

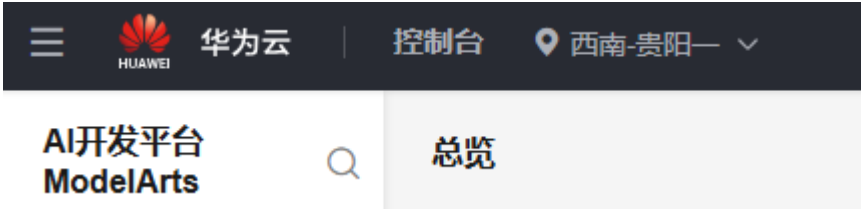
# Vscode\_DeepSeek等生成式模型集成

## 1.安装vscode插件

- vscode -> 扩展 -> 搜索插件:**continue(推荐)** 或 **cline(配置简洁)** 或 **aider** -> 安装

## 2.选择模型供应商(以 第三方服务商 + continue 为例)

部分服务器厂商会存在模型访问限制, 如华为限一分钟五次访问

- 华为ModelArts
    - 登录网址 [https://console.huaweicloud.com/modelarts/?region=cn-southwest-2#/model-studio/homepage]
- 
- 左上角选择服务器部署区域 (涉及网络时延)
  - ModelArts Studio -> 模型工具 -> 模型部署 -> 领取免费额度 -> 调用 -> 创建API\_Key -> 获取访问密钥
    - **APIurl** [https://infer-modelarts-cn-southwest-2.modelarts-infer.com/v1/infers/952e4f88-ef93-4398-ae8d-af37f63f0d8e/v1/chat/completions]
    - **访问密钥**: EG7T\_kivme4tDmXS3-qGzlpkDJq55ow31iili4DidcOnHwe9fN7zPtg80kWBaV3IWXVYutrAFxV3eAKIs0NpYw (**个人实例**)
- SiliconFlow
  - 登陆网址 [https://cloud.siliconflow.cn/models]
    - 左侧选择框 -> 账户管理 -> API密钥
    - **APIurl**: [https://api.siliconflow.cn/v1]
    - **密钥**: sk-zznwocxbuiousddacxdcsgcncgthhxcjrtthmtxfeeexfq (**个人实例**)
  - 选择需求模型, 配置文件中的 **models参数** 即为选择模型名
- Openrouter(**推荐**)
  - 登陆网址 [https://openrouter.ai/]
    - 需科学上网
    - **APIurl**: [https://openrouter.ai/api/v1]
    - **密钥**: 自行获取

**Continue** 实例配置, 其中 **models** 和 **tabAutocompleteModel** 需要替换为需求模型

[model/apiBase/apiKey/provider], 以主流开源插件 **cline**及**continue** 为例, 它并不一定支持所有公司的参数解析请求, 大部分应该是以openai为基准, 调取的第三方api若无明确说明, 先将 provider 置为 **openai**

```

{
  "models": [ //通用文本生成模型，可以配置多个，按需求加即可
    {
      "title": "deepseek-ai/DeepSeek-R1-huawei",
      "model": "DeepSeek-R1",
      "contextLength": 30000, //上下文最大限制
      "provider": "vllm", //使用类似openai的接口形式调用第三方服务
      "apiBase": "https://infer-modelarts-cn-southwest-2.modelarts-
infer.com/v1/infers/952e4f88-ef93-4398-ae8d-af37f63f0d8e/v1/chat/completions",
      "apiKey": "Bearer EG7T_kivme4tDmXS3-
qGzIpkDJq55ow31iiIi4DidcOnHwe9fN7zPtg80kWBaV3IWXVYutrAFxV3eAKls0NpYw",
      "requestHeaderProperties": { // 添加 requestHeaderProperties
        "X-Auth-Token": "${apiKey}" // 设置 X-Auth-Token 头, 值为 apiKey 的值
      }
    },
    {
      "title": "openrouter/deepseekR1",
      "model": "deepseek/deepseek-r1:free",
      "contextLength": 30000, //上下文最大限制
      "provider": "openrouter", //使用类似openai的接口形式调用第三方服务
      "apiBase": "https://openrouter.ai/api/v1",
      "apiKey": "",
      "requestOptions": { //配置项
        "extraBodyProperties": {
          "transforms": []
        }
      }
    },
    {
      "title": "deepseek-ai/DeepSeek-R1-silicon",
      "model": "deepseek-ai/DeepSeek-R1-Distill-Qwen-7B",
      "contextLength": 30000, //上下文最大限制
      "provider": "openai", //使用类似openai的接口形式调用第三方服务
      "apiBase": "https://api.siliconflow.cn/v1",
      "apiKey": "sk-zznwocxbuiounsdddacxdcsgcncgthhxcjrtthmtxffeexfq",
      "requestOptions": { //配置项
        "extraBodyProperties": {
          "transforms": []
        }
      }
    }
  ],
  "customCommands": [ //自定义预置命令配置
    {
      "name": "test", //选中代码 ctrl+L ，输入/，选择预先设置的命令，会自动执行下
      面的prompt(代码自动填充进input)
      "prompt": "{{{ input }}}\n\nWrite a comprehensive set of unit tests for
the selected code. It should setup, run tests that check for correctness including
important edge cases, and teardown. Ensure that the tests are complete and
sophisticated. Give the tests just as chat output, don't edit any file.",
      "description": "Write unit tests for highlighted code"
    },
    {

```

```

        "name": "explain",
        "prompt": "{{{ input }}}\n\n详细解释所选代码。详细说明其功能、用途和工作原理。使用示例来说明您的解释。",
        "description": "解释代码"
    },
    {
        "name": "fix",
        "prompt": "{{{ input }}}\n\n分析并修复代码中的错误。提供详细的解释和解决方案，包括但不限于语法错误、逻辑错误或性能问题。确保您的解决方案是正确的，并且能够解决所有已知的问题。",
        "description": "修复代码"
    },
    {
        "name": "check",
        "prompt": "{{{ input }}}\n\n请阅读突出显示的代码并检查是否有任何错误。您应该查找以下内容并保持高度警惕：\n- 语法错误\n- 逻辑错误\n- 安全漏洞\n- 性能问题\n- 任何其他看似错误的内容\n\n一旦您发现错误，请尽可能清楚地解释，但不要使用多余的词语。例如，不要说“我认为第 5 行有一个语法错误”，而应该说“第 5 行有语法错误”。针对发现的每个错误，给出一个要点作为答案。",
        "description": "检查代码是否存在错误"
    }
],
"tabAutocompleteModel": [ //tab代码补全模型
    {
        "title": "deepseek-ai/DeepSeek-R1",
        "provider": "openai",
        "model": "deepseek-ai/DeepSeek-R1-Distill-Qwen-7B",
        "contextLength": 30000,
        "apiBase": "https://api.siliconflow.cn/v1",
        "apiKey": "sk-zznwocxbuiounsdddacxdcsgcncgthhxcjrtthmtxffeexfq",
        "useLegacyCompletionsEndpoint": false
    }
],
"contextProviders": [ //上下文信息获取配置
    {
        "name": "code",
        "params": {
            "nRetrieve": 25, //代码段检索数量
            "nFinal": 5, //最终依赖数量
            "useReranking": true //是否使用rerank
        },
        "useLegacyCompletionsEndpoint": false
    },
    {
        "name": "docs",
        "params": {}
    },
    {
        "name": "diff", //提供代码变更 (Diff) 信息作为上下文
        "params": {
            "nRetrieve": 25,
            "nFinal": 5,
            "useReranking": true
        },
    },

```

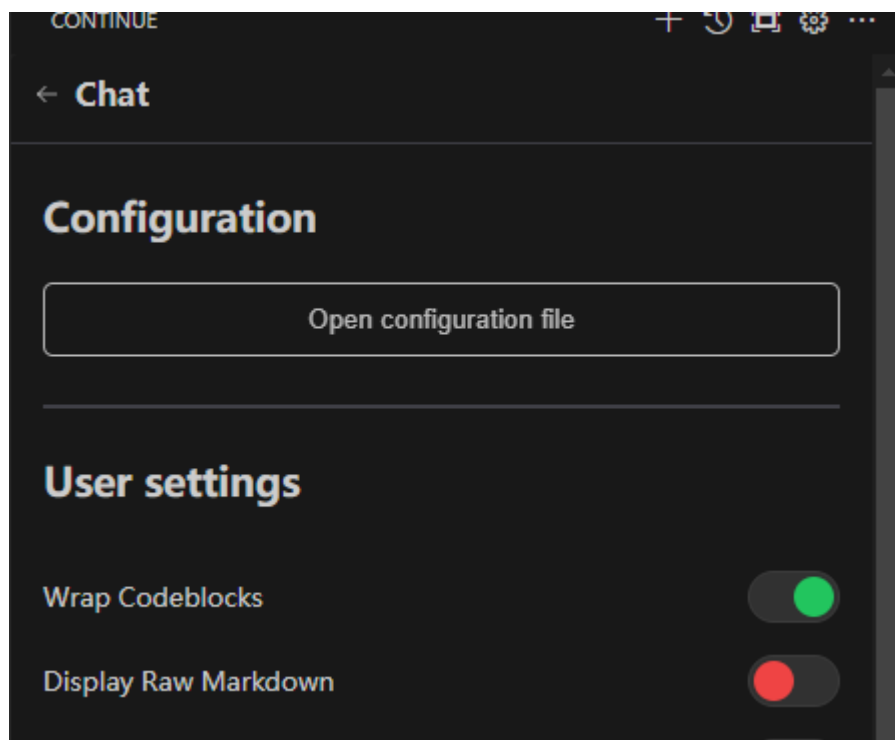
```
    "useLegacyCompletionsEndpoint": false
  },
  {
    "name": "terminal",
    "params": {}
  },
  {
    "name": "problems",
    "params": {}
  },
  {
    "name": "folder",
    "params": {}
  },
  {
    "name": "codebase",
    "params": {
      "nRetrieve": 25,
      "nFinal": 5,
      "useReranking": true
    },
    "useLegacyCompletionsEndpoint": false
  }
],
"slashCommands": [ //斜杠命令配置
  {
    "name": "edit",
    "description": "Edit selected code"
  },
  {
    "name": "comment",
    "description": "Write comments for the selected code"
  },
  {
    "name": "share",
    "description": "Export the current chat session to markdown"
  },
  {
    "name": "cmd",
    "description": "Generate a shell command"
  },
  {
    "name": "commit",
    "description": "Generate a git commit message"
  }
],
"embeddingsProvider": { //Embedding模型用于将文本转换为embeddings，以便进行语义相似
  度计算和信息检索
    "provider": "openai",
    "model": "BAAI/bge-m3", //一个免费的生成嵌入向量模型算法
    "apiBase": "https://api.siliconflow.cn/v1",
    "apiKey": "sk-zznwocxbuiousdddacxdcsgcncgthhxcjrthmtxffeexfq"
  },
  "rerank": { //重排序模型，此处选择了一个免费上下文权重模型算法
```

```

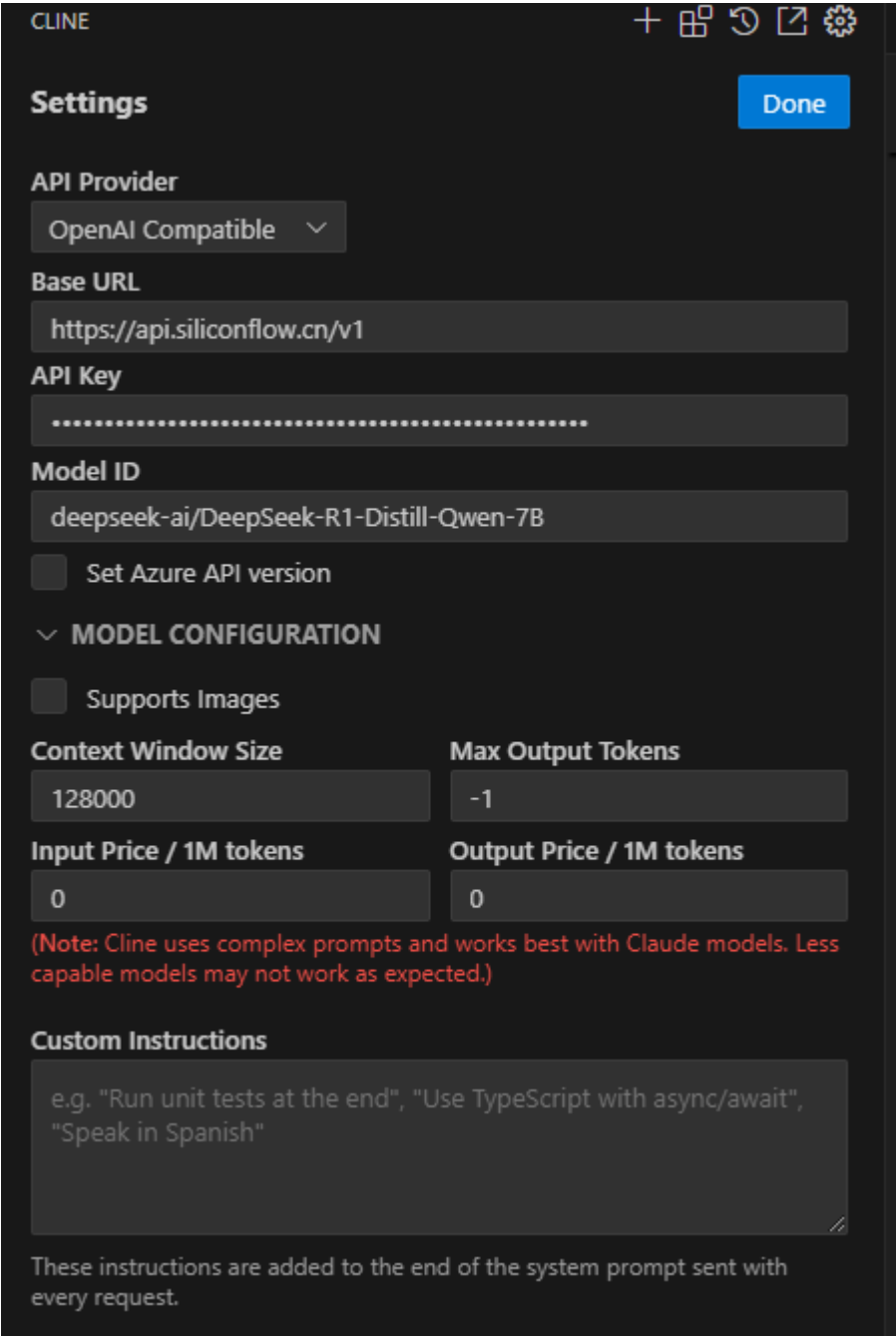
    "title": "BAAI/bge-reranker-v2-m3",
    "model": "BAAI/bge-reranker-v2-m3",
    "provider": "openai",
    "apiBase": "https://api.siliconflow.cn/v1",
    "apiKey": "sk-zznwocxbuiousdddacxdcsgcncgthhxcjrtthmtxffeexfq"
  },
  "tabAutocompleteOptions": { //tab自动补全模型配置
    "template": "Please teach me what I should write in the `hole` tag, but
without any further explanation and code backticks, i.e., as if you are directly
outputting to a code editor. It can be codes or comments or strings. Don't provide
existing & repetitive codes. If the provided prefix and suffix contain incomplete
code and statement, your response should be able to be directly concatenated to
the provided prefix and suffix. Also note that I may tell you what I'd like to
write inside comments. \n{{{prefix}}}<hole></hole>{{{suffix}}}\n\nPlease be aware
of the environment the hole is placed, e.g., inside strings or comments or code
blocks, and please don't wrap your response in ``` . You should always provide non-
empty output.\n",
    "maxPromptTokens": 2048, //最大token限制
    "prefixPercentage": 0.85, //前缀
    "maxSuffixPercentage": 0.15, //后缀，即对自动补全而言，依赖于选定光标位置前85%
的代码及后15%的代码
    "debounceDelay": 500, //去抖动延迟
    "slidingWindowPrefixPercentage": 0.85, //针对长代码文件设置滑动窗口
    "slidingWindowSize": 350, //滑动窗口大小
    "maxSnippetPercentage": 0.6, //补全代码片段非重复比例
    "recentlyEditedSimilarityThreshold": 0.3, //最近编辑相似度阈值，用于判断是否使
用最近编辑的代码
    "onlyMyCode": false, //是否只使用自己代码
    "useOtherFiles": false, //是否考虑其它文件
    "useRecentlyEdited": true, //是否使用最近编辑过的文件
    "recentLinePrefixMatchMinLength": 7 //最近编辑行匹配最小长度
  }
}

```

**User settings** 根据自己需求进行选择 (主要涉及对输出内容的格式化调整)



**cline** 不对外提供可修改详细配置信息, 基本需求内容依然为`model/BaseUrl/ApiKey/provider`, 其余配置按需要自行设置或默认即可



本地配置时, 使用ollama安装部署本地模型后, provider选择ollama, model为自己添加模型.

3.使用说明

continue

- 基本使用
  - 安装对应插件后 -> 点击对应图标 -> 出现对话框 -> 进行对话
  - 完成配置后 -> 在代码页面 -> 出现代码补全信息 -> Tab应用

```

1 def merge_sort(arr):
2     if len(arr) <= 1:
3         return arr
4
5     mid = len(arr) // 2
6     left = arr[:mid]
7     right = arr[mid:]
8
9     left = merge_sort(left)

```

- 选择代码后按 `ctrl+L`, 在弹出对话框内打出 `/`, 选择预设的prompt选项即可

```

15 def merge(left, right):
16     merged = []
17     i = j = 0
18     while i < len(left) and j < len(right):
19         if left[i] <= right[j]:
20             merged.append(left[i])
21             i += 1
22         else:
23             merged.append(right[j])
24             j += 1
25     merged.extend(left[i:])
26     merged.extend(right[j:])
27     return merged
28

```

☐ /share  
☐ /cmd  
☒ /commit Generate a git commit message  
☐ /test  
☐ /explain  
☐ /fix  
☐ /check

- 选择代码后鼠标右键 -> continue选项 -> 生成注释等功能

## cline

- 安装对应插件后 -> 点击对应图标 -> 出现对话框 -> 进行对话



