

Claude code（企业使用，报错解决）

1. 官网安装nvm
2. 使用nvm安装nodejs
3. 使用nodejs安装Claude code（原因，我们是企业使用，需要专业，因此需要使用nvm包管理器来管理node环境）
4. 打开创建好的文件夹，cmd打开，输入claude。

若遇到报错

代码块

```
1 Unable to connect to Anthropic services
2
3 Failed to connect to api.anthropic.com: ERR_BAD_REQUEST
4
5 Please check your internet connection and network settings.
6
7 Note: Claude Code might not be available in your country. Check supported
countries at
8 https://anthropic.com/supported-countries
```

解决方案，打开电脑主文件夹



代码块

```
1 搜索claude.json
2 {
3   "installMethod": "unknown",
4   "autoUpdates": true,
5   "cachedStatsigGates": {
```

```
6     "tengu_disable_bypass_permissions_mode": false
7 },
8 "firstStartTime": "2025-09-22T16:39:55.181Z",
9 "userID": "3368ffc276d5512e57211bca265c52edb8077600db2ba7c26ebfde99d2874b50",
10 "projects": {
11     "C:\\claude-code\\code-1": {
12         "allowedTools": [],
13         "history": [],
14         "mcpContextUris": [],
15         "mcpServers": {},
16         "enabledMcpjsonServers": [],
17         "disabledMcpjsonServers": [],
18         "hasTrustDialogAccepted": false,
19         "projectOnboardingSeenCount": 0,
20         "hasClaudeMdExternalIncludesApproved": false,
21         "hasClaudeMdExternalIncludesWarningShown": false
22     },
23     "C:\\claude-code": {
24         "allowedTools": [],
25         "history": [],
26         "mcpContextUris": [],
27         "mcpServers": {},
28         "enabledMcpjsonServers": [],
29         "disabledMcpjsonServers": [],
30         "hasTrustDialogAccepted": false,
31         "projectOnboardingSeenCount": 0,
32         "hasClaudeMdExternalIncludesApproved": false,
33         "hasClaudeMdExternalIncludesWarningShown": false
34     }
35 }, //注意这里要添加英文逗号
36     "hasCompletedOnboarding": true //增加这一行
37 }
```

使用记事本打开即可，保存。然后回到文件夹cmd，运行claude即可解决。

5.可以正常使用之后，退出ctrl+c。安装Claude code router，去github搜索即可。命令行为：npm install -g @musistudio/claude-code-router

6.在主文件夹创建

.claude-code-router

创建这个文件夹

这个文件夹里面创建

config.json

这个配置文件，文件内容直接复制粘贴官网的。

代码块

```
1  {
2      "APIKEY": "your-secret-key",
3      "PROXY_URL": "http://127.0.0.1:7890",
4      "LOG": true,
5      "API_TIMEOUT_MS": 600000,
6      "NON_INTERACTIVE_MODE": false,
7      "Providers": [
8          {
9              "name": "openrouter",
10             "api_base_url": "https://openrouter.ai/api/v1/chat/completions",
11             "api_key": "sk-xxx",
12             "models": [
13                 "google/gemini-2.5-pro-preview",
14                 "anthropic/clause-sonnet-4",
15                 "anthropic/clause-3.5-sonnet",
16                 "anthropic/clause-3.7-sonnet:thinking"
17             ],
18             "transformer": {
19                 "use": ["openrouter"]
20             }
21         },
22         {
23             "name": "deepseek",
24             "api_base_url": "https://api.deepseek.com/chat/completions",
25             "api_key": "sk-xxx",
26             "models": ["deepseek-chat", "deepseek-reasoner"],
27             "transformer": {
28                 "use": ["deepseek"],
29                 "deepseek-chat": {
30                     "use": ["tooluse"]
31                 }
32             }
33         },
34         {
35             "name": "ollama",
36             "api_base_url": "http://localhost:11434/v1/chat/completions",
37             "api_key": "ollama",
38             "models": ["qwen2.5-coder:latest"]
39         },
40         {
41             "name": "gemini",
42             "api_base_url":
"https://generativelanguage.googleapis.com/v1beta/models/",
```

```
43     "api_key": "sk-xxx",
44     "models": ["gemini-2.5-flash", "gemini-2.5-pro"],
45     "transformer": {
46         "use": ["gemini"]
47     }
48 },
49 {
50     "name": "volcengine",
51     "api_base_url": "https://ark.cn-
beijing.volces.com/api/v3/chat/completions",
52     "api_key": "sk-xxx",
53     "models": ["deepseek-v3-250324", "deepseek-r1-250528"],
54     "transformer": {
55         "use": ["deepseek"]
56     }
57 },
58 {
59     "name": "modelscope",
60     "api_base_url": "https://api-
inference.modelscope.cn/v1/chat/completions",
61     "api_key": "",
62     "models": ["Qwen/Qwen3-Coder-480B-A35B-Instruct", "Qwen/Qwen3-235B-A22B-
Thinking-2507"],
63     "transformer": {
64         "use": [
65             [
66                 "maxtoken",
67                 {
68                     "max_tokens": 65536
69                 }
70             ],
71             "enhancetool"
72         ],
73         "Qwen/Qwen3-235B-A22B-Thinking-2507": {
74             "use": ["reasoning"]
75         }
76     }
77 },
78 {
79     "name": "dashscope",
80     "api_base_url": "https://dashscope.aliyuncs.com/compatible-
mode/v1/chat/completions",
81     "api_key": "",
82     "models": ["qwen3-coder-plus"],
83     "transformer": {
84         "use": [
85             [
```

```
86             "maxtoken",
87             {
88                 "max_tokens": 65536
89             }
90         ],
91         "enhancetool"
92     ]
93 }
94 },
95 {
96     "name": "aihubmix",
97     "api_base_url": "https://aihubmix.com/v1/chat/completions",
98     "api_key": "sk-",
99     "models": [
100         "Z/glm-4.5",
101         "claude-opus-4-20250514",
102         "gemini-2.5-pro"
103     ]
104 }
105 ],
106 "Router": {
107     "default": "deepseek,deepseek-chat",
108     "background": "ollama,qwen2.5-coder:latest",
109     "think": "deepseek,deepseek-reasoner",
110     "longContext": "openrouter,google/gemini-2.5-pro-preview",
111     "longContextThreshold": 60000,
112     "webSearch": "gemini,gemini-2.5-flash"
113 }
114 }
```

注意上面的大部分内容都需要删除，但我们进入webui里面进行操作。

ccr ui

这个在哪里运行都可以，最好就是在之前需要操作的文件里面运行

进去之后，把不需要的都删除，ollama不需要尽量删除。我们国内用户就选择deepseek就行了，其他的自行探索，如果你用的是中转站，就在ui里面去增加新的模型。记得保存重启。

问题排除。

如果出现服务无法启动，

按照以下去排除。

7. 检查语法

代码块

```
1 检查语法代码
2 ccr validate
3
4 重启服务
5 ccr stop
6 ccr start
7 ccr status
8
9 这几个都没有问题的情况下执行下面
10 ccr stop
11 ccr code
12 如果还不行，就是说在你重启服务之后，都显示正常，
13 执行ccr code还是出现服务未运行，那可能是端口被占用了。
14 把问题代码发给AI，
15 应该会检查端口占用情况。
16 netstat -ano | findstr :3456
17 这里不一定是3456，看具体代码情况。
18 tasklist /fi "pid eq 7684"
19 这里不一定是7684，要看你上面那个命令行信息是什么。
20 taskkill /pid 7684 /f
21 将进程终止。
22 执行
23 ccr code
24
```

8.claude code常用命令行

/init 作用是对当前文件夹里面的内容进行通读，整理一个文档出来，文档自动命名为CLAUDE.md

/compact 要求内容 作用是整理上下文，去除一些无效的上下文，例如：/compact 主要保留前端对话

/clear 清楚对话内容，相当于开启新对话。

think或者think hard或者think harder或者ultrathink 在对话前面加这些，就是让AI进行深度思考，强度逐级递增。例如：think 如何修改整体风格

!感叹号用于执行命令行，这样子不用躲开一个终端，还能被ai当成上下文。例如：!npm install

#井号是进入记忆模式，后面写的内容就是作为上下文的。可以选择用户级别和项目级别，项目级别保存在CLAUDE.md里面，只对当前文件夹里面的项目生效。用户级别则是任何项目文件夹你使用的时候都生效的，相当于全局。我觉得可以写一写自己的系统型号之类的。

/ide 这个使用上感觉有些问题，首先是我的插件安装正确了吗？在idea中应该安装什么呢？还有他说的路径问题。这个只能够使用vscode，我使用idea时不行，有问题也找不到idea。vscode下载插件

即可使用，这样子vscode选中的代码也会同步到上下文。也可以直接修改代码，将修改前后的代码在vscode里面显示出来。

9.claude code使用mcp

代码块

```
1 C:\claude-code\code-1>claude mcp add --transport http context7
https://mcp.context7.com/mcp --header "Authorization:
*****
```

***这个内容是填写自己的context7的mcp密钥，去官网获取。在你对应的文件夹里面运行这个命令即可。然后再ccr code 再/mcp就能看到对应的服务了。

使用的时候告诉AI，使用context7就会进行调用。删除mcp服务，claude mcp remove context7即可，这是在命令行执行的。

10.权限设置

Ccr code --dangerously-skip-permissions

这样子就不用总是访问权限的问题的，它会直接进行修改和调用工具。

11.自定义命令

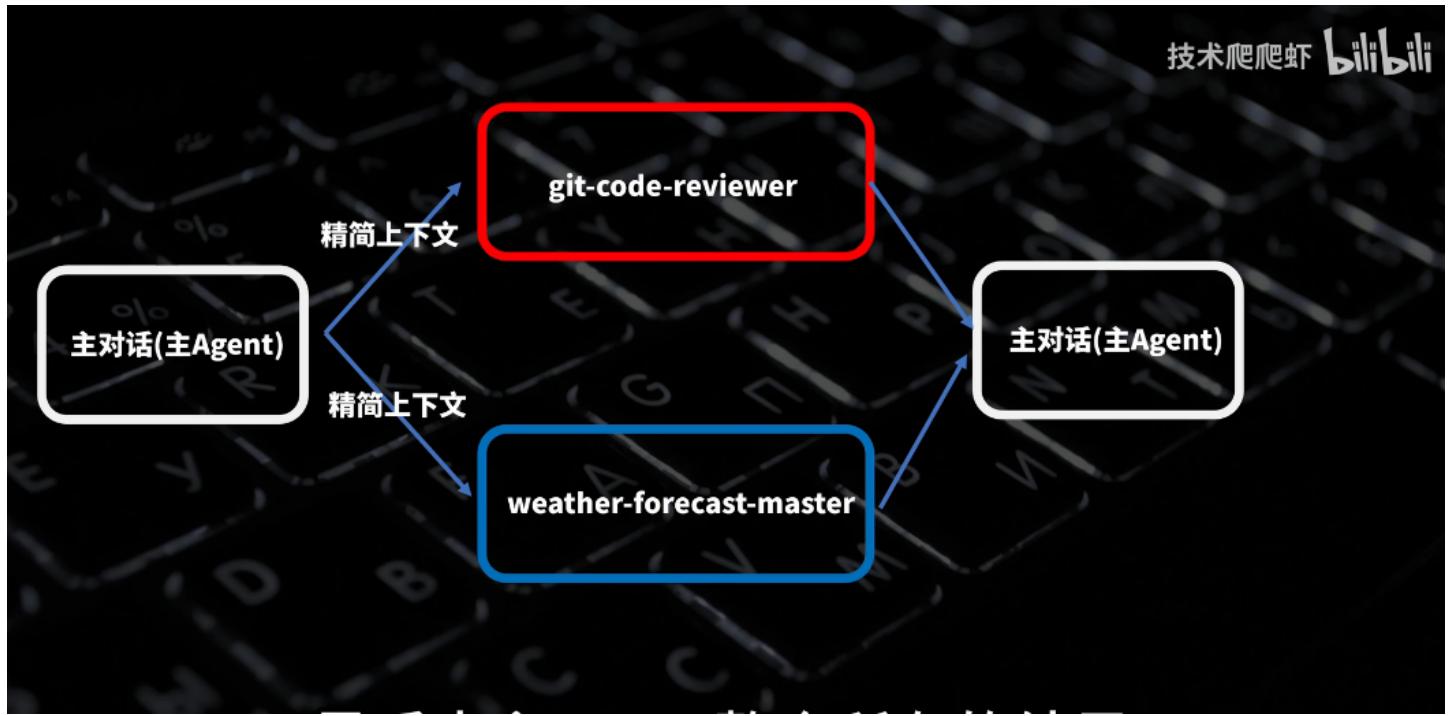
12.触发事件

hooks，执行完一个操作后会执行另一个hooks操作，hooks操作可以定义。

13.sub agent多智能体并行运行

/agents

进入创建窗口，选择项目级别，进行自然语言描述，选择颜色



14.github cli

可以探索

15.历史记录

/resume

选择对应内容，回车。按两下esc还可以选择具体的对话条数。但是这个只能回退对话，无法回退代码。

可以使用ccundo，这个需要单独安装。这个好像和github绑定的。

16.导出内容

/export

他会将内容复制到粘贴板里面

16.可视化界面

claudia

github里面找，然后再找里面有其他人做的安装包。

那我前面做的那些算什么？我真的是大大的问号。

更新：这个可视化应用真的是一坨，特别难用，还是终端好用。这个可视化没火起来肯定有他的道理。