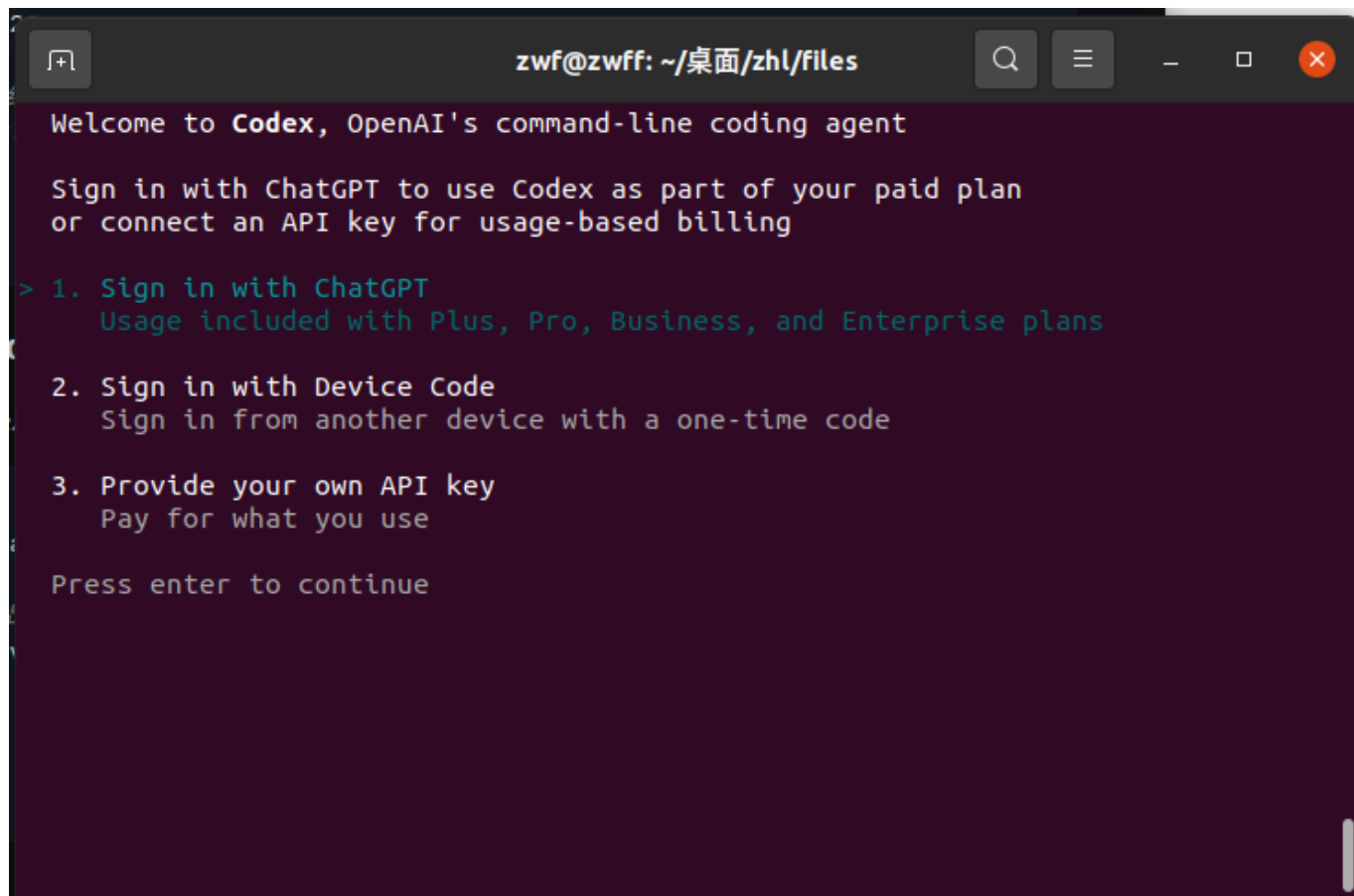
```
#不要使用 sudo npm install -g @openai/codex , 因为在sudo下不是Node.js 20的环境
npm install -g @openai/codex

#验证 , 输出版本号 codex-cli 0.143.0
codex --version

#启动
codex

#退出
Ctrl + Z
```

codex 安装完成后的启动界面：



```
zwf@zwff: ~/桌面/zhl/files
Welcome to Codex, OpenAI's command-line coding agent

Sign in with ChatGPT to use Codex as part of your paid plan
or connect an API key for usage-based billing

> 1. Sign in with ChatGPT
   Usage included with Plus, Pro, Business, and Enterprise plans

2. Sign in with Device Code
   Sign in from another device with a one-time code

3. Provide your own API key
   Pay for what you use

Press enter to continue
```

3.5、开通第三方模型API接入

这是修改codex配置文件接入第三方模型的前置准备工作，我们以火山方舟为例（新人有免费的50万tokens，用来连接测试足够）

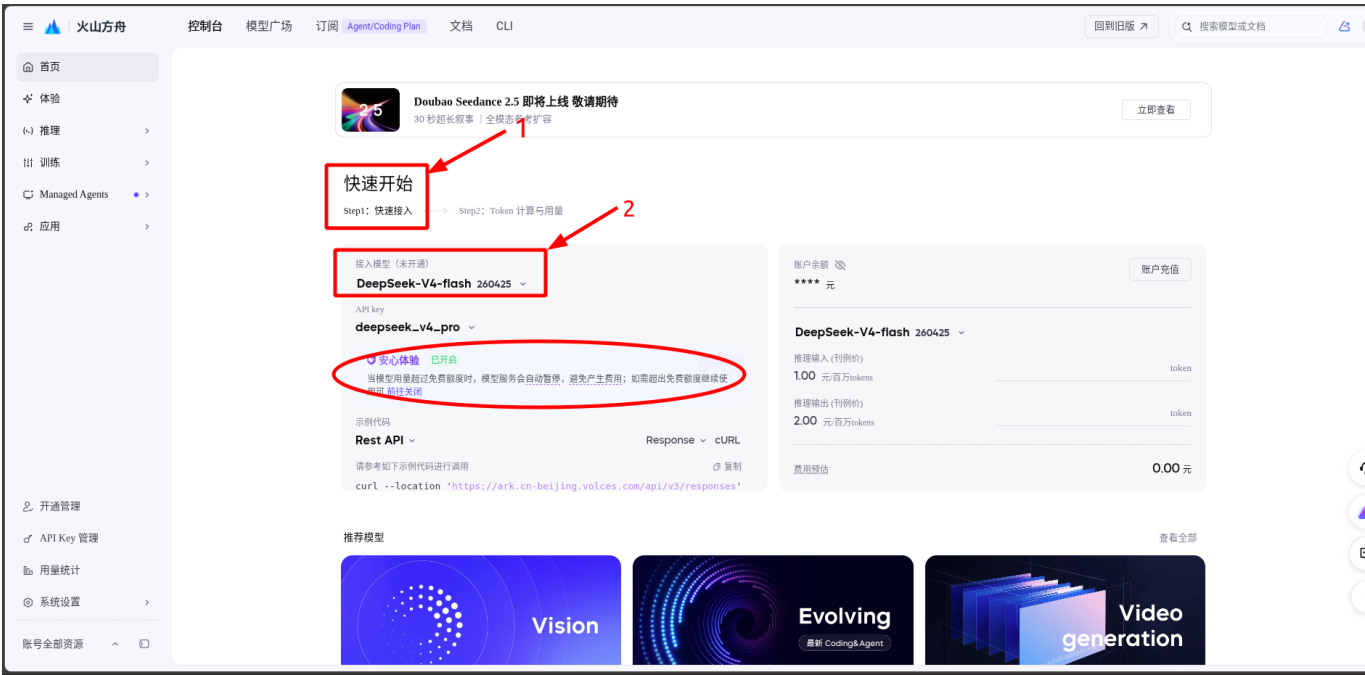
<https://www.volcengine.com/product/ark> 进入火山引擎，注册登陆。

（1）开通模型

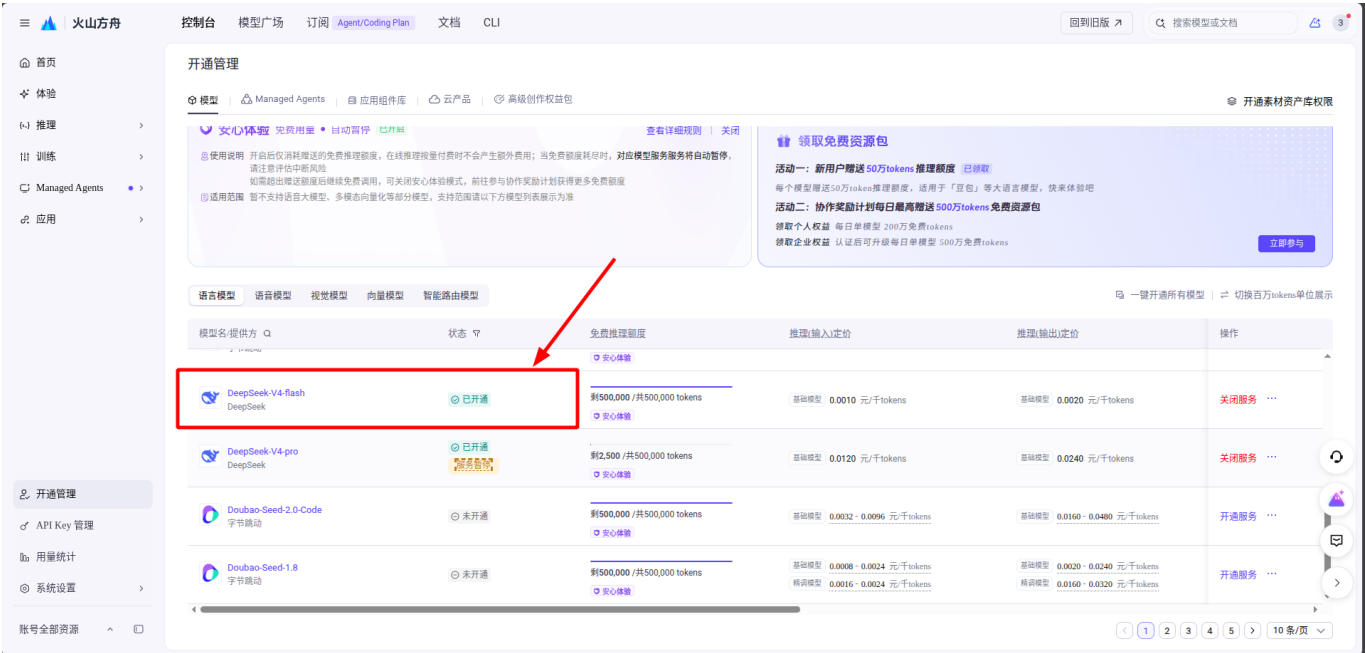
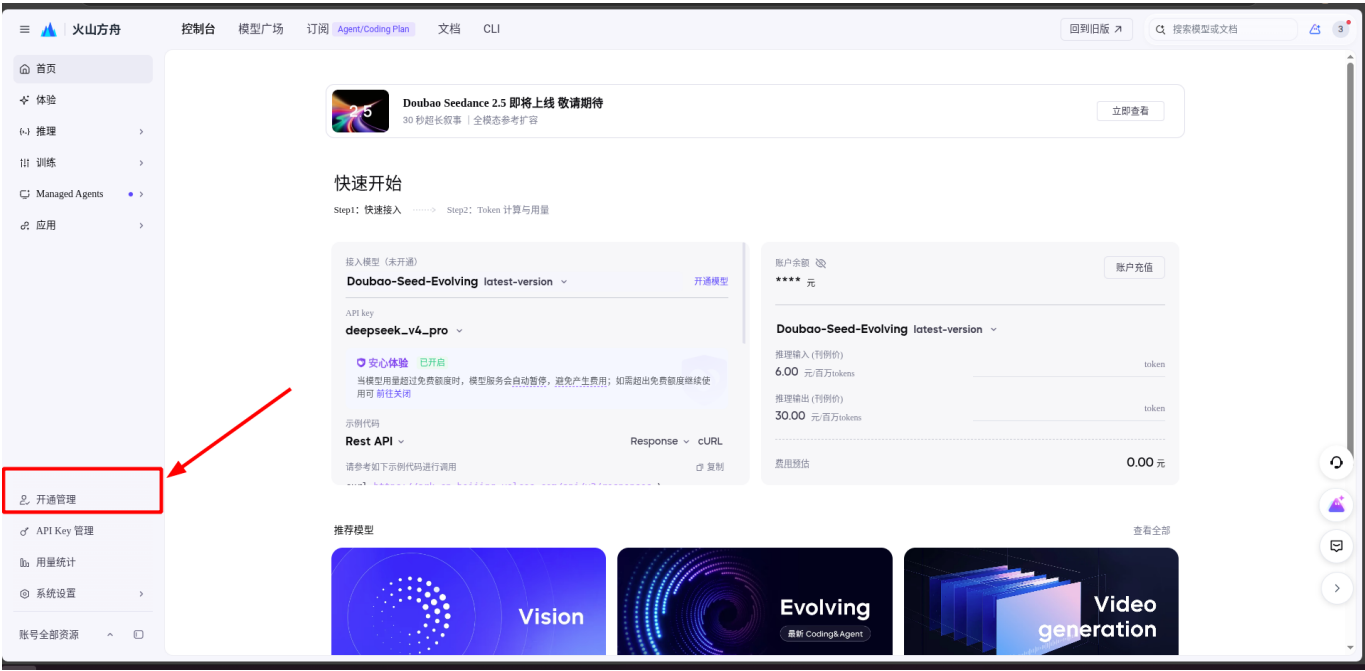
点击进入“控制台”



进入控制台后，点击1处的“快速开始”；然后在2处选择要开通的模型（每个模型都有免费的50万tokens额度；建议开通椭圆处的“安心体验”，防止免费额度用完之后继续消耗tokens）

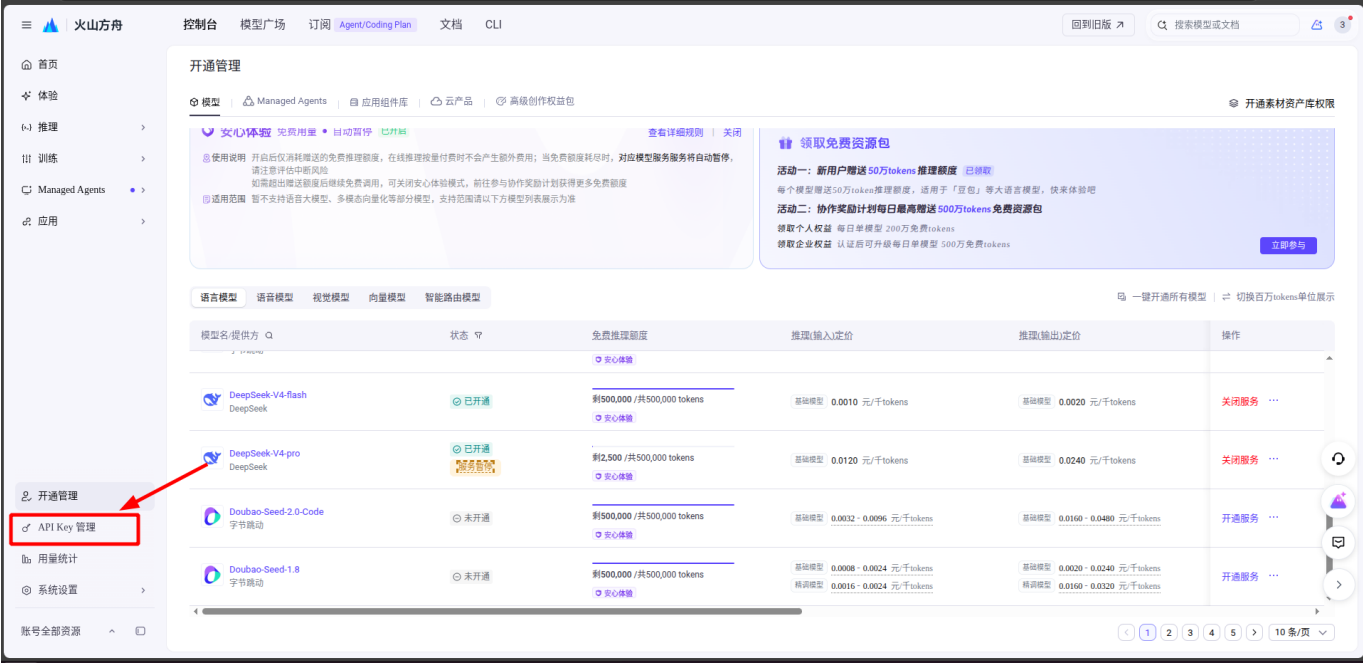


例如选择开通 deepseek-v4-flash ，开通后进入“开通管理”查看

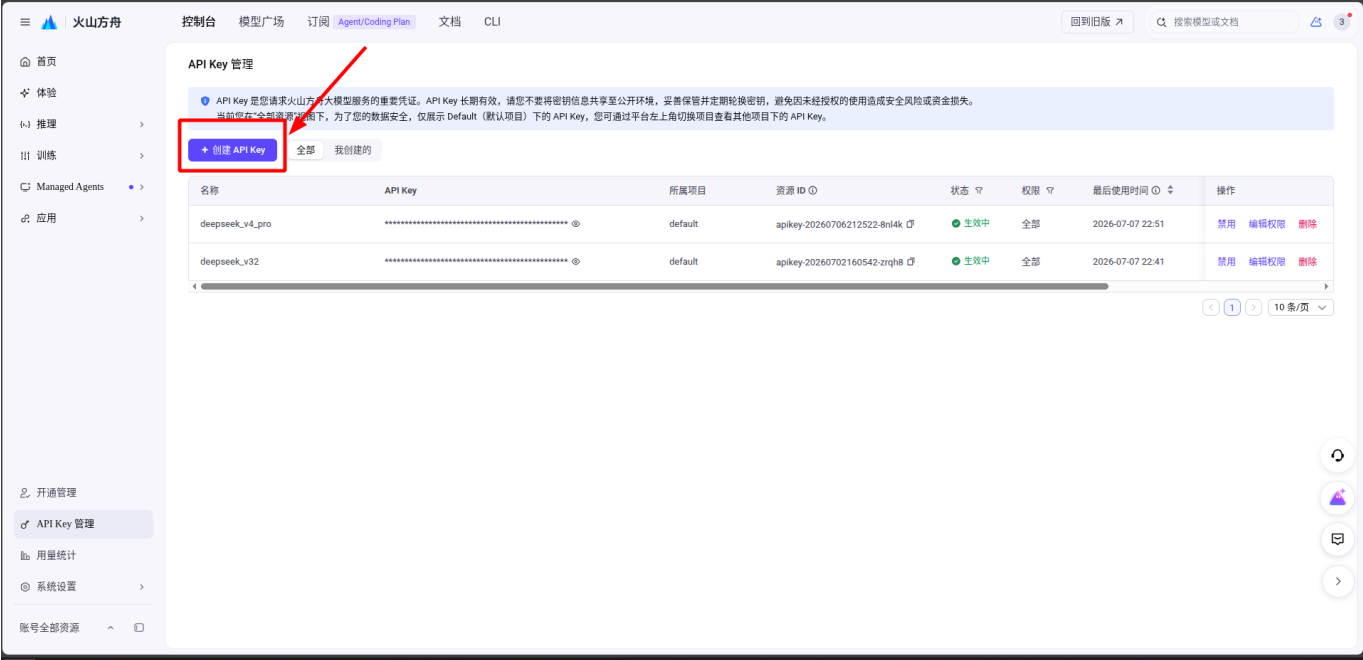


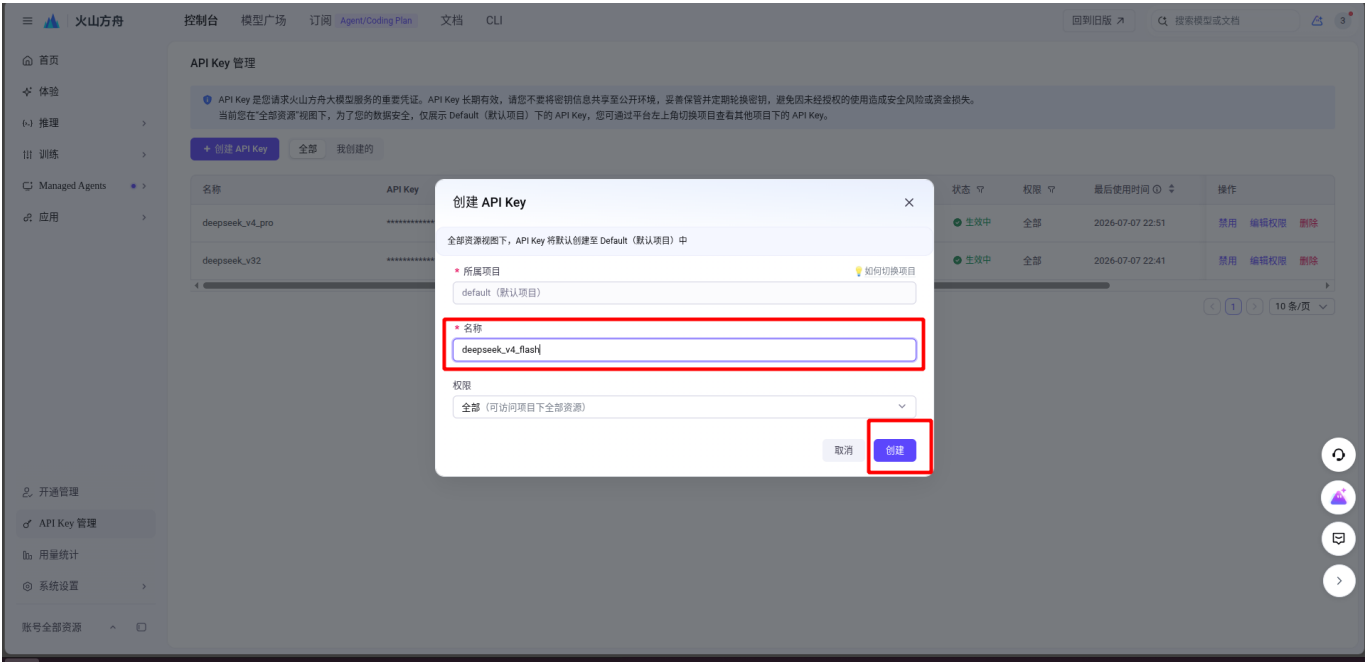
(2) 开通API Key

点击左侧的“API Key管理”

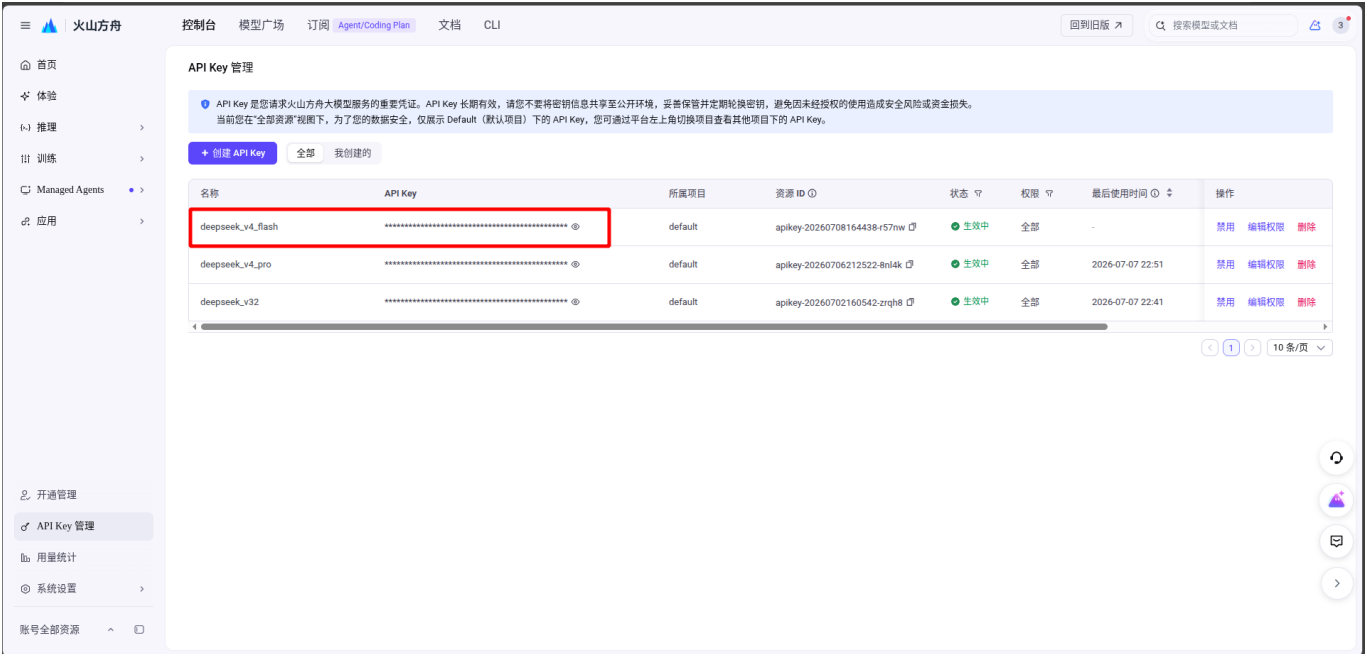


点击“创建API Key”



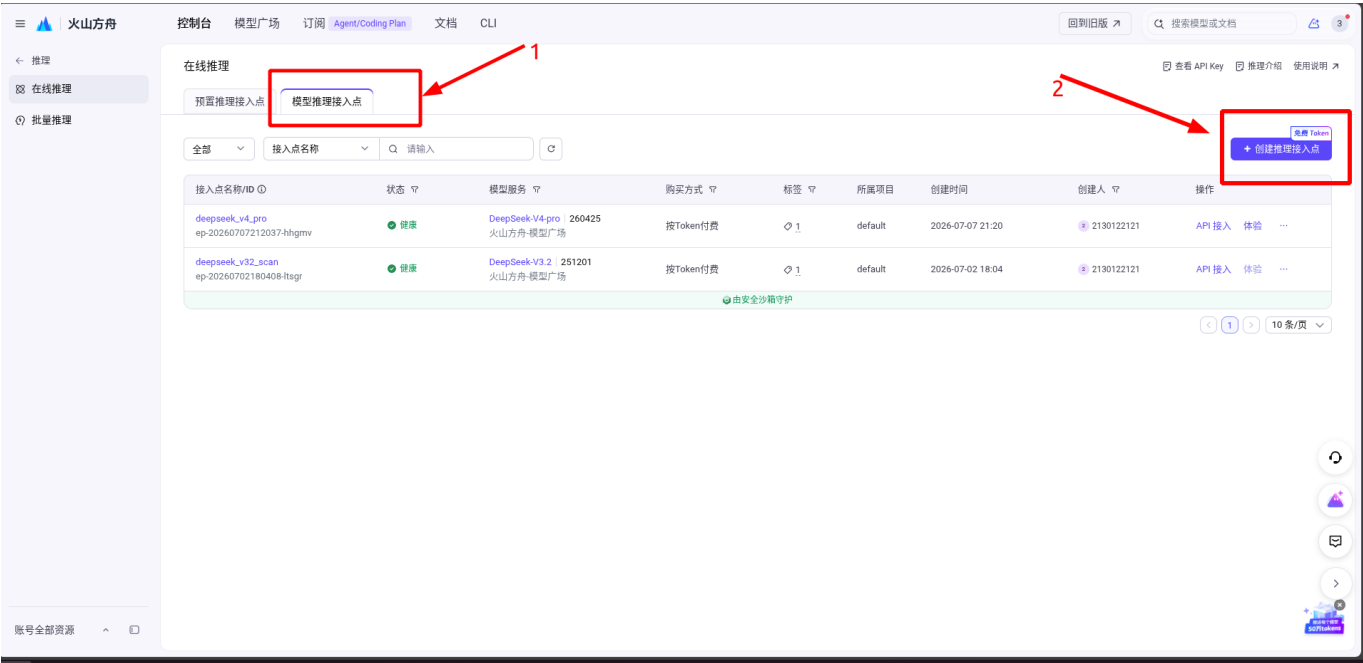
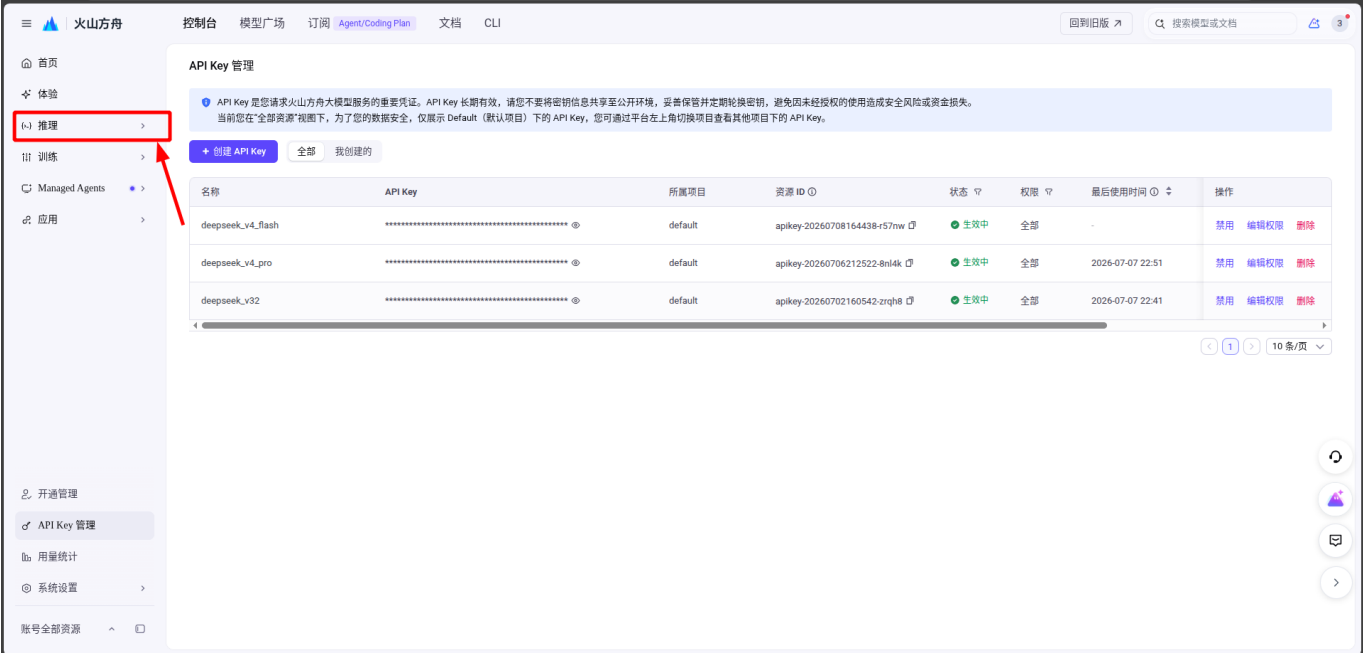


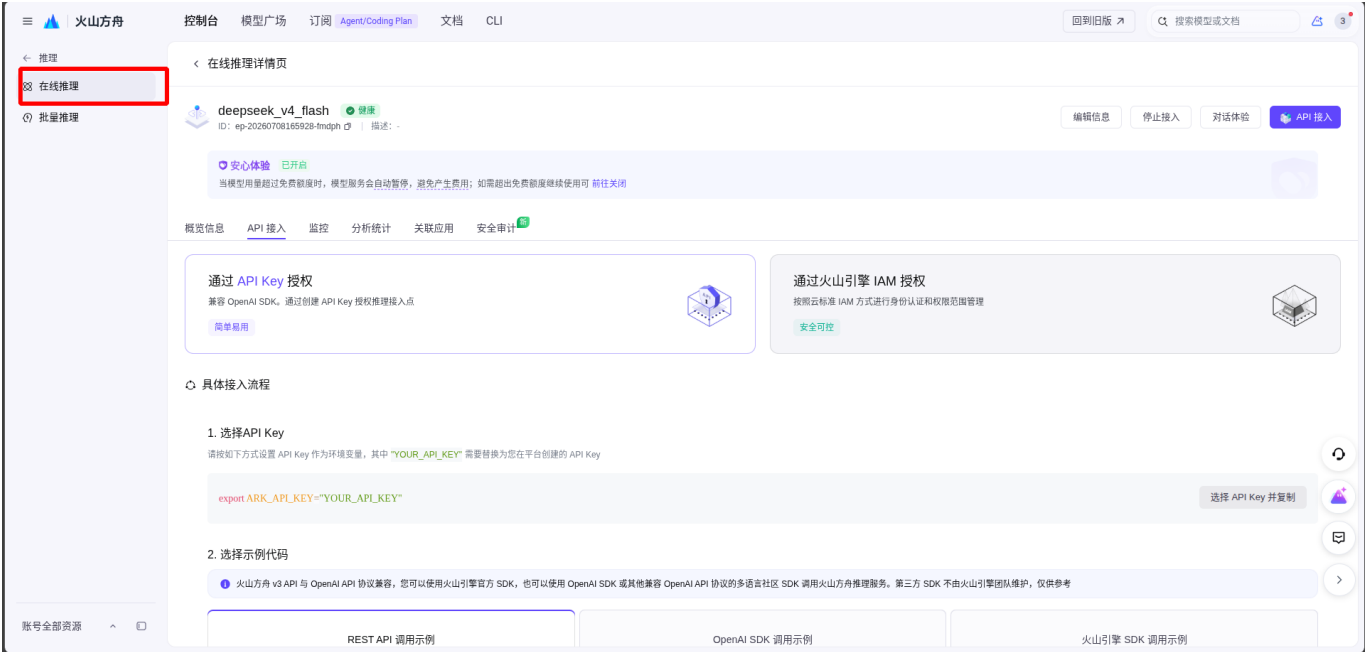
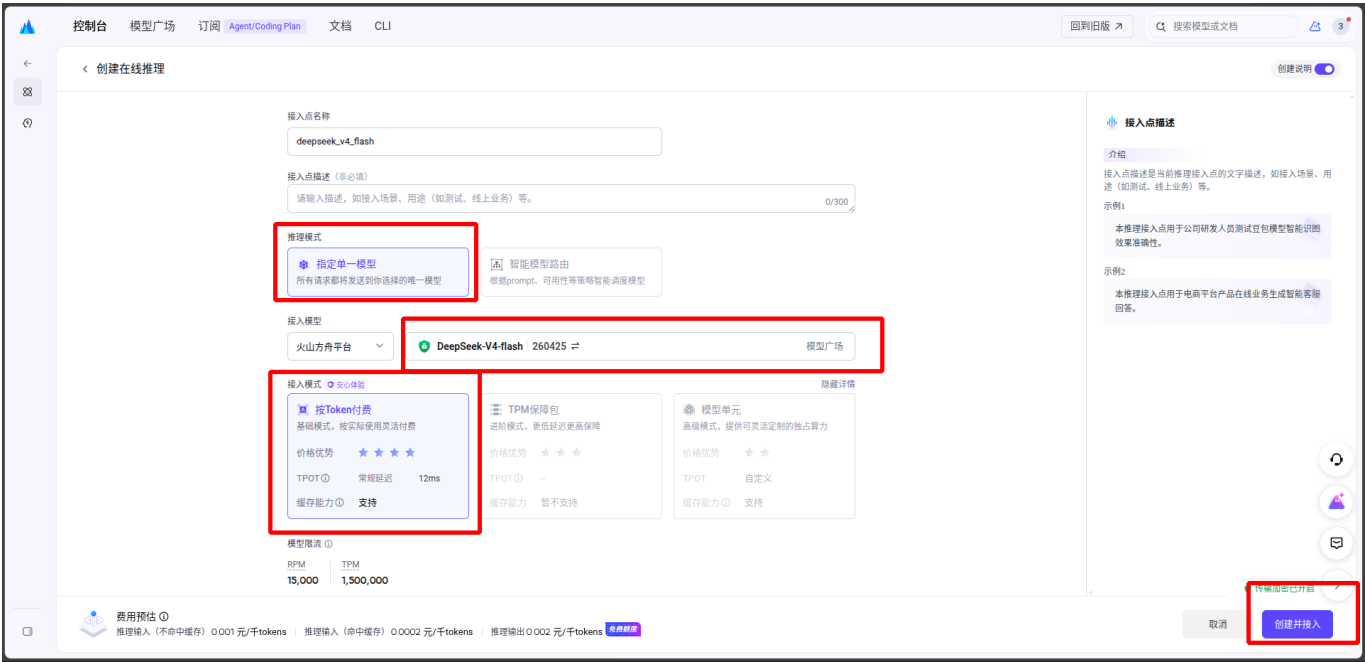
最后创建好的API Key接入点，红框里面隐藏的数字 类似 `ark-xxxxxxxxxxxxxx`



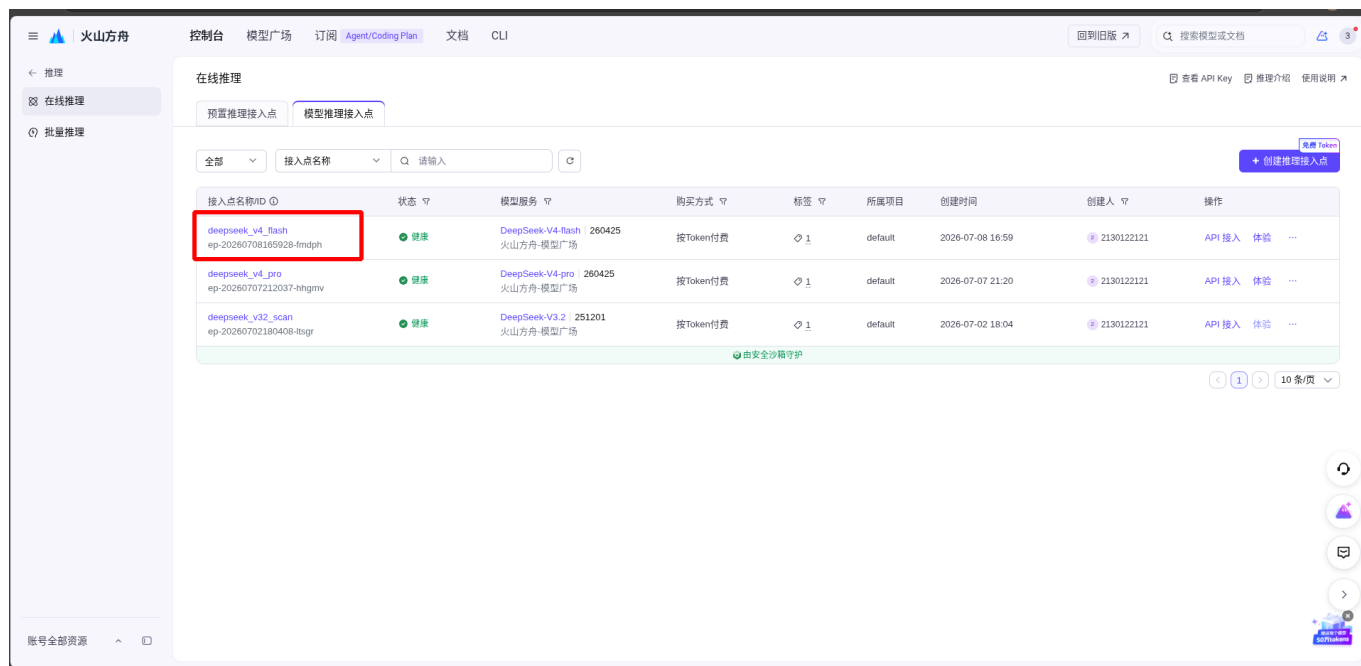
(3) 绑定模型ID

绑定deepseek_v4_flash模型ID





最后创建好的deepseek_v4_flash模型ID编号为红框里面的 ep-xxxxxxx



4、修改codex配置文件接入第三方API

现在我们使用上面完成的准备工作 API Key 和 模型ID，来修改codex配置文件。参考火山方舟官网给的API接入文档 <https://console.volcengine.com/ark/region:cn-beijing/docs/82379/2160841?lang=zh>，里面有详细的参考配置，里面还有 Claude Code接入的参考配置，经过测试，可以正常使用。

```
# vim 创建auth.json文件，填入以下内容。
vim ~/.codex/auth.json

#填入开通的API Key ark-xxxxxxxxxxxxxx，然后wq保存退出
{
  "OPENAI_API_KEY": "ark-xxxxxxxxxxxxxx"
}

# vim 创建config.toml文件，填入以下内容。
vim ~/.codex/config.toml

#填入创建的模型ID ep-xxxxxxxxxxxxxx，然后wq保存退出
model_provider = "huo_shan" #与下面的名称统一即可，都是 huo_shan
model = "ep-xxxxxxxxxxxxxx"

temperature = 0.1

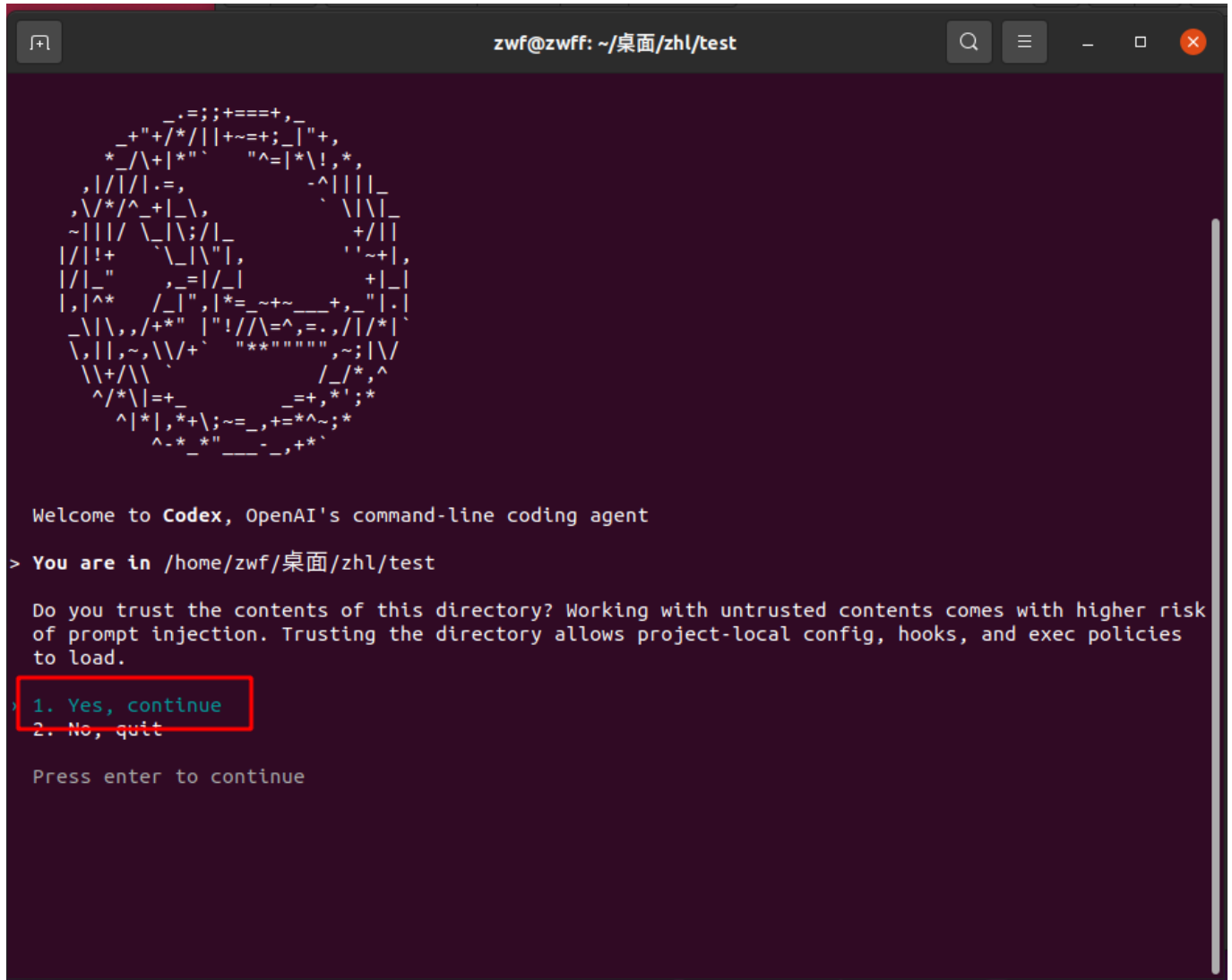
[tui.model_availability_nux]
"gpt-5.5" = 4

[model_providers]

[model_providers.huo_shan]
name = "huo_shan"
base_url = "https://ark.cn-beijing.volces.com/api/v3" #这个网址是火山官方统一的接入口
```

5、启动测试codex

#在任意位置打开终端
codex



```
zwf@zwff: ~/桌面/zhl/test

Welcome to Codex, OpenAI's command-line coding agent

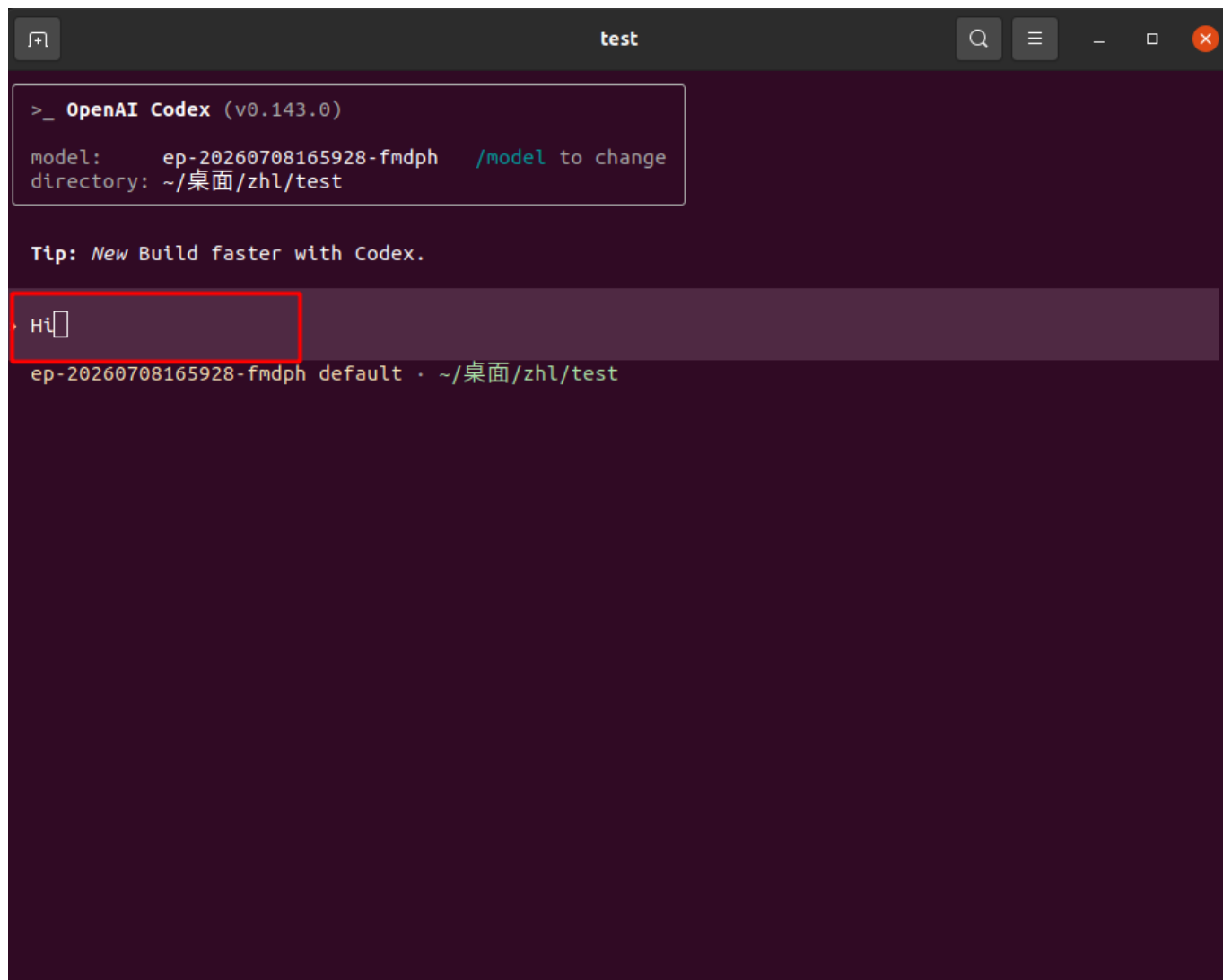
> You are in /home/zwf/桌面/zhl/test

Do you trust the contents of this directory? Working with untrusted contents comes with higher risk
of prompt injection. Trusting the directory allows project-local config, hooks, and exec policies
to load.

1. Yes, continue
2. No, quit

Press enter to continue
```

选择 yes 回车键确定

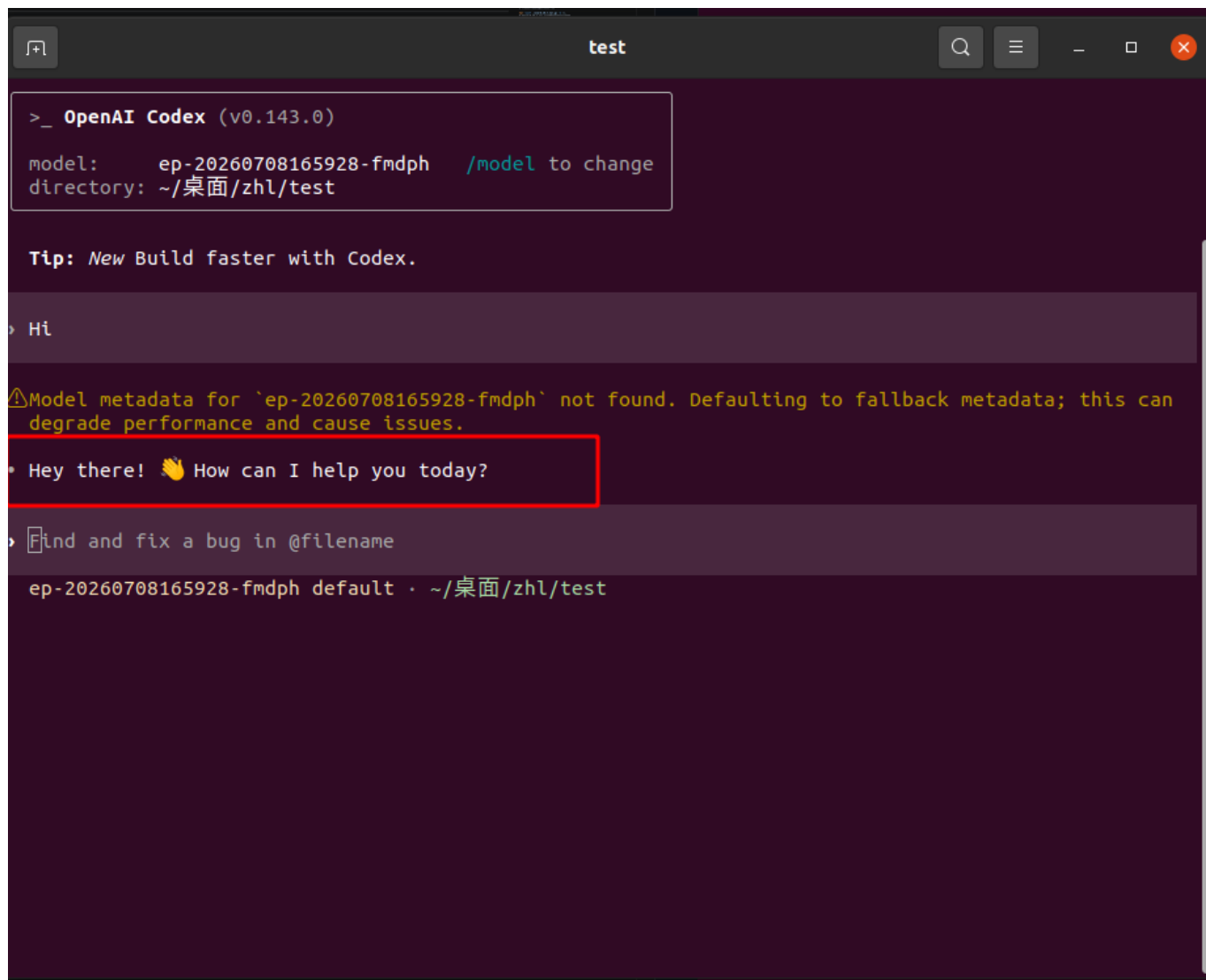


```
>_ OpenAI Codex (v0.143.0)
model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zh1/test

Tip: New Build faster with Codex.

Hi
ep-20260708165928-fmdph default · ~/桌面/zh1/test
```

在对话框中输入 Hi 进行测试



```
>_ OpenAI Codex (v0.143.0)
model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zhl/test

Tip: New Build faster with Codex.

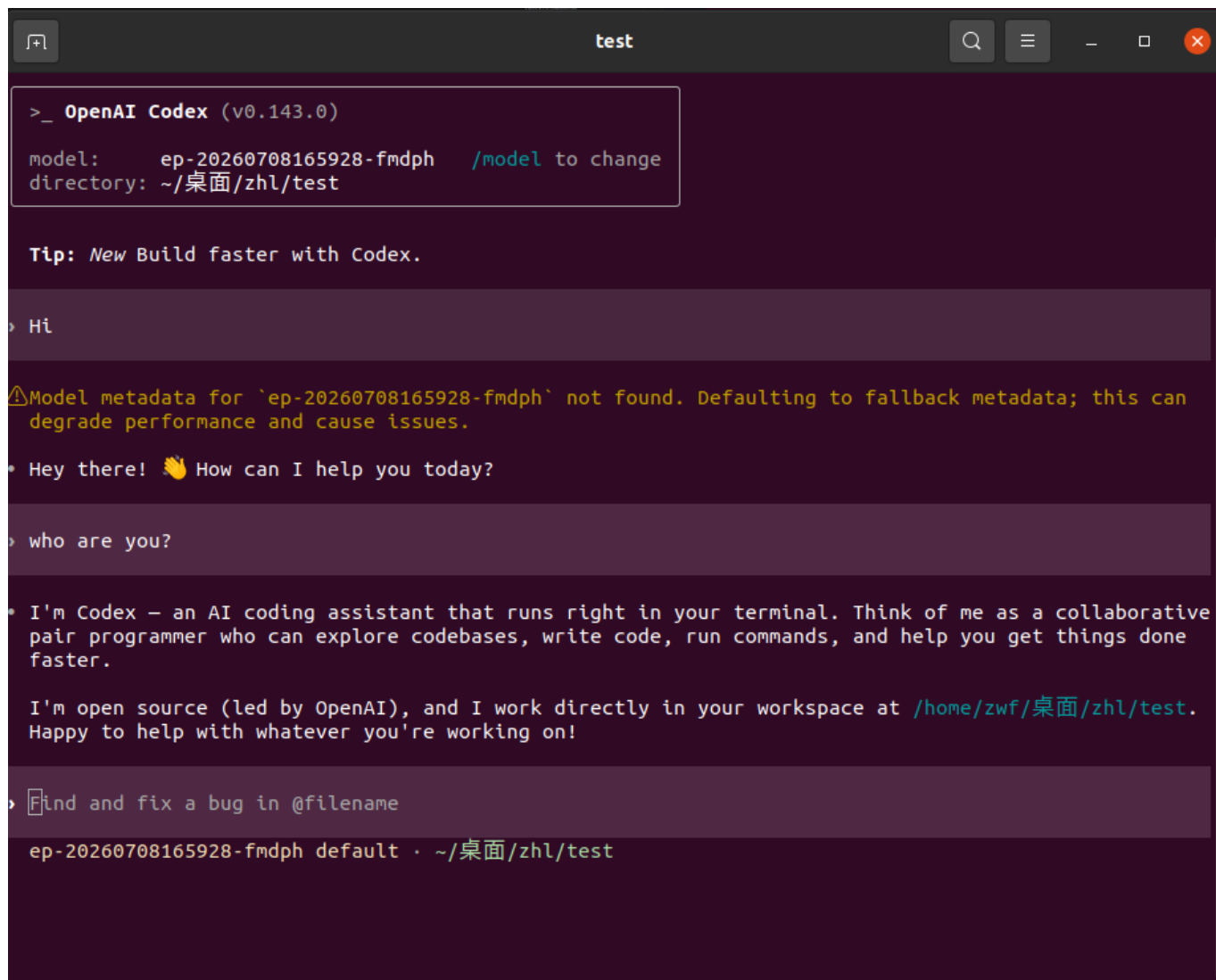
> Hi

⚠️Model metadata for `ep-20260708165928-fmdph` not found. Defaulting to fallback metadata; this can
degrade performance and cause issues.

• Hey there! 🙌 How can I help you today?

> Find and fix a bug in @filename

ep-20260708165928-fmdph default · ~/桌面/zhl/test
```



```
>_ OpenAI Codex (v0.143.0)
model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zhl/test

Tip: New Build faster with Codex.

> Hi

⚠ Model metadata for `ep-20260708165928-fmdph` not found. Defaulting to fallback metadata; this can
degrade performance and cause issues.

• Hey there! 🙌 How can I help you today?

> who are you?

• I'm Codex – an AI coding assistant that runs right in your terminal. Think of me as a collaborative
pair programmer who can explore codebases, write code, run commands, and help you get things done
faster.

I'm open source (led by OpenAI), and I work directly in your workspace at /home/zwf/桌面/zhl/test.
Happy to help with whatever you're working on!

> Find and fix a bug in @filename

ep-20260708165928-fmdph default · ~/桌面/zhl/test
```

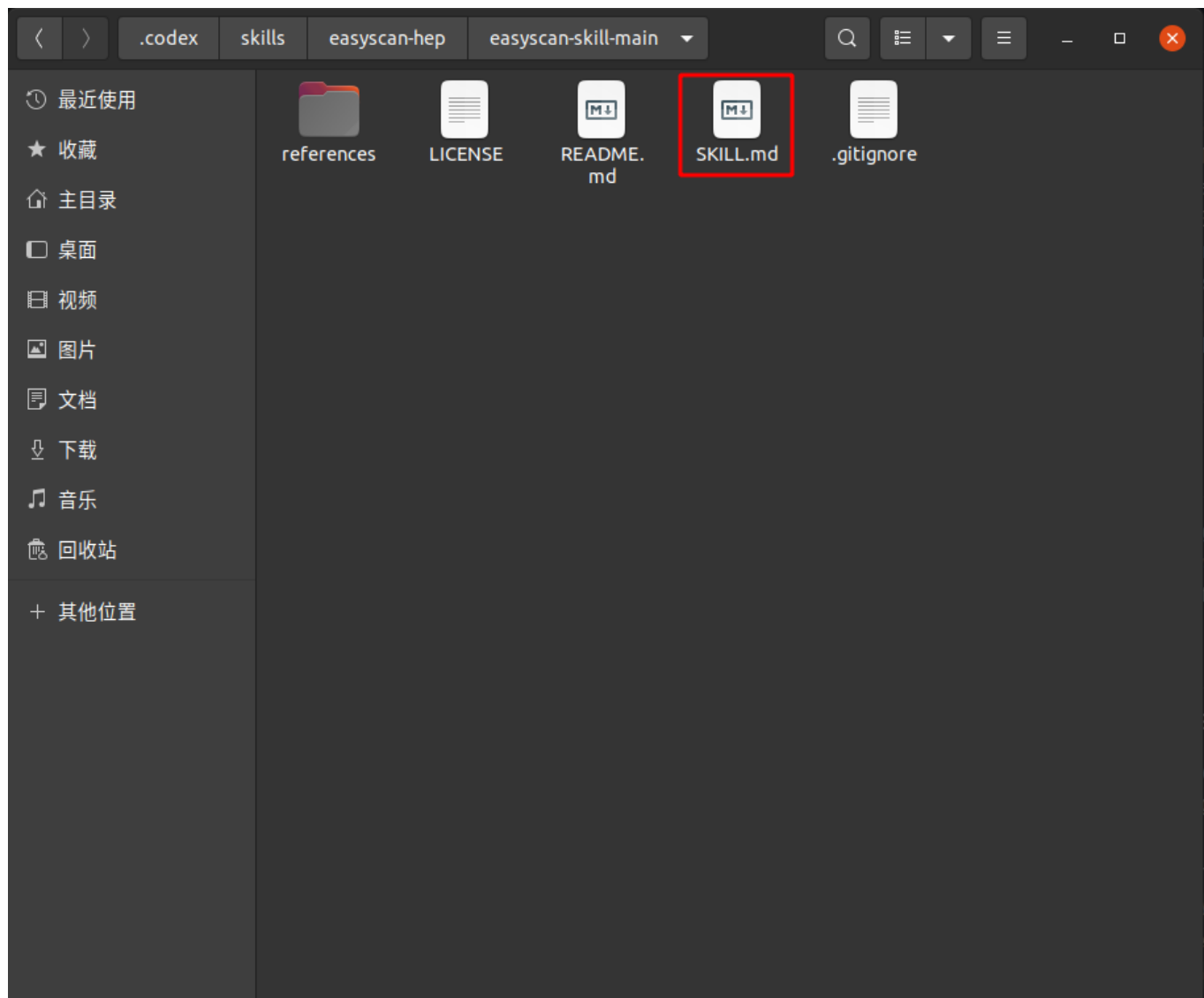
6、在codex中加载easyscan_skill

从github上下载skill

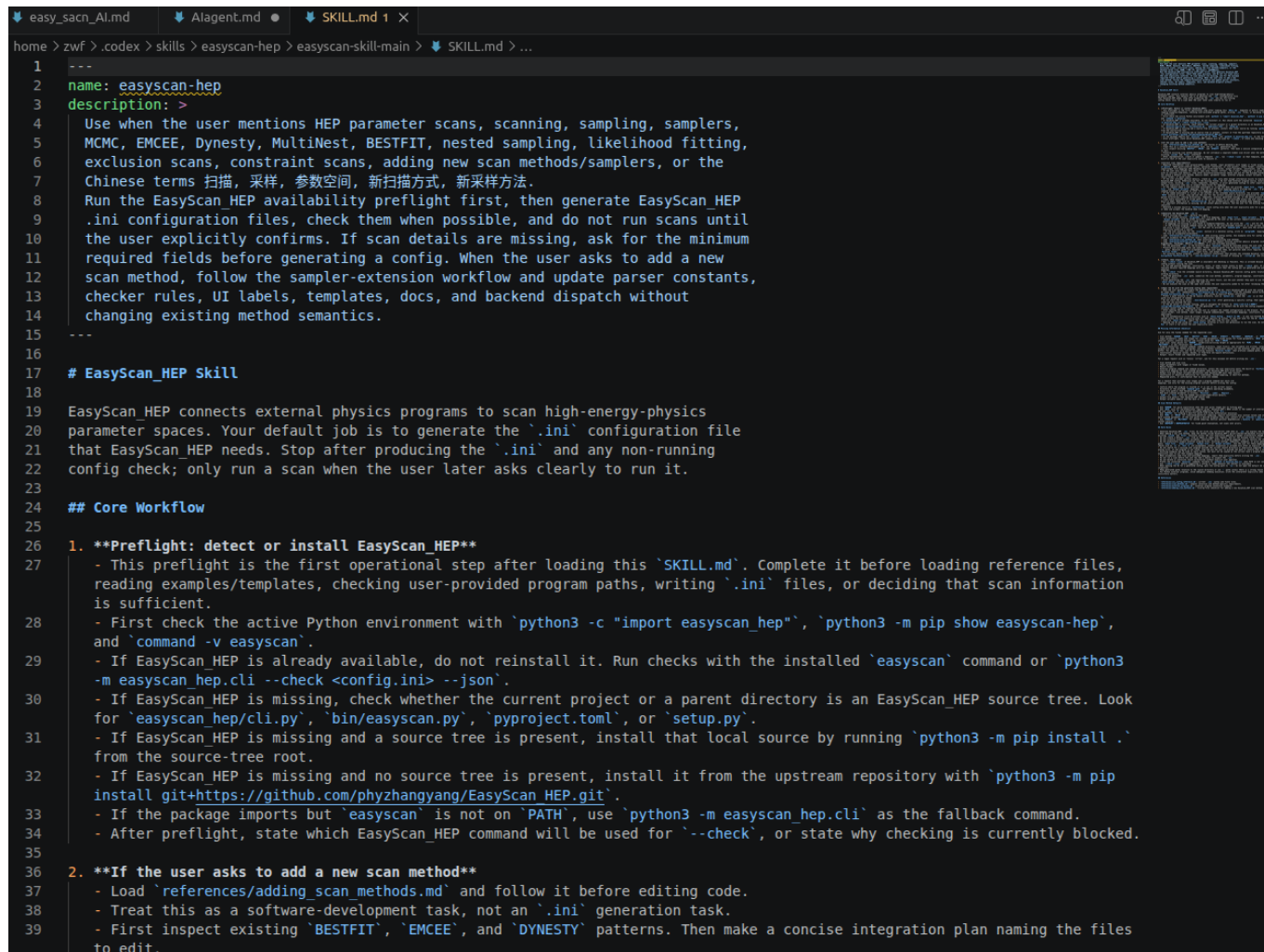
```
mkdir -p ~/.codex/skills
git clone https://github.com/Contract-Mediated-Agent/easyscan-skill.git
~/.codex/skills/easyscan-hep
```

或者从 `files/easyscan-skill-main.zip` 压缩包中解压到 `~/.codex/skills`

在 `easyscan-skill-main.zip` 文件夹中可以看到 `SKILL.md` 是一个markdown文件，



打开这个SKILL.md文件可以看到，这是告诉AI写easyscan.ini配置文件的说明书，

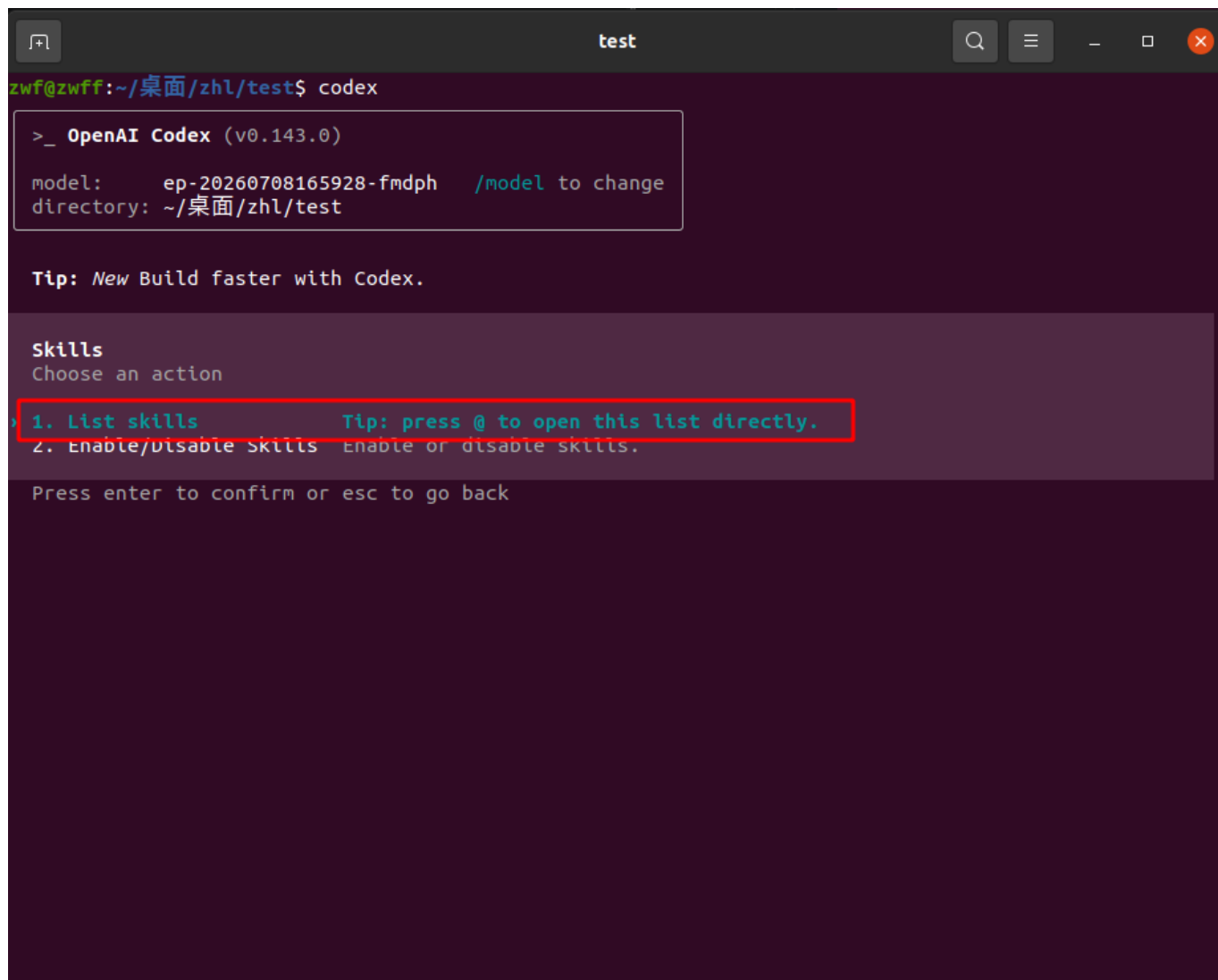


```

1  ---
2  name: easyscan-hep
3  description: >
4      Use when the user mentions HEP parameter scans, scanning, sampling, samplers,
5      MCMC, EMCEE, Dynesty, MultiNest, BESTFIT, nested sampling, likelihood fitting,
6      exclusion scans, constraint scans, adding new scan methods/samplers, or the
7      Chinese terms 扫描, 采样, 参数空间, 新扫描方式, 新采样方法.
8      Run the EasyScan_HEP availability preflight first, then generate EasyScan_HEP
9      .ini configuration files, check them when possible, and do not run scans until
10     the user explicitly confirms. If scan details are missing, ask for the minimum
11     required fields before generating a config. When the user asks to add a new
12     scan method, follow the sampler-extension workflow and update parser constants,
13     checker rules, UI labels, templates, docs, and backend dispatch without
14     changing existing method semantics.
15     ---
16
17 # EasyScan_HEP Skill
18
19 EasyScan_HEP connects external physics programs to scan high-energy-physics
20 parameter spaces. Your default job is to generate the '.ini' configuration file
21 that EasyScan_HEP needs. Stop after producing the '.ini' and any non-running
22 config check; only run a scan when the user later asks clearly to run it.
23
24 ## Core Workflow
25
26 1. **Preflight: detect or install EasyScan_HEP**
27     - This preflight is the first operational step after loading this `SKILL.md`. Complete it before loading reference files,
28       reading examples/templates, checking user-provided program paths, writing `.ini` files, or deciding that scan information
29       is sufficient.
30     - First check the active Python environment with `python3 -c "import easyscan_hep"`, `python3 -m pip show easyscan-hep`,
31       and `command -v easyscan`.
32     - If EasyScan_HEP is already available, do not reinstall it. Run checks with the installed `easyscan` command or `python3
33       -m easyscan_hep.cli --check <config.ini> --json`.
34     - If EasyScan_HEP is missing, check whether the current project or a parent directory is an EasyScan_HEP source tree. Look
35       for `easyscan_hep/cli.py`, `bin/easyscan.py`, `pyproject.toml`, or `setup.py`.
36     - If EasyScan_HEP is missing and a source tree is present, install that local source by running `python3 -m pip install .`
37       from the source-tree root.
38     - If EasyScan_HEP is missing and no source tree is present, install it from the upstream repository with `python3 -m pip
39       install git+https://github.com/phyzhangyang/EasyScan_HEP.git`.
40     - If the package imports but `easyscan` is not on `PATH`, use `python3 -m easyscan_hep.cli` as the fallback command.
41     - After preflight, state which EasyScan_HEP command will be used for `--check`, or state why checking is currently blocked.
42
43 2. **If the user asks to add a new scan method**
44     - Load `references/adding_scan_methods.md` and follow it before editing code.
45     - Treat this as a software-development task, not an `.ini` generation task.
46     - First inspect existing `BESTFIT`, `EMCEE`, and `DYNESTY` patterns. Then make a concise integration plan naming the files
47       to edit.

```

在codex对话框中输入 `skill`（可以使用`tab`键补全），选择`1、List skills`回车确认，



```
zwf@zwff:~/桌面/zhl/test$ codex

>_ OpenAI Codex (v0.143.0)

model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zhl/test

Tip: New Build faster with Codex.

Skills
Choose an action

1. List skills          Tip: press @ to open this list directly.
2. Enable/disable skills Enable or disable skills.

Press enter to confirm or esc to go back
```

用上下方向键选择`easyscan-hep`回车确认，

```
zwf@zwff:~/桌面/zhl/test$ codex

>_ OpenAI Codex (v0.143.0)

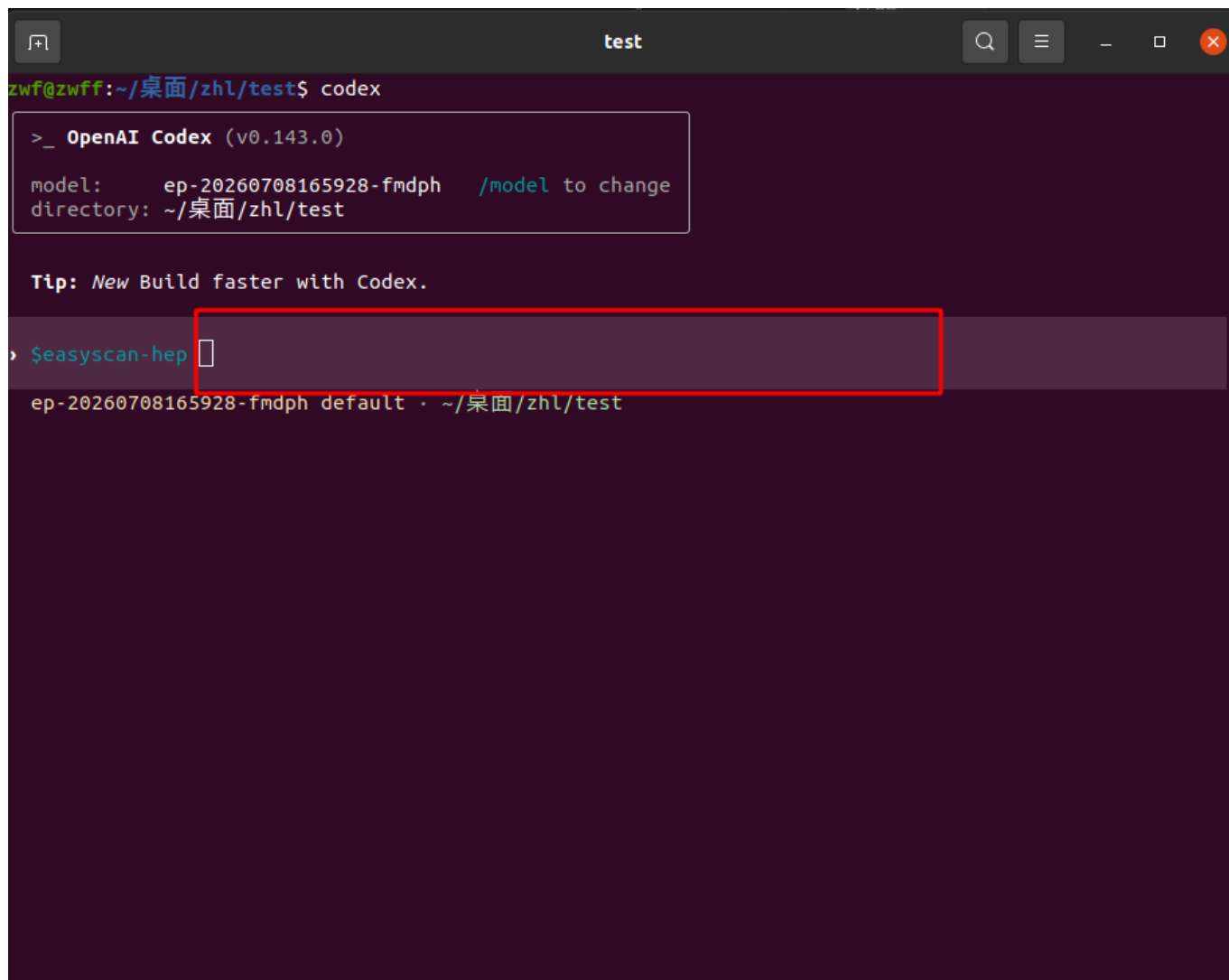
model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zhl/test

Tip: New Build faster with Codex.

> @

Image Gen      Generate or edit images for websites, games, and more      Skill
OpenAI Docs    Reference OpenAI docs, Codex self-knowledge, and model migration guidance Skill
Plugin Creator Scaffold plugins and marketplace entries                  Skill
Skill Creator   Create or update a skill                                    Skill
Skill Installer Install curated skills from openai/skills or other repos Skill
> easyscan-hep Use when the user mentions HEP parameter scans, scanning, sampling, sampl... Skill

enter insert · esc close · ←→switch search modes  [All Results]  Filesystem Only  Plugins
```



```
zwf@zwff:~/桌面/zhl/test$ codex

>_ OpenAI Codex (v0.143.0)

model:      ep-20260708165928-fmdph  /model to change
directory:  ~/桌面/zhl/test

Tip: New Build faster with Codex.

> Seasyscan-hep
```

在对话框中输入以下内容进行测试：Scan x from 0 to 3.14 and y from -3.14 to 3.14, randomly with 100 total points. Put results in example_random. Run ./TestFunction.py in utils/. Write x and y to utils/TestFunction_input.dat at row 1 columns 1 and 2. Read f from utils/TestFunction_output.dat at row 1 column 2. Add Gaussian $f = 1 \pm 0.2$. Make Color plots for x, y, f and x, y, Chi2.

最后应该会在终端路径下生成example_random.ini，大体内容如下，

```

[scan]
Result folder name: example_random
Scan method: RANDOM
Input parameters: x, flat, 0, 3.14
                  y, flat, -3.14, 3.14
Number of points: 100
Interval of print: 1

[program1]
Program name: TestFunction
Execute command: python3 TestFunction.py
Command path: .
Input file: 1, TestFunction_input.dat
Input variable: x, 1, Position, 1, 1
                y, 1, Position, 1, 2
Output file: 1, TestFunction_output.dat
Output variable: f, 1, Position, 1, 2

[constraint]
Gaussian: f, 1, 0.2

[plot]
Color: x, y, f, color_xy_f
Color: x, y, Chi2, color_xy_chi2

```

参考答案:

```

1 [scan]
2 Result folder name: example_random
3 Sampler: random
4 # VarID Prior Min MAX
5 Input parameters: x, flat, 0, 3.14
6 y, flat, -3.14, 3.14
7 Points number: 100
8 Parallel threads: 1
9 Parallel folder: utils
10
11 [program1]
12 Execute command: ./TestFunction.py
13 Command path: utils/
14 # FileID FileName
15 Input file: 1, utils/TestFunction_input.dat
16 # VarID FileID Method Row Column
17 Input variable: x, 1, Position, 1, 1

```

18	y,	1,	Position, 1,	2
19	Output file:	1,	utils/TestFunction_output.dat	
20	#	VarID	FileID	Method Row Column
21	Output variable:	f,	1,	Position, 1, 2
22				
23	[constraint]			
24	#	VarID	Mean	StandardDeviation
25	Gaussian:	f,	1,	0.2
26				
27	[plot]			
28	#	X-axis	Y-axis	Color FigureName
29	Color:	x,	y,	f, test_run_random
30		x,	y,	Chi2