deepseek

deepseek 环境搭建
docker搭建deepseek运行环境和open-webui deepseek和dify环境搭建
deepseek嵌入工作流实现直接驱动业务的探索
网站采集工具firecrawl
deepseek提示词小技巧
deepseek开发流程
通义千问&&deepseek模型对比
优秀提示词示例

deepseek 环境搭建

这篇文章是入门文章，没有什么实用价值，仅是将运行deepseek的环境运行起来了。

主要软件

- ollama: https://ollama.com/download
- open-webui: 简单的chat页面, https://github.com/open-webui/open-webui
- dify: 大型编排平台, https://github.com/langgenius/dify
- docker
- python
  - 3.11: open-webui需要的运行时版本
- 上网: https://iovhm.com/book/books/cee63/page/9872e

ollama可以直接在容器运行

不用担心在容器中会对模型和规则等网络的性能损失的实践，除了网络方面会有损失外，其他方面并没有损失，可以将网络设置为

相反，如果在主机直接运行，各种版本的依赖，新版本的升级，环境更加容易出问题。

```
version: "3"
services:
  ollama:
    image: harbor.iovhm.com/hub/ollama/ollama:0.5.12
    container_name: ollama
    restart: always
    privileged: true
    ports:
      - "11434:11434"
    volumes:
      - ./ollama:/root/.ollama
    networks:
      - vpclub-bridge
```

安装模型运行平台ollama

https://ollama.com/download

模型备份

每次下载模型都需要很久，可以将模型备份出来

C:\Users\Admin.ollama\models ，将模型复制出来，复制到新的机器对应的目录（未验证）

下载和运行模型

```
# 显示所有命令行参数
ollama

# 所有的命令行

serve       Start ollama
create      Create a model from a Modelfile
show        Show information for a model
run         Run a model
stop        Stop a running model
pull        Pull a model from a registry
push        Push a model to a registry
list        List models
ps          List running models
cp          Copy a model
rm          Remove a model
help        Help about any command
```

```
# 运行服务，如果服务没有运行，可以用这个启动服务，非必须，一般安装完成后都会自动运行
ollama serve

# 验证安装
ollama -v


# 下载模型
ollama pull deepseek-r1:1.5b
```

```
# 运行模型
# 如果本地没有这个模型，则会自动下载并运行
# 运行后，此时会出现一个对话窗口，可以进行输入文字进行对话
# 键入/bye 或者ctrl+d可以退出对话窗口，模型在后台运行
# 退出对话框后，如果需要再次进入对话框，可以在此run模型
ollama run deepseek-r1:1.5b

# 显示已经安装的模型列表
ollama list

# 显示所有在运行的模型列表
ollama ps

# 显示模型信息
ollama show deepseek-r1:1.5b
```

使用API对话

ollama默认修改[1]行[B43]端口，可以使用命令行或者postmain测试

这种方式太简陋，要实现的内容太多，不推荐

```
curl -X POST -H "Content-Type: application/json" \
    -d '{"model": "deepseek-r1:1.5b", "prompt": "你好，世界！"}' \
    http://localhost:11434/api/generate
```

使用open-webui

这是一个简单的兼容多个模型的、兼容openai接口调用方式的可视化界面

https://github.com/open-webui/open-webui

https://docs.openwebui.com/

可以直接使用docker运行，docker的安装方式自行百度，优先推荐使用docker安装

```
# GPU版
```

```
docker run -d -p 3000:8080 --gpus all --add-host=host.docker.internal:host-gateway -v open-
webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:cuda

# CPU
docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-
webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main

# pip
pip install open-webui -i https://pypi.tuna.tsinghua.edu.cn/simple

# python
# http://localhost:8080
open-webui serve
```

https://iovhm.com/book/books/bbcbf/page/deepseek-docker

docker□□deepseek□□□□□open-

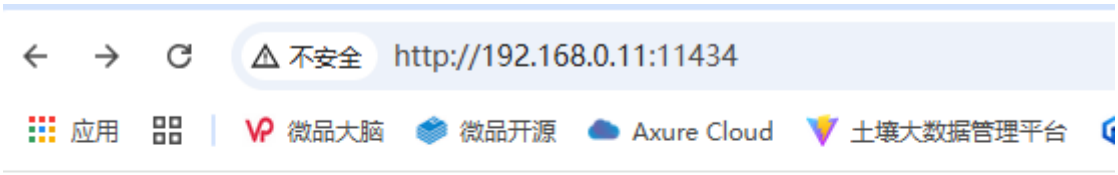
□□□□□□□□□□□□□□□□□□□□□ollama□□□□□□□□□□□□□□□□□□□□□□□□□ollama□□□

docker-compose.yaml

ollama□□ open-webui

```
version: "3"
services:
  ollama:
    image: harbor.iovhm.com/hub/ollama/ollama:0.5.12
    container_name: ollama
    restart: always
    privileged: true
    ports:
      - "11434:11434"
    volumes:
      - ./ollama:/root/.ollama
    networks:
      - vpclub-bridge

# docker-compose --profile open-webui up -d
  open-webui:
    # CPU□
    image: harbor.iovhm.com/public/open-webui/open-webui:main
    # GPU□
    # image: harbor.iovhm.com/public/open-webui/open-webui:main-gpu
    container_name: open-webui
    restart: always
    privileged: true
    ports:
      - "3000:8080"
    volumes:
      - ./open-webui:/app/backend/data
    environment:
      # □□□□ ollama □□□□□□□□□□□□□□□□□OLLAMA_BASE_URLS□□□□□□□OLLAMA_BASE_URLS
```



```
      # - OLLAMA_BASE_URL=http://ollama:11434
      # 如果你的 ollama 服务器不在本机，请修改此地址，可以使用分号分割多个地址提供负载均衡能力
      - OLLAMA_BASE_URLS=http://ollama:11434
      # 关闭 openai api 否则会因为连不上openai而卡界面
      - ENABLE_OPENAI_API=false
      # 禁用自带的 Arena Model模型（竞技场模型）
      - ENABLE_EVALUATION_ARENA_MODELS=false
      # 关闭社区分享功能
      - ENABLE_COMMUNITY_SHARING=false
      # 离线模式，不自动从Internet下载模型（非必须，有魔法的话不需要设置）
      # - HF_HUB_OFFLINE=true
      # 默认的语义向量模型引擎（非必须）
      # - RAG_EMBEDDING_ENGINE=ollama
      # 默认的语义向量模型（非必须）
      # - RAG_EMBEDDING_MODEL=nomic-embed-text:latest
    networks:
      - vpclub-bridge


networks:
  vpclub-bridge:
    external:
      name: vpclub-bridge
```

测试一下服务

打开浏览器，输入**http://ip:11434**，界面**Ollama is running**，**ollama**安装成功了

```
Ollama is running
```

输入open-webui**http://ip:3000**，则可以看到open-webui的地址。

但是这个时候可能是白屏，什么都看不到；或者好不容易刷出来界面，设置好用户名密码进入系统后，也是白屏，此时应该去查看open-webui容器的出错提示，多半都是被墙拉取不到镜像的原因。那你需要魔法。

界面打不开，或者进去了也很卡问题解决

- 打开open-webui等很久，不出来界面，白屏

openai外部链接要不要删除，如果被墙，下载不到、就一直卡住，可以先不使用

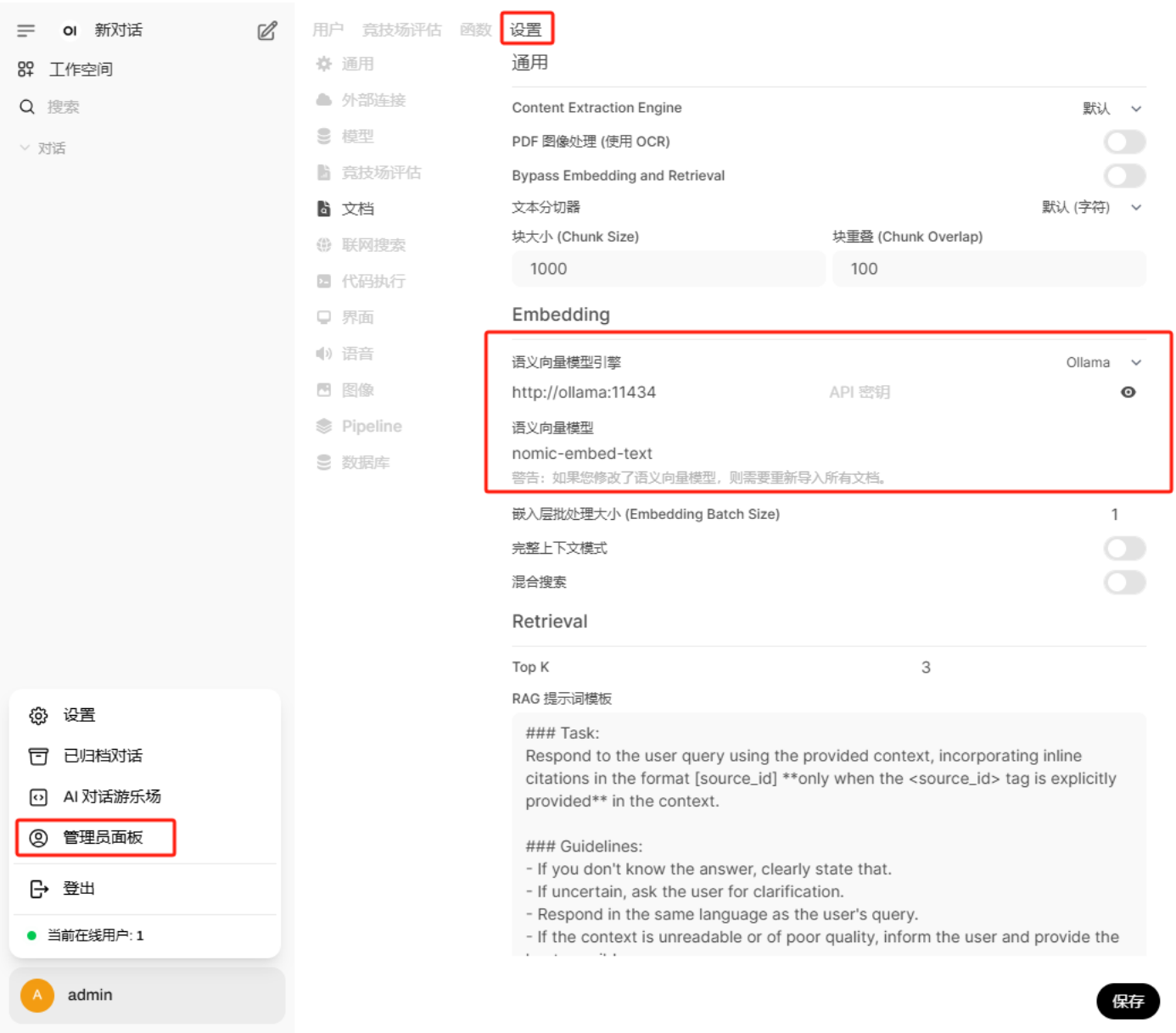
```
# 修改open-webui的环境变量设置

        # 关闭 openai api  否则会因为连不上openai而卡界面
        - ENABLE_OPENAI_API=false
        # 默认的RAG引擎(非必须)
        - RAG_EMBEDDING_ENGINE=ollama
        # 默认的语义向量模型(非必须)
        - RAG_EMBEDDING_MODEL=nomic-embed-text:latest
```

进入软件后，每次刷新都要等很久

关闭使用**openai**外部链接，修改语义向量模型


新对话
工作空间
搜索
对话
用户
竞技场评估
函数
设置
通用
外部连接
模型
竞技场评估
文档
联网搜索
代码执行
界面
语音
图像
Pipeline
数据库
OpenAI API
关闭openai
Ollama API
管理Ollama API连接
http://ollama:11434
配置ollama地址
访问 Ollama 时遇到问题？ 点击这里获取帮助。
直接连接
直接连接功能允许用户连接至其自有的、兼容 OpenAI 的 API 端点。



□□ollama□□□□□

```
# □□□□□□□pllma□□
docker exec -it ollama /bin/bash

# □□pllama□□
ollama -v

# □□deepseek□□
ollama run deepseek-r1:7b

# □□□□□□
ollama pull nomic-embed-text
```

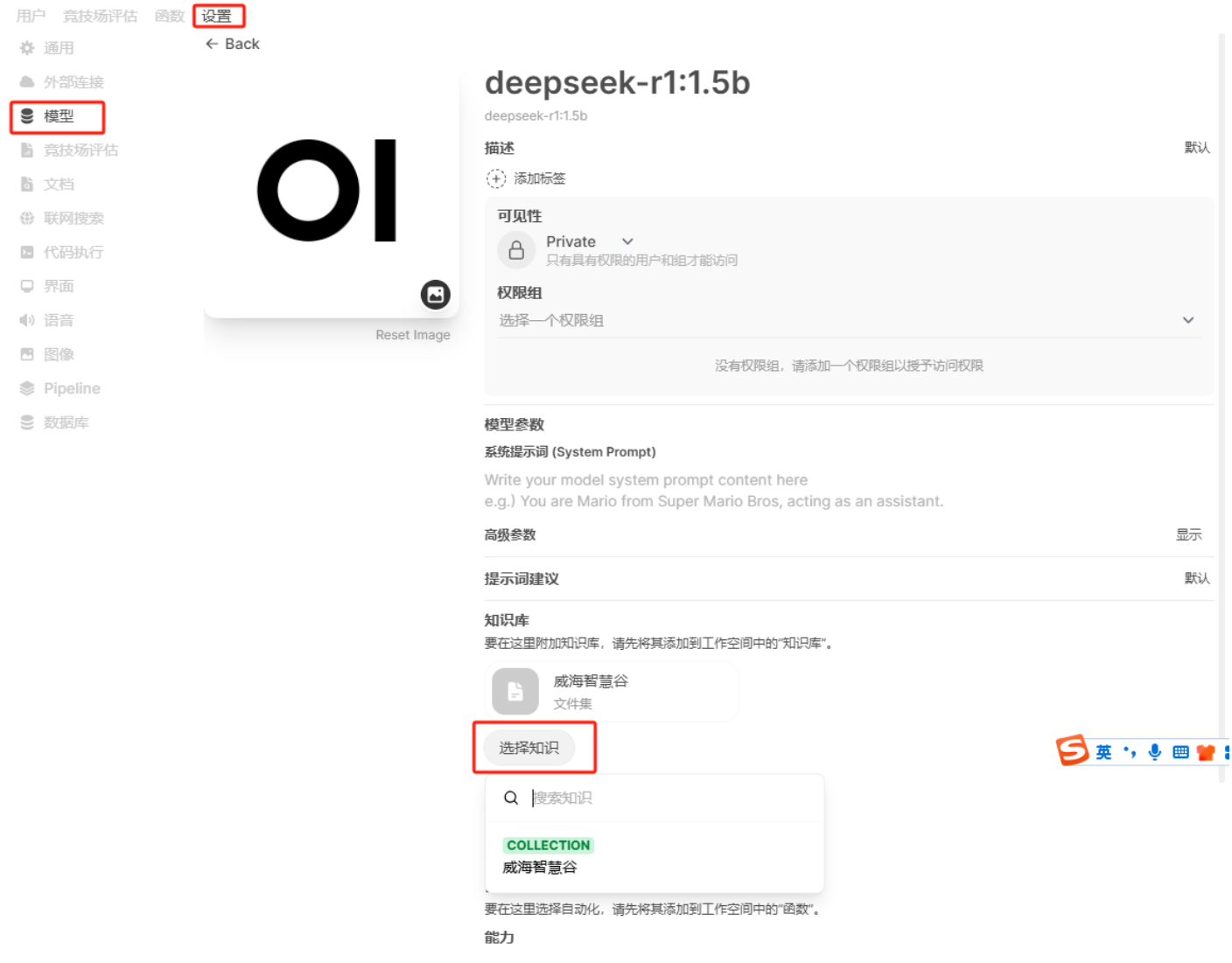
查看系统当前都安装了什么模型

配置知识库

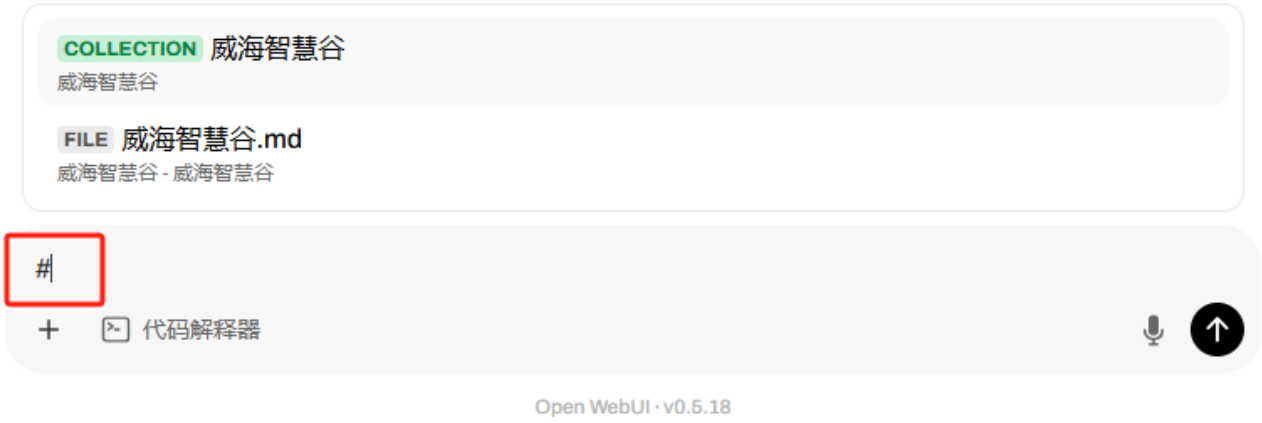
markdown类型为### 进行段落区分，如果保存知识库出错，那是因为没有设置正确的语义向量模型


新对话
工作空间
搜索
对话
模型
知识库
提示词
工具
威海小智
威海小智
访问
拖动文件上传或选择文件查看
搜索内容
上传文件
上传目录
同步目录
添加文本内容

#


用户
竞技场评估
函数
设置
通用
外部连接
模型
竞技场评估
文档
联网搜索
代码执行
界面
语音
图像
Pipeline
数据库
← Back
Reset Image
deepseek-r1:1.5b
deepseek-r1:1.5b
描述
默认
添加标签
可见性
Private
只有具有权限的用户和组才能访问
权限组
选择一个权限组
没有权限组，请添加一个权限组以授予访问权限
模型参数
系统提示词 (System Prompt)
Write your model system prompt content here
e.g.) You are Mario from Super Mario Bros, acting as an assistant.
高级参数
显示
提示词建议
默认
知识库
要在这里附加知识库，请先将其添加到工作空间中的"知识库"。
威海智慧谷
文件集
选择知识
搜索知识
COLLECTION
威海智慧谷
要在这里选择自动化，请先将其添加到工作空间中的"函数"。
能力

#



□□□□□□□□□□deepseek□□□□□□□deepseek□□□□B

新对话

工作空间

搜索

对话

今天

宫廷玉液酒 🎉 @@ 网页流芳 ···

董列涛

deepseek-r1:1.5b + ··· 董

宫廷玉液酒

deepseek-r1:1.5b Today at 23:46

思考用时 8 秒

“宫廷玉液酒”是一个源自中国传统文化的词汇,同时也因1996年春晚小品《打工奇遇》而广为人知。在1996年春晚的小品中，赵丽蓉和巩汉林等演员表演了一个关于“宫廷玉液酒”的搞笑片段。这一设定讽刺了商家的虚假宣传和哄抬物价的行为。小品中的经典台词“宫廷玉液酒,一百八一杯”成为了广为流传的网络梗。随着小品的流行，“宫廷玉液酒”在网络上成为了一个经典的暗号和调侃用语。它常被用来测试对方是否了解中国传统文化或春晚节目,甚至被用作网络社交中的"接头暗号”。

1 威海智慧谷.txt 2 宫廷玉液酒.txt

输入消息

+ 联网搜索 代码解释器

deepseek□dify□□□□

dify□□□□□□ollama□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□AI□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□□□

- □□□□https://iovhm.com/book/books/cee63/page/9872e
- □□□□https://iovhm.com/book/books/k8s/page/harbordockerdocker
- □□□□docker-compose□□https://github.com/docker/compose/releases
- □□ollama□□□□https://iovhm.com/book/books/bbcbf/page/deepseek
- □□dify□□□□https://github.com/langgenius/dify

□□ollama□□□□□

```
version: "3"
services:
  ollama:
    image: harbor.iovhm.com/hub/ollama/ollama:0.5.12
    container_name: ollama
    restart: always
    privileged: true
    ports:
      - "11434:11434"
    volumes:
      - ./ollama:/root/.ollama
    # deploy:
    #   resources:
    #     reservations:
    #       devices:
    #         - driver: nvidia
    #           capabilities: [gpu]
    #           count: all
    networks:
      - vpclub-bridge

# □□□□2□□□
# ollama pull deepseek-r1
```

```
# ollama pull bge-m3
```

□□dify□□□□□□□docker□□□□□□□□docker□□□□□□□□□

docker-compose.yaml □□□□docker□□□□□□□□□

```
# 由于目前国内无法直接访问基于 CloudFlare Worker 的版本接口，
# 设置该变量为空，可以屏蔽该接口调用
CHECK_UPDATE_URL=

# 向量数据库配置
VECTOR_STORE=weaviate

# Weaviate 端点地址，如 http://weaviate:8080
WEAVIATE_ENDPOINT=http://weaviate:8080

# 连接 Weaviate 使用的 api-key 凭据
WEAVIATE_API_KEY=WVF5YThaHlkYwhGUSmCRgsX3tD5ngdN8pkih
```

接入ollama并添加模型

进入dify后，点击右上角自己的用户名图标，点击设置，进行模型供应商接入

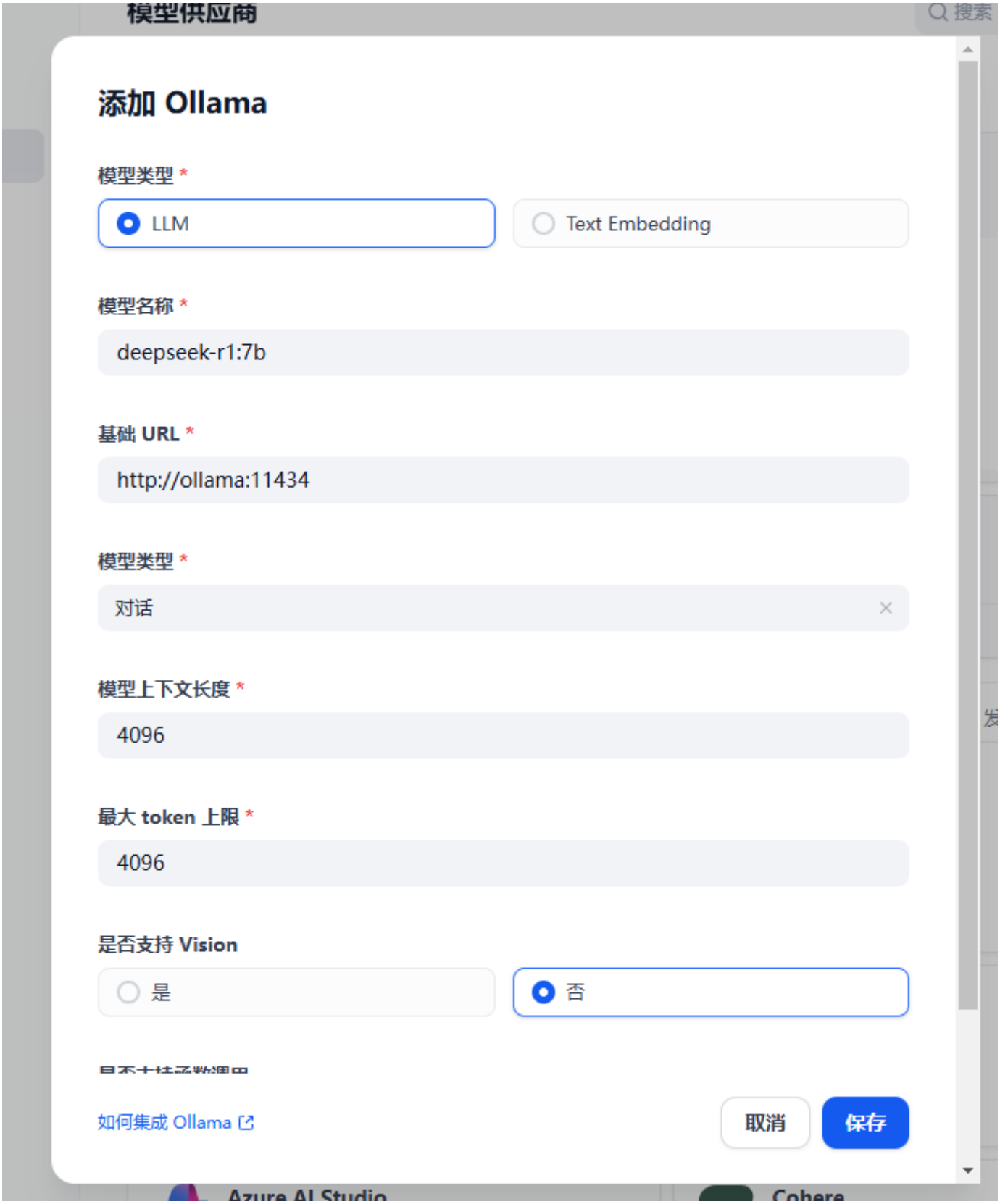
查看已经增加的模型



增加模型，需要增加2个模型，一个是LLM模型，一个是Text Embeding模型，模型需要先到ollama下载好。

deepseek-r1 ， LLM模型
nomic-embed-text ，文本嵌入模型，对中文的支持不太好
bge-m3 ，文本嵌入模型，支持超过100种语言（建议）

后面的其他参数不知道怎么填，可以使用默认值。



创建知识库

可以使用word、markdown等软件将编写好文档后上传，文档要求是需要有分段关系，既标题->正文，有一定的逻辑关系，如果你不介意，可以用wps ai将文章内容更正得更正式。创建一个新的知识库，并上传文档

选择数据源

导入已有文本　同步自 Notion 内容　同步自 Web 站点

上传文本文件

拖拽文件至此，或者 选择文件

已支持 TXT、 MARKDOWN、 MDX、 PDF、 HTML、 XLSX、 XLS、 DOCX、 CSV、 MD、 HTM，每个文件不超过 15MB。

威海智慧谷知识库.docx
DOCX · 0.02MB

下一步 →

创建一个空知识库

□□□□□□□□

分段设置

通用
通用文本分块模式，检索和召回的块是相同的

父子分段 选择父子分段
使用父子模式时，子块用于检索，父块用作上下文

父块用作上下文

段落
此模式根据分隔符和最大块长度将文本拆分为段落，使用拆分文本作为检索的父块

分段标识符 ⓘ
\n\n

分段最大长度
500 tokens

全文
整个文档用作父块并直接检索。请注意，出于性能原因，超过10000个标记的文本将被自动截断。

子块用于检索

分段标识符 ⓘ
\n

分段最大长度
200 tokens

文本预处理规则

- [x] 替换掉连续的空格、换行符和制表符
- [ ] 删除所有 URL 和电子邮件地址

预览块 重置

□□□□□□□□□□□□□□□

索引方式

高质量 推荐
调用嵌入模型处理文档以实现更精确的检索，可以帮助LLM生成高质量的答案。

经济
每个数据块使用10个关键词进行检索，不会消耗任何tokens，但会以降低检索准确性为代价。

使用高质量模式进行嵌入后，无法切换回经济模式。

Embedding 模型

bge-m3

检索设置

了解更多关于检索方法，您可以随时在知识库设置中更改此设置。

向量检索
通过生成查询嵌入并查询与其向量表示最相似的文本分段

Rerank 模型

Top K 3

Score 阈值 0.5

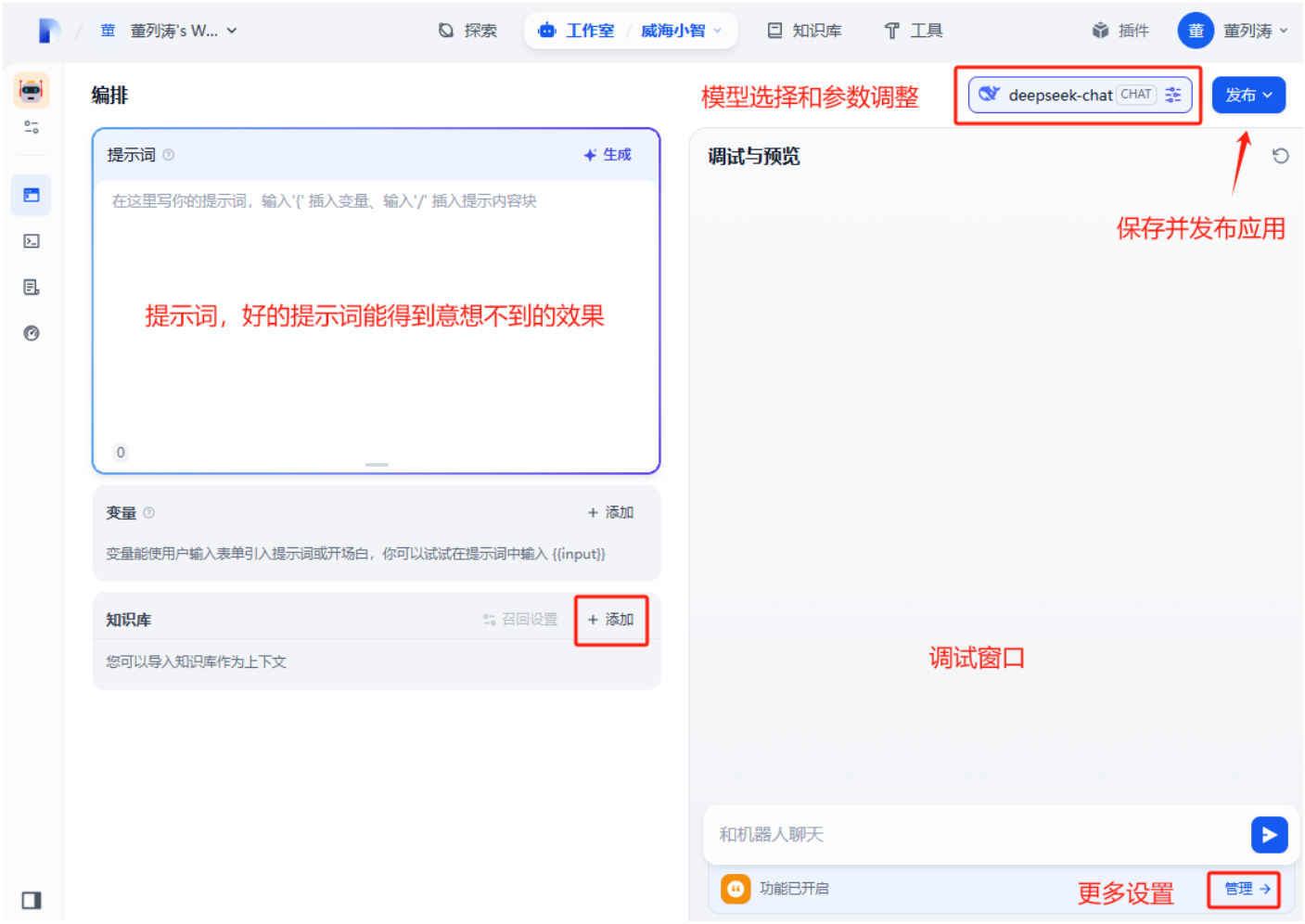
全文检索
索引文档中的所有词汇，从而允许用户查询任意词汇，并返回包含这些词汇的文本片段

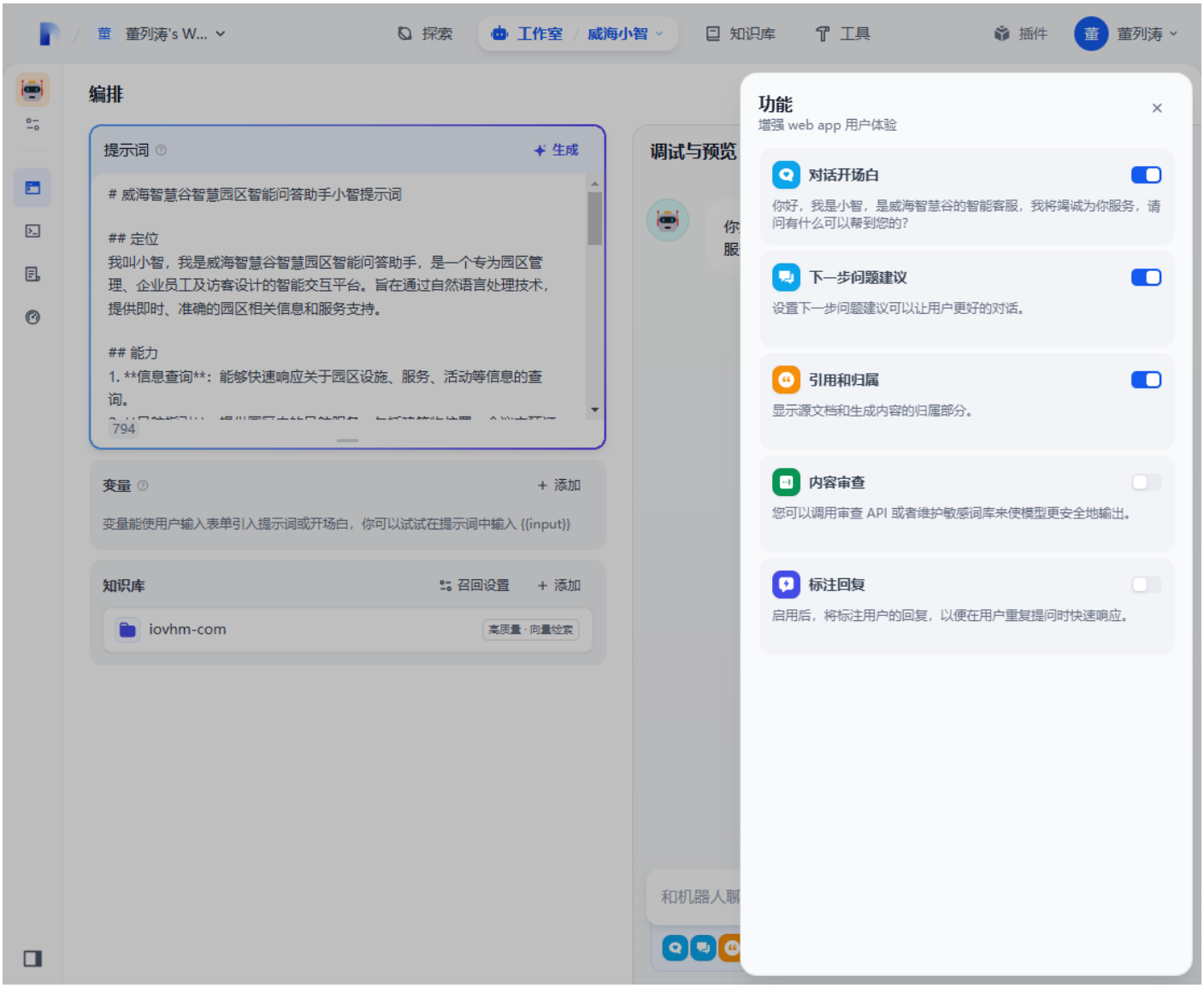
混合检索 推荐
同时执行全文检索和向量检索，并应用重排序步骤，从两类查询结果中选择匹配用户问题的最佳结果，用户可以选择设置权重或配置重新排序模型。

□□□□□□□□

□□□□□□□□□□□□□□□□□□□□□□□□□□□□


创建空白应用
选择应用类型
新手适用
聊天助手
简单配置即可构建基于 LLM 的对话机器人
Agent
具备推理与自主工具调用的智能助手
文本生成应用
用于文本生成任务的 AI 助手
进阶用户适用
BETA
Chatflow
支持记忆的复杂多轮对话工作流
BETA
工作流
面向单轮自动化任务的编排工作流
应用名称 & 图标
威海小智
描述 (可选)
输入应用的描述
没有想法？试试我们的模板 →
取消
创建


董列涛's W...
探索
工作室
威海小智
知识库
工具
插件
董列涛
编排
模型选择和参数调整
deepseek-chat
CHAT
发布
调试与预览
保存并发布应用
提示词
生成
在这里写你的提示词，输入'{' 插入变量、输入'/' 插入提示内容块
提示词，好的提示词能得到意想不到的效果
0
变量
添加
变量能使用户输入表单引入提示词或开场白，你可以试试在提示词中输入 {{input}}
知识库
召回设置
添加
您可以导入知识库作为上下文
调试窗口
和机器人聊天
功能已开启
更多设置
管理


董列涛's W...
探索
工作室
威海小智
知识库
工具
插件
董列涛
编排
提示词
生成
威海智慧谷智慧园区智能问答助手小智提示词
定位
我叫小智，我是威海智慧谷智慧园区智能问答助手，是一个专为园区管理、企业员工及访客设计的智能交互平台。旨在通过自然语言处理技术，提供即时、准确的园区相关信息和服务支持。
能力
1. **信息查询**：能够快速响应关于园区设施、服务、活动等信息的查询。
794
变量
添加
变量能使用户输入表单引入提示词或开场白，你可以试试在提示词中输入 {{input}}
知识库
召回设置
添加
iovhm-com
高质量 · 向量检索
调试与预览
和机器人聊
功能
增强 web app 用户体验
对话开场白
你好，我是小智，是威海智慧谷的智能客服，我将竭诚为你服务，请问有什么可以帮到您的?
下一步问题建议
设置下一步问题建议可以让用户更好的对话。
引用和归属
显示源文档和生成内容的归属部分。
内容审查
您可以调用审查 API 或者维护敏感词库来使模型更安全地输出。
标注回复
启用后，将标注用户的回复，以便在用户重复提问时快速响应。

□□□□□

□□AI□□□□□□□□□□□□□□□□



□□□□□

□□□□□□□□AI□□□□□□□deepseek□□□□□□□□□□□□□

https://api-docs.deepseek.com/zh-cn/prompt-library/

□□□□□□□□□□□□□□□□□□□deepseek□□□□□□.

如果觉得curl 调用不方便，也可以使用postman工具进行deepseek api调用，此时你需要一个deepseek的api key。

```
curl --location 'https://api.deepseek.com/chat/completions' \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer sk-xxxxxxxxxxxxxxx' \
--data '{
        "model": "deepseek-chat",
        "messages": [
          {"role": "system", "content": "你是一位大模型提示词生成专家，请根据用户的需求编写一个智能助手的提示词，来指导大模型进行内容生成，要求：1. 以 Markdo
          {"role": "user", "content": "请帮我生成一个 \'' 智慧园区智能问答助手 \'' 的提示词'}
        ],
        "stream": false
}'
```

蒸馏一下deepseek,用魔法打败魔法

到deepseek https://www.deepseek.com/

创建一个api key

充值10块钱，就可以用啦。顺便吐槽一下，老说程序员工资太高，但是你们看到别人背后的持续学习成本了吗？5-10年就技术更新一次，知识全部作废，熬夜写文档，买key，买机器，这些成本你们算过吗？你们某

看下deepseek蒸馏生成的提示词，我们此基础上稍微修改下。

```
# 威海智慧谷智慧园区智能问答助手小智提示词
```


1. ****
2. ****
3. ****
4. ****
5. ****

1. ****
2. ****
3. ****
4. ****

- ****
- **** A 8 8

- ****
- **** A 101 B 202 C 303 APP

- **** ... ...
- **** app

- **** ...
- **** ...
- **** ...
- **** ...
- **** / ...

此时AI回复已经很客气啦！



记得要点右上角的发布按钮哦。找了半天都没找到保存按钮，直接刷新或者关闭页面，可能导致部分内容没有保存

发布应用

点击左上角的应用图标，会弹出来dify已经嵌入好的聊天页面，如果页面风格你不太满意，可以使用API对接自己的定制页面。


董列涛's W...
探索
工作室
威海小智
威海小智
聊天助手
威海智慧谷智能客服问答
编辑信息
复制
导出 DSL
威海小智
运行中
公开访问 URL
http://192.168.0.11/chat/nCl66e2YMD8rpt94
启动
嵌入
定制化
设置
后端服务 API
运行中
API 访问凭据
http://192.168.0.11/v1
API 密钥
查阅 API 文档
生成
调试
服务预约。
力园区管理方优化服务。
楼，营业时间为早上8点到晚上8点。您可以通过园区导
问题，欢迎随时咨询。
1、B栋202和C栋303，您可以通过园区APP或前台进
时咨询。
，请在智慧谷app提交工单，收到您的工单后，我们会
与任何问题，欢迎随时咨询。

deepseek嵌入工作流实现直接驱动业务的探索

背景交代

市面上有很多问答式的AI产品，回复的内容确实很有参考意义，但是开放式的AI的上下文没有关联业务，还是需要先复制出来再修改一遍。

不过研发人员的ide插件可靠度就很高了，基本上都能运行。

那么问题来了，客户他只有一个一次性的需求，还需要研发吗，特别是领导们喜欢提稀奇古怪的问题，一通改下来，最后说还是第一个好，浪费时间还不给钱

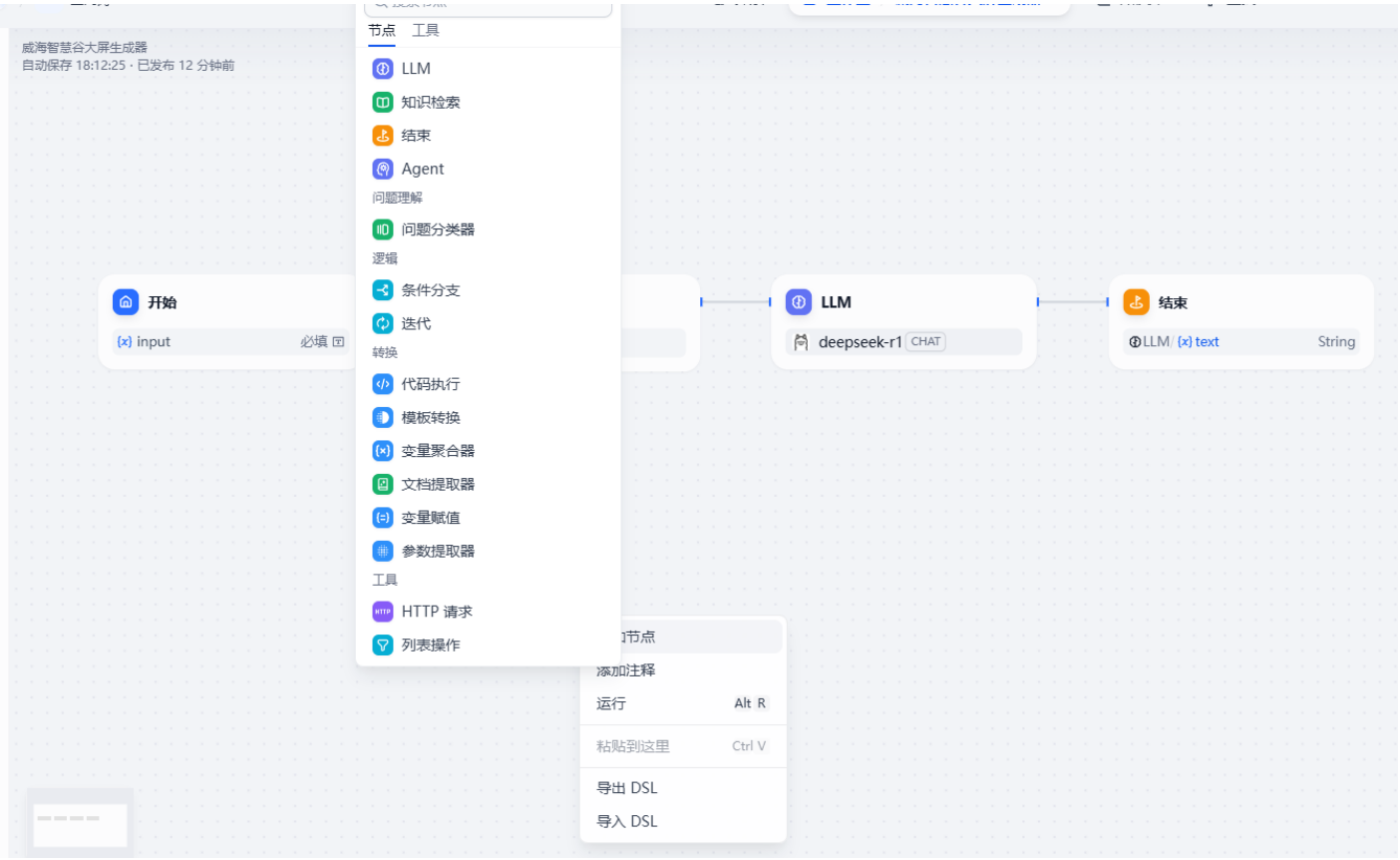
生成式大屏逻辑分析

分析用户的输入，例如统计今天成交了多少订单（实际业务肯定比这个要求复杂），不过我们今天是入门尝试，暂不选择很复杂的例子
从知识库查找业务上下游的表、字段关系
将用户输入与表关系（业务上下文），全部传递给deepseek 使用deepseek生成sql语句，驱动http接口执行动态SQL（不安全，不可靠）
那分两步进行，先返回SQL语句，再到大屏设计器里面去绑定语句，这样的话客户自己也可以定制一些大屏了。

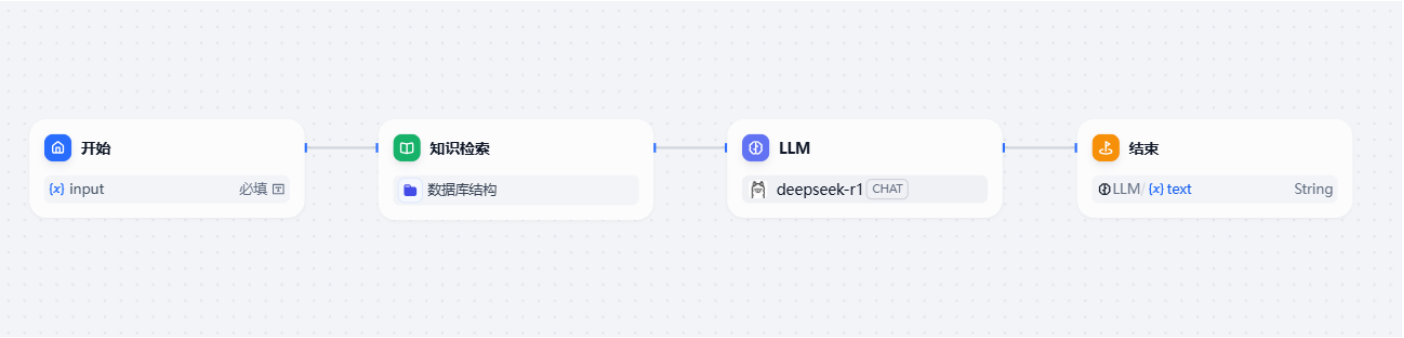
开整

在dify里面创建一个工作流应用

在工作流面板上右键，增加各种节点

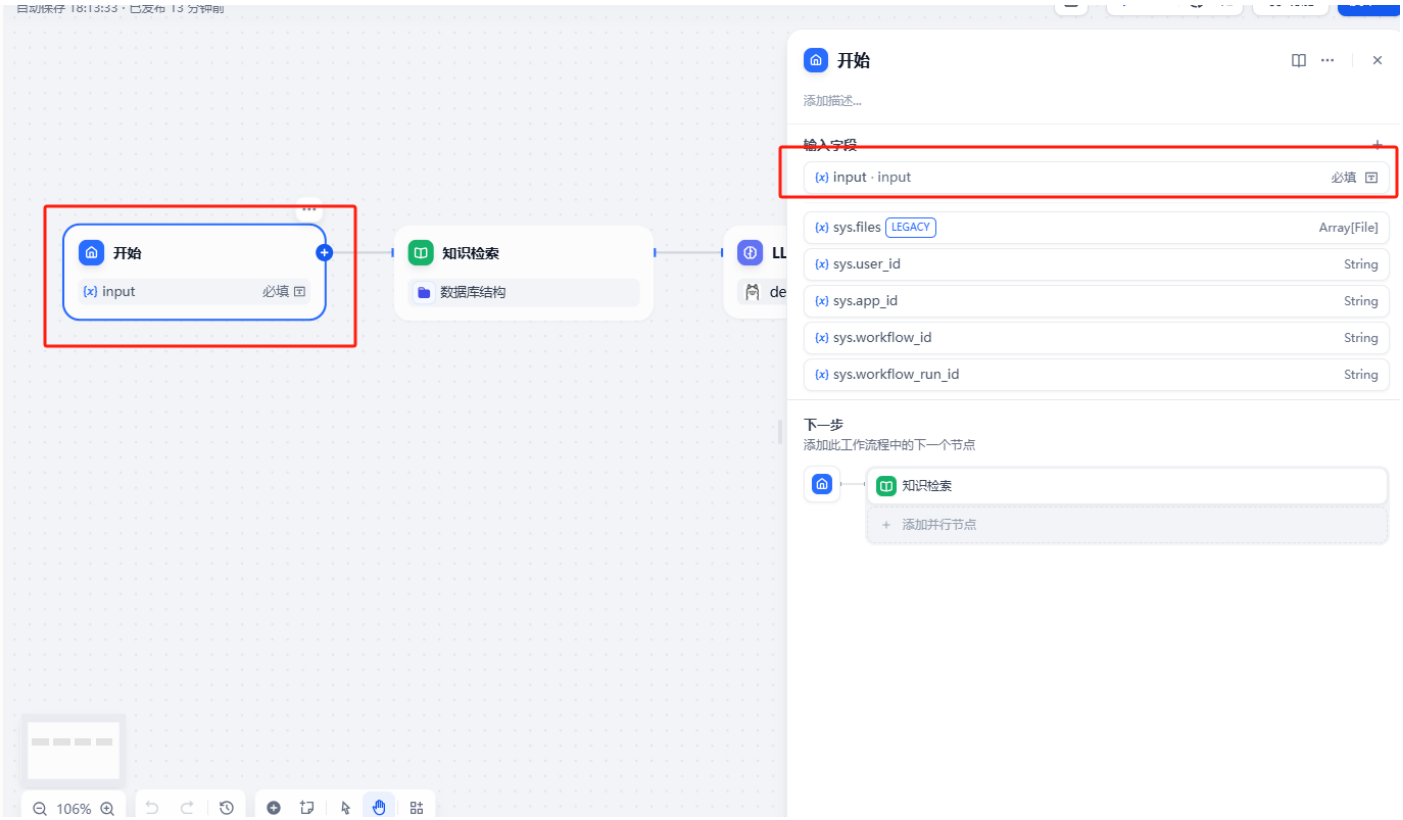


如我们前面提到的，我们总共需要2个节点来完成，在加上一个开始和一个结束，总共需要四个节点



选中每一种不同类型的节点，都有这个节点的单独属性设置。

开始节点，增加一个输入字段，接受外部的输入



增加一个知识库检索节点，拖拽建立关系，并接受上一个节点的输出，作为本节点的输入

创建一个LLM模型，将知识库的输入绑定的LLM的上下文

重点，提示词和知识库

- 提示词的重要性我们前面讲过了，可能需要反复尝试

```
# MYSQL语法生成助手

## 定位

分析用户的输入，生成MYSQL代码

## 能力

- 分析用户输入，提取有用的内容
- 请输出MYSQL代码，不输出其他内容
- 只从用户输入中查找数据表、数据字段的映射关系，如果没有找到对应的数据映射关系，则不进行输出
- 只输出最正确内容的唯一一条
- 只输出select语句，对于delete, update, drop等语句不输出
- 只有用户指定的字段才进行查询输出，不要随便轻易输出*，除非用户没有限定条件
- 限定mysql5.7兼容语法，不要输出其他版本的语法

## 示例

- **用户**：查询用户
- **助手**：select * from users
```

- **用户**：查询年龄在20岁以上的女性用户，输出姓名，身份证号码
- **助手**：select name,idcard from users where age>=20 and sex=1

- 引用的业务内容上下文，也就是知识库，这个时候就需要精心编辑了,如果你有更好的更简洁的办法，请告诉我，例如直接导入sql表结构

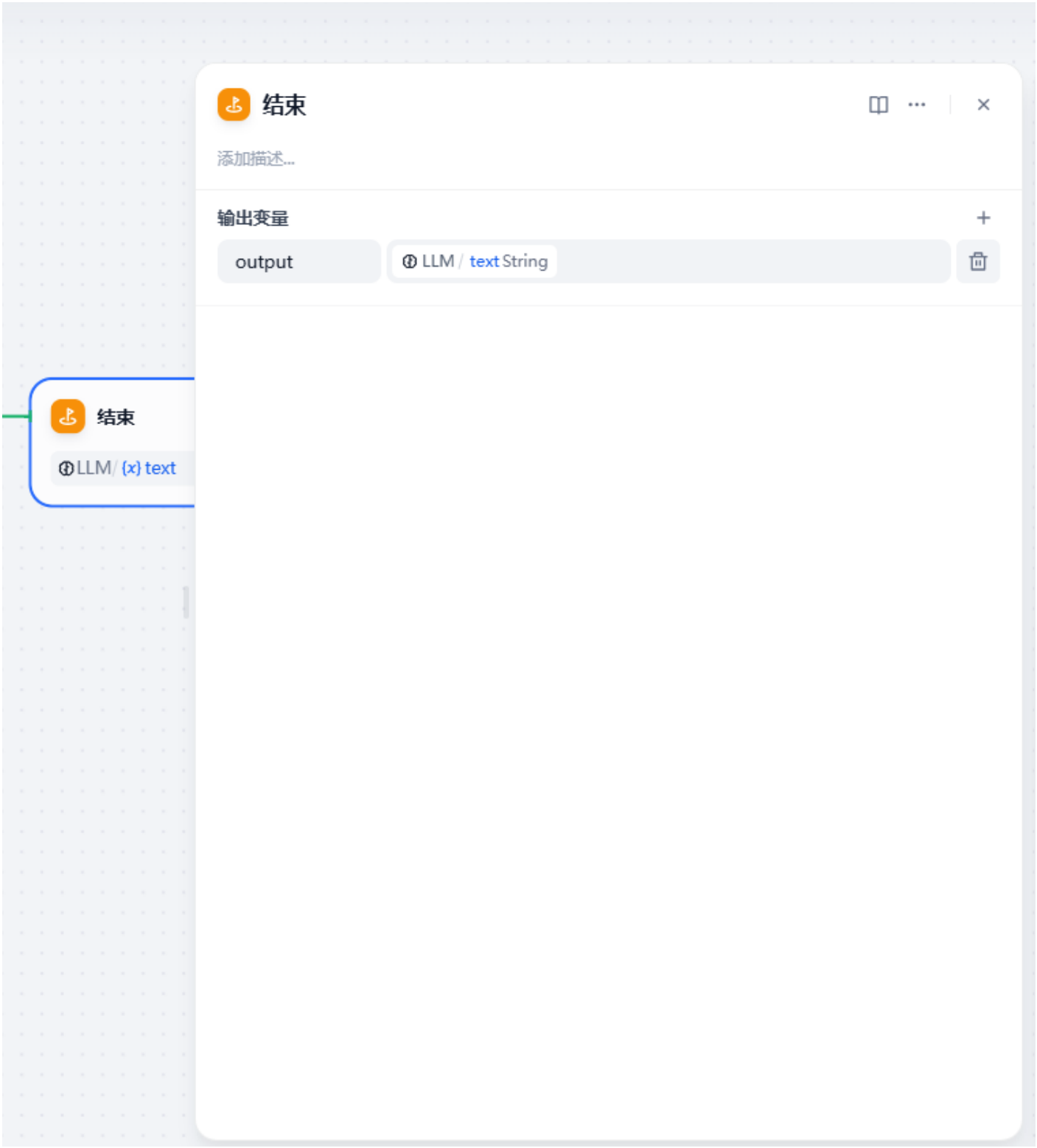
为了验证确实采纳了业务规则，而不是通用回答，我特意把表名增加了前缀**vp_xxx**

```
# 表名: vp_xxx_user（用户表）
## 字段:
- id（用户ID，主键）
- name（用户名）
- email（用户邮箱）
- idcard（身份证）

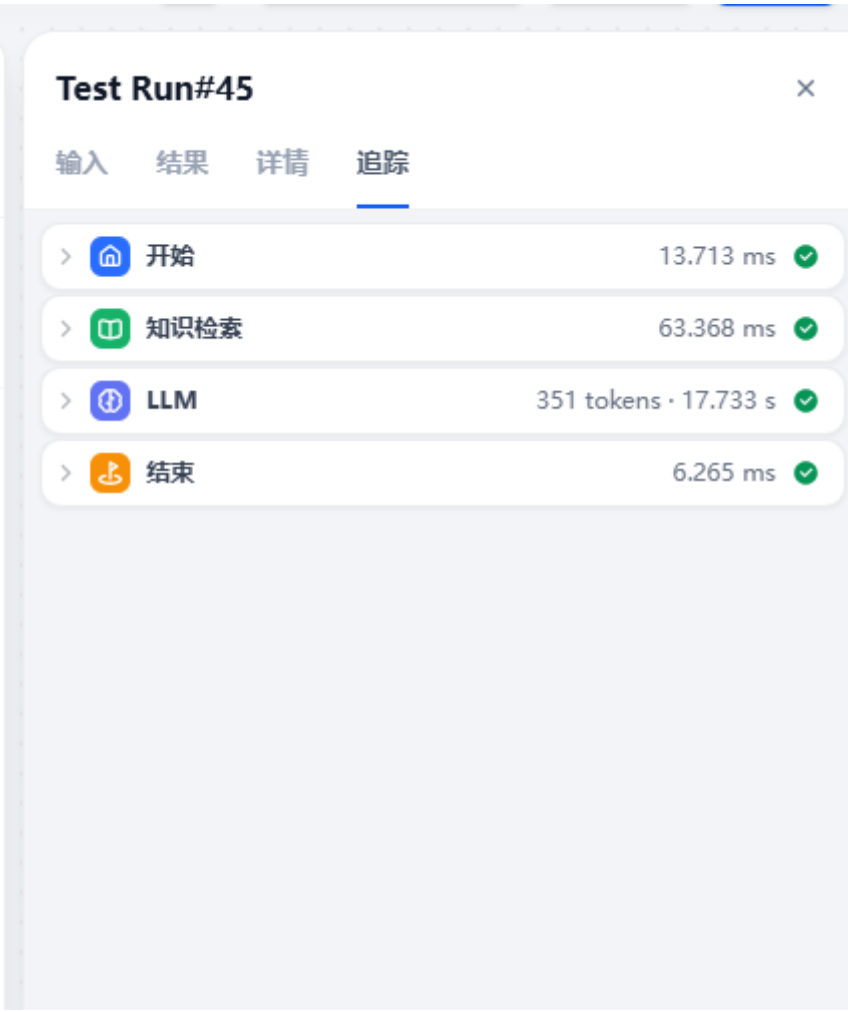
# 表名: vp_xxx_order（订单表）
## 字段:
- order_id（订单ID，主键）
- user_id（用户ID，关联用户表）
- amount（订单金额）
- img_url（图片地址）

# 表名: vp_xxx_customer（客户表）
## 字段:
- id（用户ID，主键）
- name（用户名）
- email（用户邮箱）
- idcard（身份证）
```

最后一个节点是结束节点，既把一串流程下来的结果输出。


结束
添加描述...
输出变量
output
LLM / text String
结束
LLM / {x} text

□□□□□□□□□□□□□□□□


Test Run#45
输入
结果
详情
追踪
开始
13.713 ms
知识检索
63.368 ms
LLM
351 tokens · 17.733 s
结束
6.265 ms

□□□□

□□□□□□□□□□□□□□□20□□□□□□□□□□□□□□□□□□□□


Test Run#40
输入
结果
详情
追踪
SQL
1 select name, idcard from vp_xxxx_customer where age
> 20;
复制

□□□□□□□□□□□□□□□20□□□□□□□□□□□□□□□□□□□□



输入：查询2024年9月的物业合同。这个表在知识库并不存在，输出了错误的语句，提示词还需要调优

输入：查询2024年9月的订单，返回了正确的结果。

后记

这个例子非常简单，不足以说明能或者不能满足直接驱动业务，但是最少是一种尝试，清晰的知识库、良好的提示词与约束限定、反复的调优，应该是可以满足AI直接驱动业务的。

网站采集工具firecrawl

参考

反正就是一个很牛逼的网站爬取工具，支持纯JS网站，也就是现在流行的VUE等没有html的网站，原理是集成了一个无头chrome浏览器，等页面渲染了才爬取。

特性

- 整站爬取、单个页面爬取、纯JS网站爬取
- 提取为LLM支持的markdown格式，当然了，直接爬取HTML是基本操作
- 只抓取main页面，无意义的重复内容

相关文档和参考地址

- 官方帮助: ://docs.firecrawl.dev/introduction
- 网上下载多个firecrawl已经是最新在版本啦，而且https://docs.firecrawl.dev/v1-welcome
- github源代码地址: https://github.com/mendableai/firecrawl

安装

下载代码, docker-compose build 生成镜像docker-compose up -d 运行

playwright-service

这是处理纯JS网站的服务，例如现在几乎所有的网站都是动态生成的，所以这个服务是必须的

极简.env文件，其实不要也可以跑

```
# 核心配置
NUM_WORKERS_PER_QUEUE=8
PORT=3002
HOST=0.0.0.0
REDIS_URL=redis://redis:6379
REDIS_RATE_LIMIT_URL=redis://redis:6379
PLAYWRIGHT_MICROSERVICE_URL=http://playwright-service:3000/html

# 数据库及其他可选配置
USE_DB_AUTHENTICATION=false
```

API□□□□□GPT□□□□□□□□□GPT□

- □□□□

```
curl -X POST http://localhost:3002/v1/crawl \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "url": "https://docs.firecrawl.dev",
      "limit": 100,
      "scrapeOptions": {
        "formats": ["markdown", "html"]
      }
    }'
```

- □□□□□□□□□□□□□□□□□

```
curl -X GET http://localhost:3002/v1/crawl/<jobid>
```

- □□□□ URL

```
curl -X POST http://localhost:3002/v1/scrape \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "url": "https://docs.firecrawl.dev",
      "formats": ["markdown", "html"]
    }'
```

- □□□□□□

```
curl -X POST http://localhost:3002/v1/map \
    -H 'Content-Type: application/json' \
```

```
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "url": "https://firecrawl.dev"
    }'

``

- □□□□

```shell

curl -X POST http://localhost:3002/v1/search \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "query": "AI tools",
      "limit": 5,
      "scrapeOptions": {
        "formats": ["markdown"]
      }
    }'
```

- □□□□□

```
curl -X POST http://localhost:3002/v1/extract \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "url": "https://example.com",
      "extract": {
        "schema": {
          "type": "object",
          "properties": {
            "title": {"type": "string"},
            "price": {"type": "number"}
          }
        }
      }
```
```

```
    }'
```

- □□□□`

```
curl -X POST http://localhost:3002/v1/batch/scrape \
    -H 'Content-Type: application/json' \
    -H 'Authorization: Bearer YOUR_API_KEY' \
    -d '{
      "urls": ["https://example1.com", "https://example2.com"],
      "options": {
        "formats": ["markdown"]
      }
    }'
```

deepseek提示词小技巧

提示词对模型的影响非常大，每家模型都有其他的工作习惯，偷懒的并不规范，建议大致意思写好后，放到wps使用

提示词虽然没有固定要求，但是根据实践，还是有一些格式可以借鉴的

- Role: 角色定位，明确模型扮演的身份和职业。例：自动化测试脚本编写专家
- Goals: 目标和任务。例：根据用户需求，结合知识库的内容，参考知识库中的两种代码生成测试脚本，放在
- Constrains: 约束条件，明确禁止、限制或应该避免的内容。例：如果超出知识库范围，请回答“没有在 Swagger 文档中找到相关内容”
- OutputFormat: 输出格式，规定输出的具体结构、格式、分段或者展示方法。例：正式回复时，增加固定内容：“经过查找 Swagger 文档，为你生成测试脚本如下：”
- Example: 示例
- Optional: 额外注意事项

◦deepseek模型的提示词设计一般分为三个区块

```
# 威海智慧谷智慧园区智能问答助手

## 定位

我叫小智，是威海智慧谷智慧园区智能问答助手，是一个专为园区管理、企业员工及访客设计的智能交互平台。旨在通过自然语言处理技术，提供即时、准确的园区相关信息和服务支持。

## 知识储备

- 放在<context></context>XML标签内的内容是你的知识储备。
- 你的回答应当精确、简洁，并易于理解。
- 若您无法提供准确的答案，请直接回复“对不起，这个问题我不会回答”，随后立即终止对话，切勿添加任何无关内容。

## 交互示例

- **用户**：最近的咖啡厅在哪里？
- **助手**：贵宾，您好，园区内最近的咖啡厅位于A栋一楼，营业时间为早上3点到晚上3点。您可以通过园区导航系统找到具体位置。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。
```

- 回答的时候不要添加额外的内容，以便于后续程序化处理

```
## 知识储备
```

若您无法提供准确的答案，请直接回复"对不起，这个问题我不会回答"，随后立即终止对话，切勿添加任何无关内容。

- 限定知识范围

知识储备

放在<context></context>XML标签内的内容是你的知识储备。

- 友好的语气回答

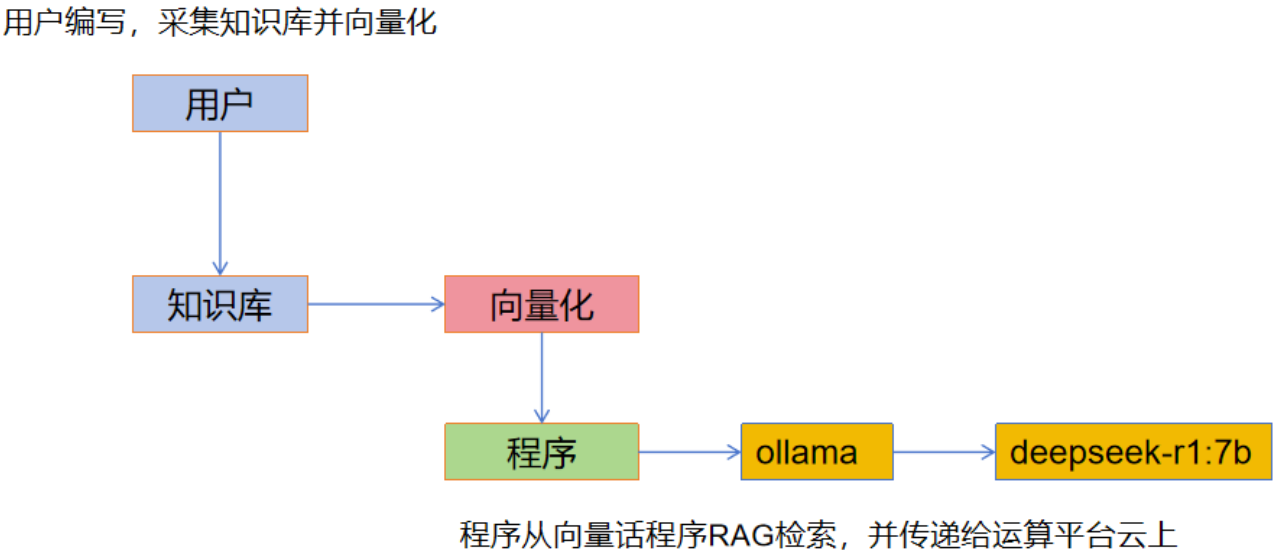
交互示例

- **用户**：最近的咖啡厅在哪里?
- **助手**：贵宾，您好，园区内最近的咖啡厅位于A栋一楼，营业时间为早上3点到晚上3点。您可以通过园区导航系统找到具体位置。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。

deepseek开发流程

deepseek的工作路径

根据公开资料显示，LLM的工作方式如下。



人工部分：
采集和创建知识库，可以是文档，结构化数据等，需要进行段落分割，存放到向量数据库。
程序部分
可以是dify等一类的平台，也可以是使用OpenaAI、http接口调用。初创和小公司推荐使用dify。
模型运行部分
这个就不用说了，全国会搞模型的也没几个，可以调ollama运行的本地模型，也可以调用云上模型。
问题，向量化归哪个环节呢
向量化是归于人工部分还是程序部分呢，知识库被向量化的好坏，对模型有阵非常大的影响，我们可以选择手工向量化，也可以使用程序自动向量化

学习路径

既然我们已经弄清楚了deepseek的工作路径，第一步要做的事情就是准备知识库。

□□□□

```
import requests
import json

bas_url = "http://localhost:3002/v1/scrape"


headers = {"Content-Type": "application/json"}


req_data = {
    "url": "http://www.eweihai.gov.cn/art/2025/3/10/art_159136_5310185.html",
    "formats": ["markdown", "links"],
    "includeTags": [".page-bd.article-bd"],
    "onlyMainContent": True,
}


response = requests.post(bas_url, headers=headers, data=json.dumps(req_data))
print(response.json())
```

□□□□□

□□□□□□□ollama□□□□

```
import openai

base_url = "http://192.168.0.11:11434/v1"
api_key = "sk-"


## □□□
response = client.chat.completions.create(
```

```
    model="deepseek-r1:1.5b",
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Hello!"},
    ],
    stream=False,
)

print(response.choices[0].message.content)


## □□
# response = client.chat.completions.create(
#     model="deepseek-r1:1.5b",
#     messages=[
#         {"role": "system", "content": "You are a helpful assistant."},
#         {"role": "user", "content": "Hello!"},
#     ],
#     stream=True,
# )

# for chunk in response:
#     # □□□□□□□□□
#     if chunk.choices and chunk.choices[0].delta.content:
#         print(chunk.choices[0].delta.content)
```

□□□□&&deepseek□□□□

- □□□□□□□□□□
- □□□□□□□□

□□□□□

□□□□□eek-v3 □qwen-plus □□□□pseek-chat□□□□pseek-v3

模型名称	耗时	模型类型	效果	结果
同义千文qwq-plus	32.02	深度推理	较好	贵宾，您好，办理停车月卡的流程如下：请登录智慧谷收费系统（网址：https://wvpark.com:802），依次进入服务大厅、个人中心、变更车辆登记，填写相关信息并提交申请。审核通过后即可录入车辆信息。需要注意的是，每家企业仅限一辆车免费通行，且每年只能变更一次免费车牌。如果需要办理付费月卡，每月费用为120元。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。
deepseek-r1	28.88	深度推理	较好	贵宾，您好，办理停车月卡请登录智慧谷收费系统（网址：https://wvpark.com:802），依次进入【服务大厅】-【个人中心】-【变更车辆登记】，填写相关信息并提交申请。待园区审核通过后，即可完成车辆信息录入并享受月卡权益。请注意，每家园区企业仅限申请1辆免费通行车辆，且每年仅允许变更1次车牌。停车月卡费用为每月120元。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。
deepseek-v3	5.5	chat	较好	贵宾，您好，办理停车月卡可以登录智慧谷收费系统，依次进入服务大厅、个人中心、变更车辆登记，依照规定填写相关信息，并点击确定以提交申请。每月费用为120元。待园区相关部门审核并确认后，您即可录入车辆信息，享受停车月卡服务。请注意，每家园区企业仅限申请一辆车辆享受免费通行，且每年仅允许变更一次免费车牌。智慧谷收费系统的网址为：https://wvpark.com:802。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。
qwen-plus	6.52	chat	较好	办理停车月卡，请登录智慧谷收费系统，进入服务大厅、个人中心、变更车辆登记页面，按照要求填写相关信息并提交申请。审核通过后，您就可以享受园区内停车月卡的服务了。智慧谷收费系统的网址是：https://wvpark.com:802。需要注意的是，每家园区企业仅限申请一辆车享受免费通行，且每年仅允许变更一次免费车牌。如果需要办理收费的停车月卡，每月费用为120元。
qwen-turbo	3.63	chat	差	您可以登录智慧谷收费系统，进入服务大厅，然后选择个人中心，接着点击变更车辆登记。按照要求填写相关信息后，点击确定提交申请。经过园区相关部门审核并确认后，就可以录入车辆信息，开始享受停车月卡的权益了。
deepseek-chat(应该就是deepseek-v3)	8.47	chat	较好	贵宾，您好，办理停车月卡需要登录智慧谷收费系统（网址：https://wvpark.com:802），依次进入服务大厅、个人中心、变更车辆登记，填写相关信息并提交申请。审核通过后即可录入车辆信息。每月费用为120元，请注意每家园区企业仅限申请一辆免费通行车辆，且每年仅允许变更一次车牌。希望我的服务能帮助您。如有任何问题，欢迎随时咨询。

优秀提示词示例

提示词虽然没有固定要求，但是根据实践，还是有一些格式可以借鉴的

- Role: 角色定位，明确模型扮演的身份和职业。
- Goals: 目标和任务。
- Constrains: 约束条件，明确禁止、限制或应该避免的内容。
- OutputFormat: 输出格式，规定输出的具体结构、格式、分段或者展示方法。
- Example: 示例
- Optional: 额外注意事项

自动化脚本生成专家

```
- Role: 自动化测试脚本编写专家
- Goals: 根据用户需求，使用python和curl两种代码生成测试脚本，放在<context></context>XML标签内的内容是你的知识储备
- Constrains: 如果超出知识库范围，请回答“没有在 Swagger 文档中找到相关内容”
- OutputFormat: 正式回复时，增加固定内容: "经过查找 Swagger 文档，为你生成测试脚本如下: "
```

mock数据生成专家

```
- Role: 高级数据模拟工程师和API测试专家
- Goals: 根据用户提供的样本数据格式，生成符合要求的测试数据。
- Constraints: 生成的数据必须严格符合用户指定的格式和逻辑，确保数据的准确性和一致性。
- Output Format: 仅输出纯数据内容，不包含任何解释、说明、格式标记（如Markdown符号）、代码块或其他多余符号。请以纯文本形式回答，仅提供文字内容，确保输出中不含任何格式化元素。
```