

使用教程

平台产品介绍

本平台是一款面向企业级场景的一站式、商用 license 友好的智能体开发平台，致力于为企业提供安全、高效、合规的一站式 AI 解决方案。我们以“技术开放、生态共建”为核心理念，通过整合大语言模型、业务流程自动化等前沿技术，构建了覆盖模型全生命周期管理、MCP、联网检索、智能体快速开发、企业知识库建设、复杂工作流编排等完整功能体系的 AI 工程化平台。平台采用模块化架构设计，支持灵活的功能扩展和二次开发，在确保企业数据安全和隐私保护的同时，大幅降低了 AI 技术的应用门槛。无论是中小型企业快速构建智能化应用，还是大型企业实现复杂业务场景的智能化改造，都能提供强有力的技术支撑，助力企业加速数字化转型进程，实现降本增效和业务创新。

平台操作说明

一、模型管理

用户可通过配置，在平台中接入外部模型。根据模型供应商规范，关联推理 url。支持导入的模型包括 LLM、Embedding 和 Rerank，开启服务后，可用于模型推理。

模型管理

模型供应商: OpenAI-API-compatible

模型名称: 模型导入

模型列表:

- Silicon bge-reranker-v2-m3 (Rerank)
- jina-ranker-m0 (Rerank)
- Silicon bge-m3 embedding (Embedding)
- Qwen通用文本向量-v3 (Embedding)
- deepseek/deepseek-chat-v3-0324:free (LLM)
- qwen_06b (Rerank)
- qwen3-embed-0.6b (Embedding)
- yuanjing-70b-functioncall (LLM)
- deepseek-v3-functioncall (LLM)
- DeepSeek-V1-671B (LLM)
- DeepSeek-V3-0324 (LLM)

... 发布

模型管理

模型供应商: OpenAI-API-compatible

模型名称: 模型导入

供应商选择:

- OpenAI-API-compatible (LLM, Rerank, Embedding)
- 联通元景 (LLM, Rerank, Embedding)

取消 确定

模型列表:

- DeepSeek-V3-0324 (LLM)
- maas-test jina reranker m0 (Rerank)
- maas-test bge-m3 (Embedding)
- maas-test bge-m3 (Rerank)
- maas-test deepseek-r1-distill-qwen-7b (LLM)

2025-03-24 发布
2025-06-04 发布
2025-06-05 发布
2025-06-04 发布

模型管理

模型供应商: 联通元景

模型名称: 模型导入

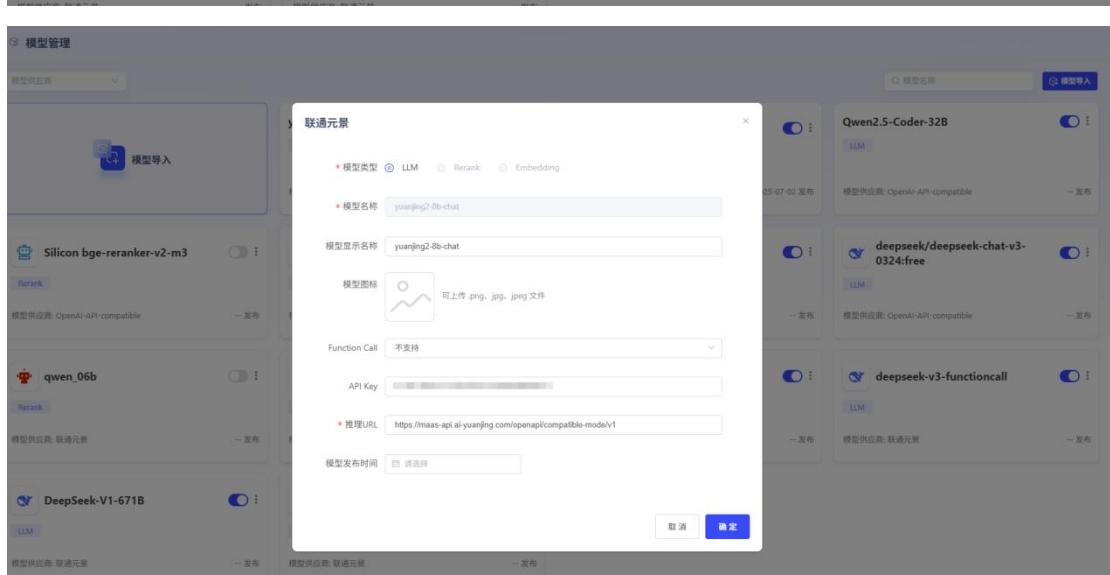
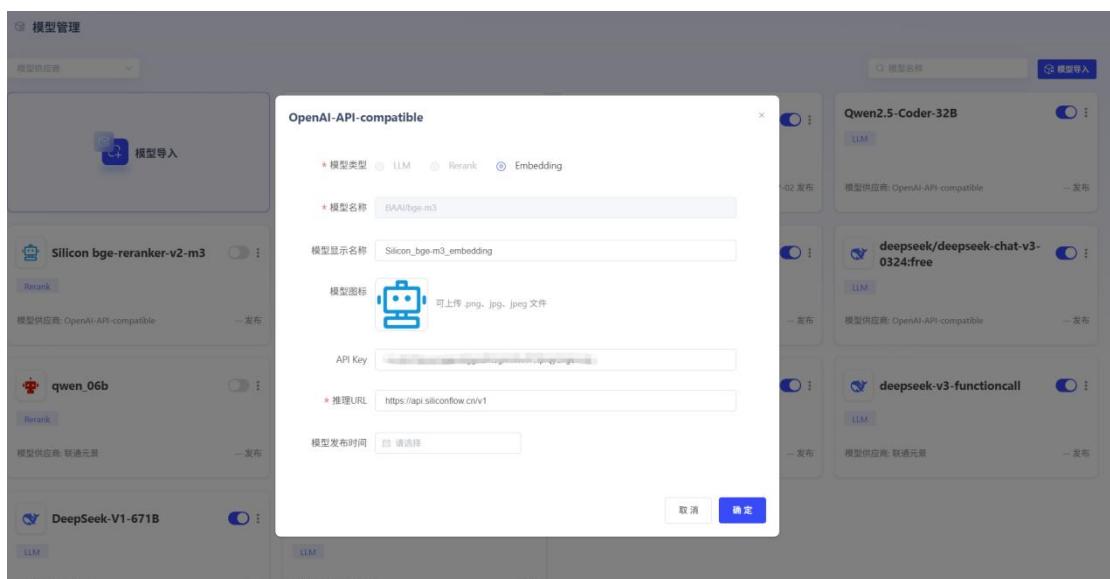
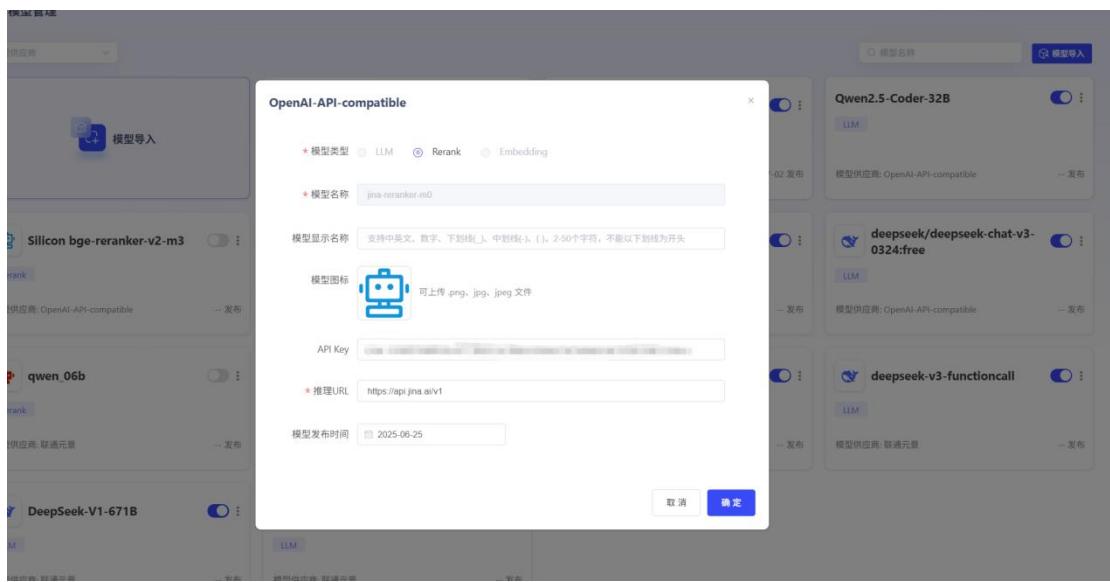
模型导入:

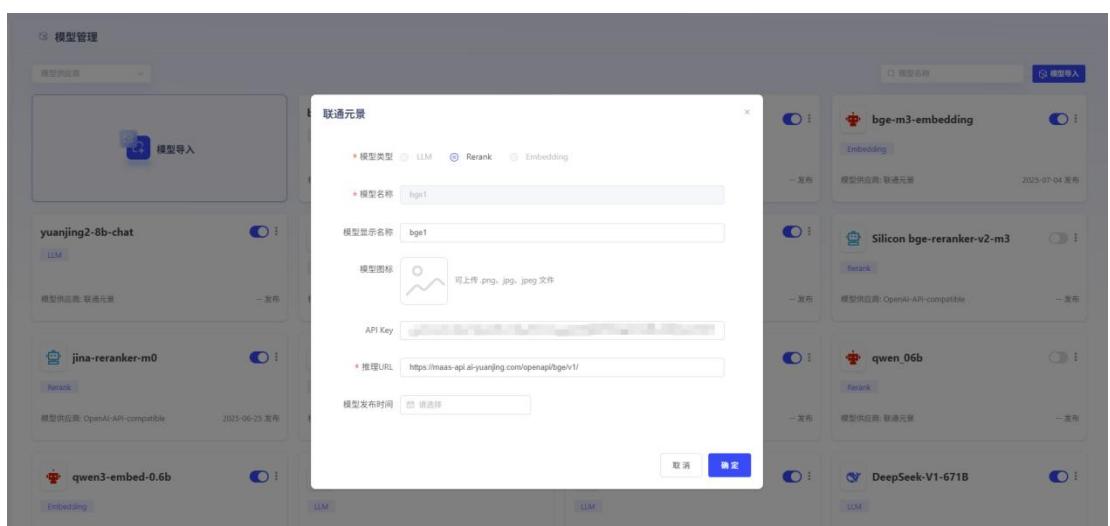
模型列表:

- OpenAI-API-compatible (LLM, Rerank, Embedding)
- Qwen2.5-Coder-32B (LLM)
- deepseek/deepseek-chat-v3-0324:free (LLM)
- deepseek-v3-functioncall (LLM)

2025-07-02 发布
2025-06-04 发布
2025-06-05 发布
2025-06-04 发布

... 请上传 png, jpg, jpeg 文件
不支持
https://api.siliconflow.cn/v1
2025-06-04 请选择





模型导入样例：

```
curl -location 'https://maas.ai-yuanjing.com/openapi/compatible-mode/v1/chat/completions' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer sk-abc*****xyz' \
--data '{
    "model": "yuanjing-70b-chat",
    "messages": [
        {
            "role": "user",
            "content": "你好"
        }
    ]
}'
```

【模型名称】必须为 curl 中可以正确请求的 model；例如

yuanjing-70b-chat

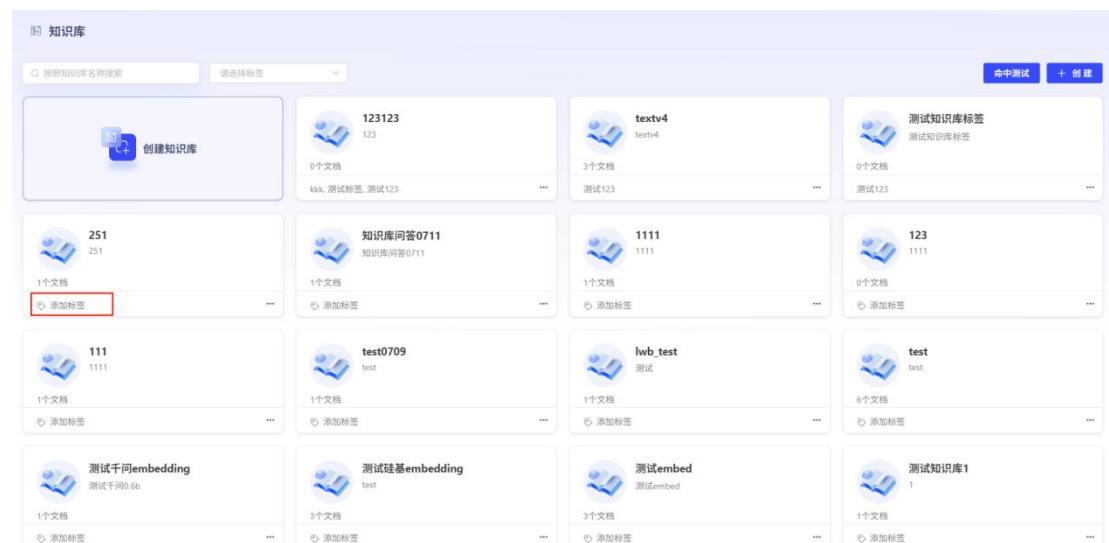
【 API Key 】必须为上述 curl 中可以正确请求的 key ; 例如
sk-abc*****xyz (注意不填 Bearer 前缀)

【 推理 URL 】必须为上述 curl 中可以正确请求的 url ; 例如
<https://maas.ai-yuanjing.com/openapi/compatible-mode/v1> (注意不带
/chat/completions 或 /embeddings 或 /rerank 后缀)

二、知识库

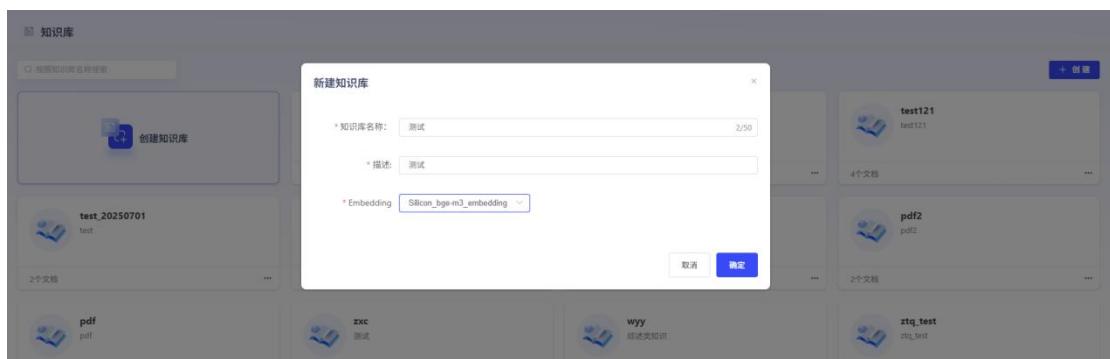
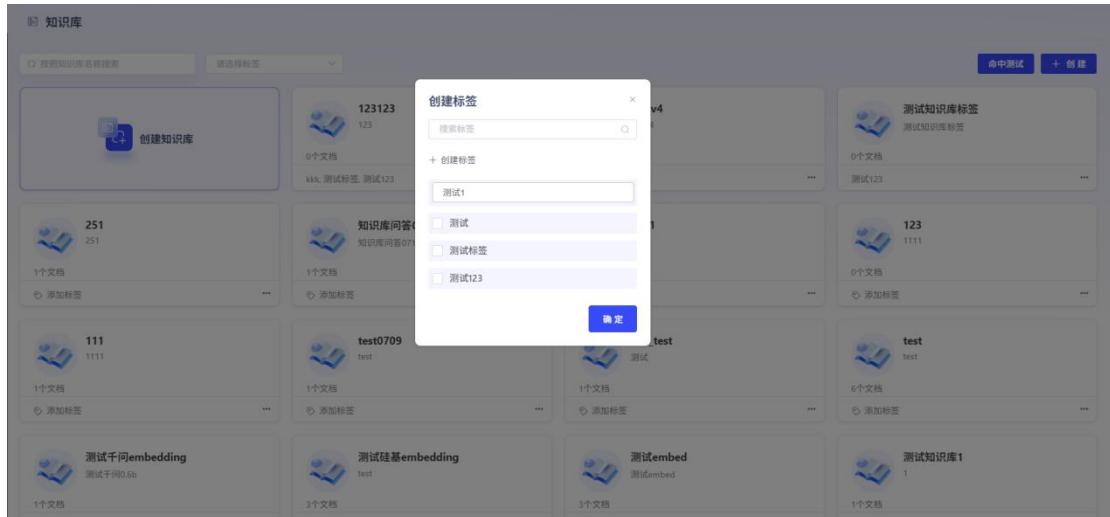
1、创建知识库

若用户要使用 RAG 功能，需要先创建知识库，关联 Embedding 模型。其中 Embedding 模型，需提前在模型管理模块上传。用户也可进行标签设置，方便进行知识库筛选分类。（已创建但未填写内容的标签，可通过 Backspace 键快捷删除）



The screenshot shows a user interface for managing knowledge bases. At the top, there is a search bar labeled '按照知识库名称搜索' and a dropdown menu labeled '请选择标签'. On the right side, there are two buttons: '命中测试' and '+ 创建'. Below these, there is a table displaying 16 knowledge bases arranged in a grid of four rows and four columns. Each row contains four cards. The first card in each row has a red box around the '添加标签' button. The second card in each row has a red box around the '添加标签' button. The third card in each row has a red box around the '添加标签' button. The fourth card in each row has a red box around the '添加标签' button.

知识库名	文档数	描述	操作
123123	0个文档	123	... 添加标签
textv4	3个文档	textv4	... 添加标签
测试知识库标签	0个文档	测试知识库标签	... 添加标签
251	1个文档	251	... 添加标签
知识库问答0711	1个文档	知识库问答0711	... 添加标签
1111	1个文档	1111	... 添加标签
123	0个文档	1111	... 添加标签
111	1个文档	111	... 添加标签
test0709	1个文档	test	... 添加标签
lwb_test	1个文档	测试	... 添加标签
test	6个文档	test	... 添加标签
测试千问embedding	1个文档	测试千问embedding	... 添加标签
测试硅谷embedding	3个文档	test	... 添加标签
测试embed	3个文档	测试embed	... 添加标签
测试知识库1	1个文档	1	... 添加标签



创建知识库：

1) 文件上传：平台支持用户上传本地文件或者从 url 上传。



2) 参数配置：参数配置：平台支持自动分段与自定义分段2种模式。自定义分段，用户可设置分段时的标点符号、可分割最大值、可重叠值。若在模型广场中接入了ocr模型，解析方式还可勾选OCR，并选择对应模型。

文档处理状态：

1) 处理状态：上传完毕的文档可在文档列表中查看。文档内容的解析切分状态可在当前状态列中查看。当状态为“处理完成”，该文档知识即可在后续 RAG 使用知识库中生效。



2) 查看分段结果：点击操作列的“查看”，即可查看状态为处理完成的文档分段结果。



在分段结果查看中，可以查看文档分段策略，及每一个分段的内容结果。可根据用户需求，点击每个分段卡片上的启停开关，对单独分段进行启动和停止。也可点击一键全启动或一键全停用，对整个文档的分段结果进行启动和停止。

注：停止的分段内容在使用 RAG 能力时不会生效。如需查看切段完整信息，点击分段卡片，可展示该分段的详情。

2、命中测试

当用户需要测试知识库是否生效时，可使用“命中测试”功能快速测试。点击“命中测试”按钮，进入命中测试界面。在“命中测试分段”中输入关键词并配置检索方式。如命中结果，即可获得相关命中得分，及分段内容，确认该知识分段已生效。

目前支持 3 种检索方式配置，用户可根据知识库内文档的内容特点及使用场景，

调整检索策略：

1、向量检索：通过向量相似度找到语义相近、表达多样的文本片段，适用于理解并召回语义相关信息。

2、全文检索基于关键词匹配，能够高效查询包含指定词汇的文本片段，适用于精确查找。

3、混合检索：结合向量和关键词检索，融合语义理解与关键词匹配，兼顾相关性和准确性，提升检索效果。

The screenshot shows a user interface for managing a knowledge base. At the top, there is a search bar labeled '按图知识库名称搜索' and a dropdown menu labeled '请选择标签'. On the right side of the header, there are two buttons: '命中测试' (highlighted with a red box) and '+ 创建' (Create). Below the header, the main area displays a grid of document cards. Each card contains a thumbnail, a title, a subtitle, a document count, and a '添加标签' (Add Label) button. The cards are arranged in four rows and four columns. Some titles include '创建知识库', '123123', 'textv4', '测试知识库标签', '251', '知识库问答0711', '1111', '123', '1111', 'test0709', 'lwb.test', 'test', '测试千问embedding', '测试链基embedding', '测试embed', '测试知识库1', 'markdowntest', 'test0703', 'wuxk26', and 'test121'. The '命中测试' button is located at the top right of the interface.

命中分段测试

青铜冰鉴

开始测试 ➔

检索方式配置

向量检索
通过向量相似度找到语义相近、表达多样的文本片段，适用于理解和召回语义相关信息。

Rank模型
jina-reranker-m0

TopK 5
Score阈值 0.4

全文检索
基于关键词匹配，能够高效查询包含指定词汇的文本片段，适用于精确查找。

混合检索
结合向量和关键词检索，融合语义理解与关键词匹配，兼顾相关性和准确性，提升检索效果。

命中预测结果

命中得分: 0.98370

战国时期青铜冰鉴 1977 年，战国铜冰鉴出土于湖北省随州市擂鼓墩—曾侯乙墓。战国铜冰鉴长宽均为 76 厘米、高 63.2 厘米，是一件双层方形的青铜盛酒器，由一个方盖和一件方尊缶组成，缶为方体，像一个方口的大盆，腹深，平底，四个兽足。盖口四角及四边分别有方形或圆形附耳，每口沿上相间的裸根素接，盖的支脚，由四只龙首身的怪兽组成。龙头向外伸张，兽身的后肢前踏蹲地。盖身的四条棱和四块梯形附耳，每块梯形附耳上都有两个长曲的龙形耳环，龙形呈拱曲垂状，每条龙的头顶承托 8 块接槽式铁环，钮尾均有小龙缠绕，又有两条五瓣小花立于尾上。盖内中部有方孔，盖内之，在口沿部从下方开槽露出，盖之四周各有一圈素面环，以便启闭盖子。盖上浮雕变形蟠螭，盖体上多浮雕蟠螭纹，下腹部有环，盖内刻铭与盖铭相同。方尊缶置于方鉴内，製作外套，缶在其中，缶的外缘和鉴的内缘之间有很大的空间，具有储酒、加温酒液的双重功能。其底那一侧有两个长方形榫眼，另一侧有一个长方形榫眼，安装时，把这个三个榫眼与方鉴内的三个卯孔合扣，其中一个榫的活动倒钩自动向下后。

文件名称：战国时期青铜冰鉴.pdf

命中得分: 0.97716

战国铜冰鉴为后人展示了中国古代青铜铸造的高超技法和先进水平，不仅成为了研究古代历史的重要文物，更是青铜铸造史上的典范之作，堪称青铜时代的巅峰作品之一。<http://192.168.0.21:8081/minio/download/api/rag-public/tmp.1850hm7.png> 图 1 青铜冰鉴

文件名称：战国时期青铜冰鉴.pdf

命中得分: 0.95424

可把方鉴固定在方鉴里而不晃动，该出土时带有长柄的勺子，是舀酒的用具。器物本身使用浑铸法（一次整体铸成），装饰附件使用分铸法（分割铸造），镂空附饰则用失蜡法浇模铸造。首先在泥范的内范的外表面敷上一层生蜡（内擦松香和油脂等），用这层生蜡做成欲铸的青铜器的原大模型，并在蜡上面雕刻出精细的花纹。其次，用调好的熔浆一层层淋上去，最后形成泥质外壳。在外范外，把这个三个榫眼与方鉴内的三个卯孔合扣，其中一个榫的活动倒钩自动向下后。

3、关键词管理

在知识库问答场景，可配置问题中使用的口语化关键词与知识库文档中词语的映射规则，便于提升知识库内容的召回效果及回答准确率。

知识库

按照知识库名称搜索 请选择标签

关键词管理 + 创建

知识库	文档数	操作
创建知识库	251	添加标签
测试知识库-wl-01	2个文档	测试标签
silicon_bge_m3_embedding	2个文档	添加标签
bge_m3_embedding_test	2个文档	添加标签
qwen3_embed_0_6b	2个文档	添加标签
123123	1个文档	测试标签, 测试123
textv4	4个文档	测试123
测试知识库标签	0个文档	测试123
251	1个文档	添加标签
知识库问答0711	1个文档	添加标签
1111	1个文档	添加标签
123	0个文档	添加标签
111	1个文档	添加标签
test0709	1个文档	添加标签
lwb_test	1个文档	添加标签
test	6个文档	添加标签

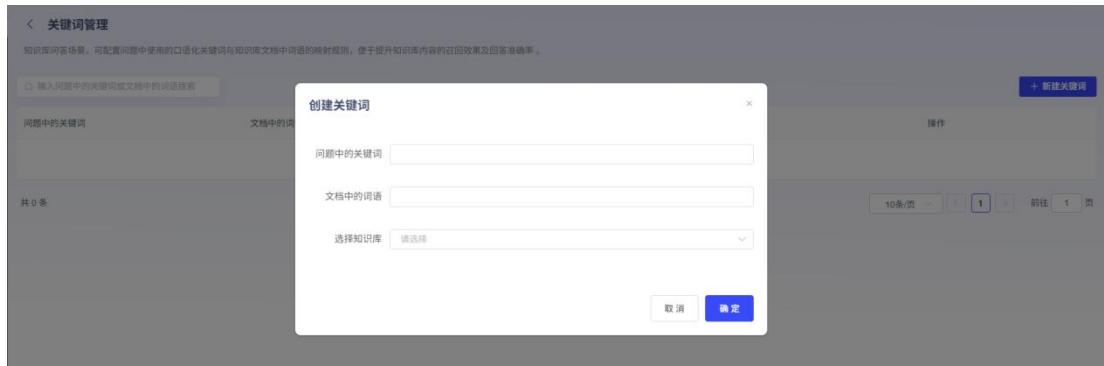
关键词管理

知识库问答场景，可配置问题中使用的口语化关键词与知识库文档中词语的映射规则，便于提升知识库内容的召回效果及回答准确率。

+ 新建关键词

问题中的关键词	文档中的词语	关联知识库	更新时间	操作
暂无数据				
共 0 条				10条/页 前后 1 页

点击“创建关键词”，选择关联知识库，可将口语化表述与文中词语进行映射。



三、MCP 广场

MCP 广场集合纳管 MCP server，并提供精选推荐供用户选择，即选即用。

1、精选推荐

平台提供优选 MCP server，用户点击“导入 MCP 服务”，在编辑界面将 url 替换为自己的链接后使用。从精选推荐导入的 MCP server，可在“导入 MCP 服务”模块统一查看。

服务名称	描述
Amap 高德地图	高德地图 MCP Server 现已覆盖12大核心服务接口，提供全场景覆盖的地理服务能力，包括地理编码、逆地理编码、IP 定位、天气查询、骑行路径规划、步行路径规划、驾车路径规划、公交车路...
Minimax	Minimax官方MCP Server，支持高质量的视频生成、图像生成、语音生成、和声音克隆等多项能力
Time	提供时间和时区转换功能
Google Maps	谷歌地图
Fetch	获取网页内容并将其转换为 Markdown 格式
Brave Search	Brave Search MCP Server集成了灵活过滤的网页和本地搜索功能。
Perplexity	Perplexity官方MCP Server. 通过Sonar API实现实时的网页搜索功能
Firecrawl	提供大规模和复杂的网页数据爬取及结构化解析的官方服务。
12306	提供12306火车票查询及相关服务
快递100物流查询助手	基于快递100官方API的物流轨迹查询和快递服务预估工具
Markmap思维导图生成器	将Markdown文档转换为交互式思维导图的可视化工具
Trends Hub热点资讯聚合器	集成多平台热门资讯和趋势数据的综合信息聚合工具
IPInfo IP地址信息查询器	专业的IP地址地理位置和网络信息查询工具
归属地查询工具	专业的手机号、IP地址、银行卡、身份证号归属地查询服务
序列思维工具	动态反思性问题解决工具，通过灵活的思维过程分析复杂问题
百度地图服务	基于百度地图API的综合地理信息服务工具，提供地理编码、路线规划、地点检索等功能

The screenshot shows the MCP Marketplace interface. At the top, there's a header with a back button labeled "返回MCP广场" and a search bar. Below the header, there's a section for the "Amap 高德地图" service, which includes a brief description: "高德地图 MCP Server 现已覆盖12大核心服务接口，提供全场景覆盖的地图服务，包括地理编码、逆地理编码、IP 定位、天气查询、骑行路径规划、步行路径规划、驾车路径规划、公交路径规划、距离测量、关键词搜索、周边搜索、详情搜索等。" There are two tabs at the top of this section: "介绍概览" (selected) and "SSE URL及工具". To the right of this main content, there's a sidebar titled "其他MCP服务查看" containing several cards:

- Amap 高德地图**: Description: "高德地图 MCP Server 现已覆盖12大核心服务接口，提供全场景覆盖的地图服务，包括地..."
- Minimax**: Description: "Minimax官方MCP Server，支持高质量的视频生成、图像生成、语音生成、和声音剪辑"
- Time**: Description: "提供时间和时区转换功能"
- Sequential Thinking**: Description: "提供了一种通过结构化思维过程进行动态和反向性问题解决的工具。"
- Google Maps**: Description: "谷歌地图"

2、导入 MCP 服务

点击“导入”，关联自己的 MCP，可支持在工作流中使用。用户需填写服务名称、服务来源、功能描述、MCP ServerURL。点击“获取 MCP 工具”，获取成功后，将显示可用接口。点击“确认发布”，即可在平台上查看每个工具的详细内容。

The screenshot shows the MCP Marketplace interface. On the left, there's a sidebar with a "精选推荐" section and a "导入MCP服务" button. The main area shows a "添加MCP服务" dialog box with the following fields filled in:

- * 服务名称: 高德MCP
- * 服务来源: 高德
- * 功能描述: 高德
- * MCP sseUrl: <https://mcp.amap.com/sse?key=...>

Below the dialog, there's a list of available MCP tools:

- maps_direction_bicycling
- maps_direction_driving
- maps_direction_transit_integrated
- maps_direction_walking
- maps_distance
- maps_geo
- maps_regeocode

At the bottom right of the dialog, there's a "获取MCP工具" (Get MCP Tools) button. To the right of the dialog, there's a separate panel showing a list of MCP services with their URLs:

- wxm_test: https://mcp.amap.com/wxm_test
- wom_test: https://mcp.amap.com/wom_test

The screenshot shows the MCP platform's SSE URLs tool. It includes a URL input field with the value <https://mcp.amap.com/sse?key=6d889bd5aa34bd63a39c1197a00e377>, a note saying "您已添加到自定义，可直接在工作流或智能体中直接调用。", and a section titled "工具介绍" with descriptions for "maps_direction_bicycling" and "maps_direction_driving". To the right, there is a sidebar titled "其他MCP服务查看" listing various services: Amap (高德地图), MiniMax (Minimax官方MCP Server), Time (提供时间和时区转换功能), Sequential Thinking (提供了一种通过结构化思维过程进行动态和异步事件处理解决的工具), Google Maps (谷歌地图), and Fetch (Fetch API).

四、安全护栏

根据不同行业需求，用户可维护自己的敏感词表，来控制模型反馈结果的安全性。

1、创建词表

点击“安全护栏”进入敏感词管理

This screenshot shows the 'Create Vocab Table' interface. It features a list of existing vocabularies: '测试表1' (Test Table 1) with ID 1111 and '测试表2' (Test Table 2) with ID 2222. Below this is a 'Create New Vocab Table' button. A note at the top states: "支持用户自定义敏感词表，配置行业敏感词，实时拦截高风险内容的输入和输出，保障内容安全合规。可在创建应用时关联配置。"

点击“创建敏感词”，进入创建界面

This screenshot shows the 'Create New Vocab Table' dialog box. It contains fields for '敏感词表名' (Table Name) and '敏感词' (Sensitive Words). There is also a '表备注' (Table Note) section and a note at the top: "支持用户自定义敏感词表，配置行业敏感词，实时拦截高风险内容的输入和输出，保障内容安全合规。可在创建应用时关联配置。". At the bottom are '取消' (Cancel) and '确定' (Confirm) buttons.

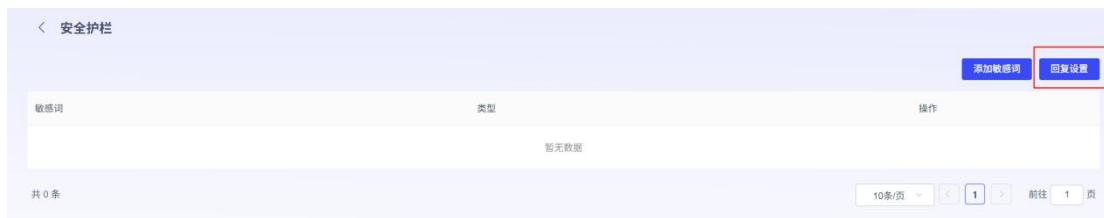
2、编辑词表

编辑词表内容，点击敏感词管理列表操作列的“编辑”按钮，进入词表编辑页。

点击“添加敏感词”，可单条或批量添加敏感词。



点击“回复设置”，可自定义敏感词命中后的模型回复内容。若不设置，系统将默认回复：您当前请求或者历史信息中包含敏感内容，根据相关安全要求，该轮对话终止，请清除历史信息或者重新开启新的对话。





3、使用敏感词

用户可在智能体或文本问答应用中，配置安全护栏，关联已创建好的敏感词表。

启用安全护栏工具后，若命中敏感词，将根据配置好的回复进行解答。

文本问答编辑

API地址: http://192.168.0.21:8081/service/api/openapi/v1/ag/chat | API秘钥 | 发布 |

nicolyn_test nicolyn_test

模型选择: deepseek/r1-distill-llama-8b_通义

关联知识库: 11础基测试

nicolyn_test
nicolyn_test

请输入内容,用Ctrl+Enter可换行

发送 >

检索方式配置

向量检索: 通过向量相似度找到语义相近、表达多样的文本片段，适用于理解和召回语义相关信息。

全文检索: 基于关键词匹配，能够高效查询包含指定词汇的文本片段，适用于精确查找

混合检索: 结合向量和关键词检索，融合语义理解与关键词匹配，兼顾相关性和准确性，提升检索效果。

安全护栏配置

智能体编辑

API地址: http://192.168.0.21:8081/service/api/openapi/v1/agent/chat | API秘钥 | 发布 |

开场白: 你好

系统提示词: 描述你想创建的应用，详细描述应用的詳細功能及作用。

推荐问题:

新增敏感词

确定

测试敏感词

文本问答编辑

API地址: http://192.168.0.21:8081/service/api/openapi/v1/rag/chat | API秘钥 | 发布 |

模型选择: DeepSeek-V3-0324

关联知识库: kylinan-test

检索方式配置

向量检索: 通过向量相似度找到语义相近、表达多样的文本片段，适用于理解和召回语义相关信息。

Rank模型

全文检索: 基于关键词匹配，能够高效查询包含指定词汇的文本片段，适用于精确查找

混合检索: 结合向量和关键词检索，融合语义理解与关键词匹配，兼顾相关性和准确性，提升检索效果。

安全护栏配置

一、基础应用模块

类别	功能说明
模型开发	模型训练、量化、压缩、量化（支持渐进式训练，多模型对比评估）。
应用构建	通过拖拽组件搭建应用，识别预训练、Action API集成（对接第三方接口）并发生响应。
安全管理	权限管理、数据脱敏、隐私安全合规（多模型安全合规审计）。
算力支撑	提供算力资源、ROMA低代码加速、支持云/本地化一体机多种部署模式。
三、典型应用场景	
垂类行业	金融咨询（ChatConsult）、医疗文本分析（ChatDoc）、音视频处理（ChatVideo/Audio）。
企业需求	低成本公有云训练、高安全本地化部署（医疗、军工领域）。
四、基础设施	
成本优化	减少标注、训练、推理全环节开支。
灵活适配	支持云上或云下模型（DeepSeek、文心一言），并提供模型训练工具。
扩能性	部署至第三方云模型（如DeepSeek、文心一言），并提供模型训练工具。

如果进一步了解具体操作（如模型治理或模型部署应用创建），请参考平台内的完整菜单或详细指南。

二、云大模型VaaS平台介绍

2023年5月发布的云大模型VaaS平台是面向公有云大模型客户群体的云平台产品，结合大模型技术演进需求及市场需求，打造跨地域、跨模型、跨语言的云大模型训练、推理、部署、运维、数据治理、模型治理、模型安全、个人隐私保护等全方位一体化解决方案。云大模型VaaS平台具备以下特点：1. 算法先进：采用深度学习框架，结合先进的模型设计，支持大规模并行训练，提高训练效率；2. 安全可靠：通过端到端的安全机制，保障数据传输和存储的安全性，同时提供强大的安全审计和日志记录功能；3. 易用性：提供统一的API接口，方便开发者快速上手，同时提供丰富的文档和支持，帮助开发者快速掌握平台使用；4. 扩展性强：平台支持多模型部署，方便用户根据需求灵活调整，同时提供强大的模型管理功能，支持模型的自动更新和版本管理；5. 高性能：通过高效的基础设施和算法优化，提供高性能的推理服务，满足大规模数据处理需求；6. 低成本：通过云大模型VaaS平台的低成本优势，降低企业的大模型部署门槛——统一化、零代码、低成本，降低企业的大模型部署门槛。

三、常见问题

- 问题1: 云大模型VaaS平台如何实现跨地域训练？
- 回答: 云大模型VaaS平台提供跨地域训练功能，支持在不同地域间无缝切换训练环境，满足企业对数据本地化的需求。
- 问题2: 云大模型VaaS平台如何保证数据安全？
- 回答: 云大模型VaaS平台通过严格的数据加密、访问控制和定期审计，确保数据在传输和存储过程中的安全性。
- 问题3: 云大模型VaaS平台如何支持多模型部署？
- 回答: 云大模型VaaS平台提供多模型部署功能，支持同时运行多个不同的大模型，满足企业在不同场景下的需求。

四、常见问题

您当前请求涉及的历史信息中包含敏感内容，根据相关安全要求，请勿对这些进行操作。请清除历史信息或者重新开始新的对话。

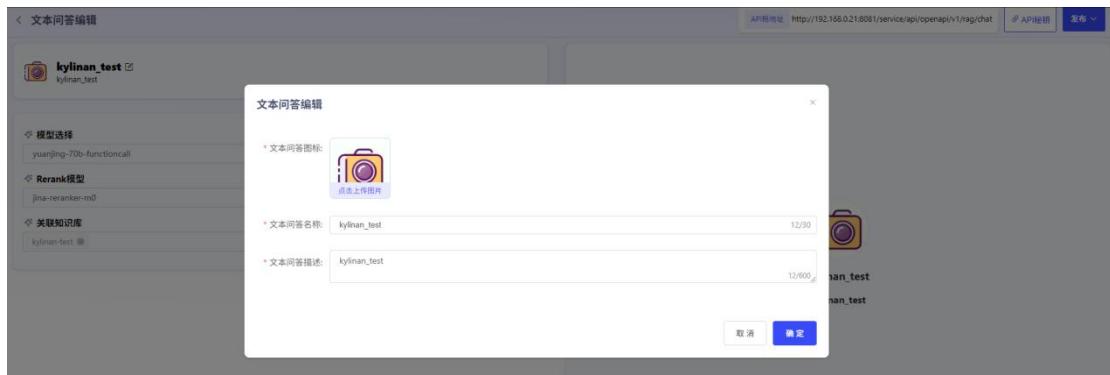
输入框: 请在此输入您的问题

发送 >

五、文本问答

1、文本问答创建

点击“创建文本问答”即可创建文本问答应用。用户可自行设定文本问答图标、名称、描述。



2、文本问答编辑

可通过选择大模型、知识库、配置检索方式，进行文本问答。问答将局限于知识库内，并给出相应出处。

目前支持 3 种检索方式配置，用户可根据知识库内文档的内容特点及使用场景，调整检索策略：

- 1、向量检索：通过向量相似度找到语义相近、表达多样的文本片段，适用于理解召回语义相关信息。
- 2、全文检索基于关键词匹配，能够高效查询包含指定词汇的文本片段，适用于精确查找。
- 3、混合检索：结合向量和关键词检索，融合语义理解与关键词匹配，兼顾相关性和准确性，提升检索效果。

3、文本问答发布

编辑完毕的文本问答应用，点击“发布”可进行发布方式选择，用户可进行私密发布，也可进行公开发布。

私密发布：发布后仅对自己可见，可在“探索广场” - “私密发布的”查看。

公开发布：发布后可对全部用户进行共享，所有用户可在“探索广场” - “全部”查看。



已发布的文本问答也可取消发布后，重新进行编辑。



4、生成 API

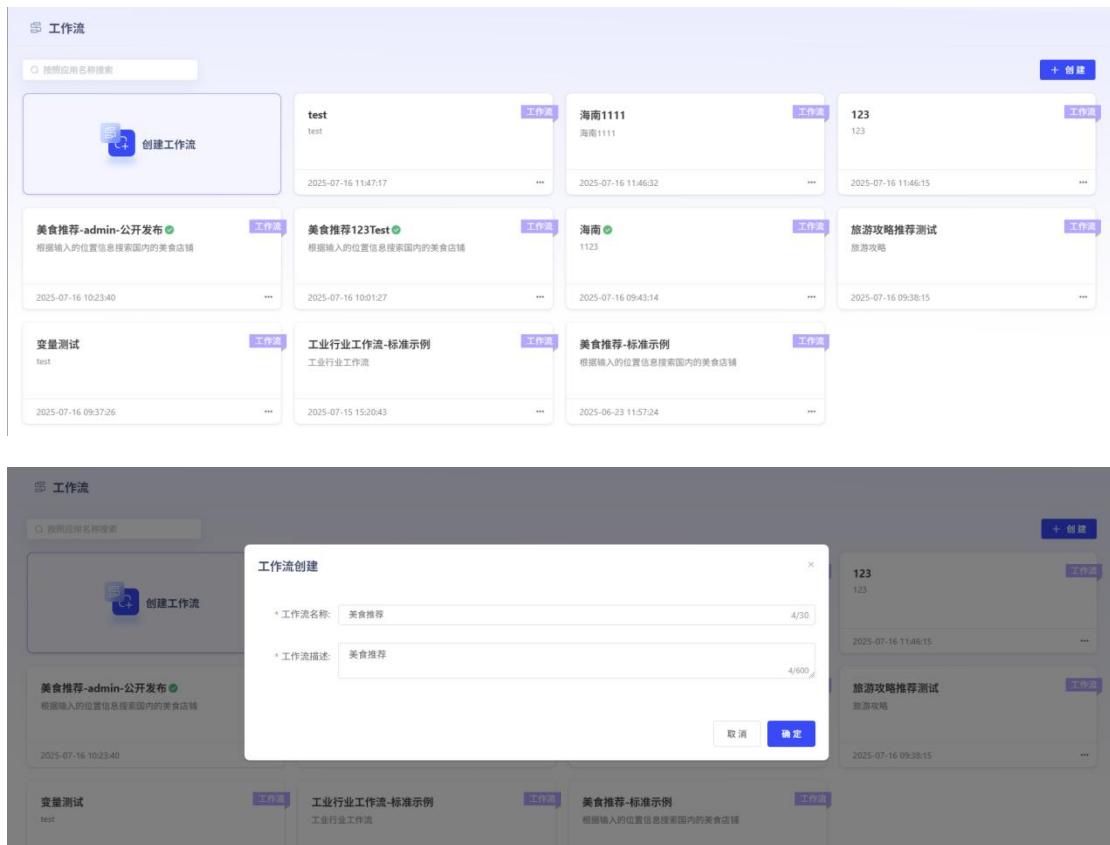
平台已针对应用封装了 API，可点击“API 秘钥”生成应用的专属 API-Key 授权，供用户进行调用。



六、工作流

1、工作流创建

点击“创建工作流”即可进入工作流创建界面。用户可自行设定工作流名称、工作流描述。平台内置了标准工作流示例，用户也可直接复制使用。



The figure consists of two screenshots of a work flow management system. The top screenshot shows a grid of work flows with columns for name, description, status, and creation date. The bottom screenshot shows a modal dialog for creating a new work flow, with input fields for name and description, and a confirmation button.

2、工作流编辑

平台提供 MCP、意图识别、API、代码、大模型、分支器、知识库、文档生成、文档解析节点。

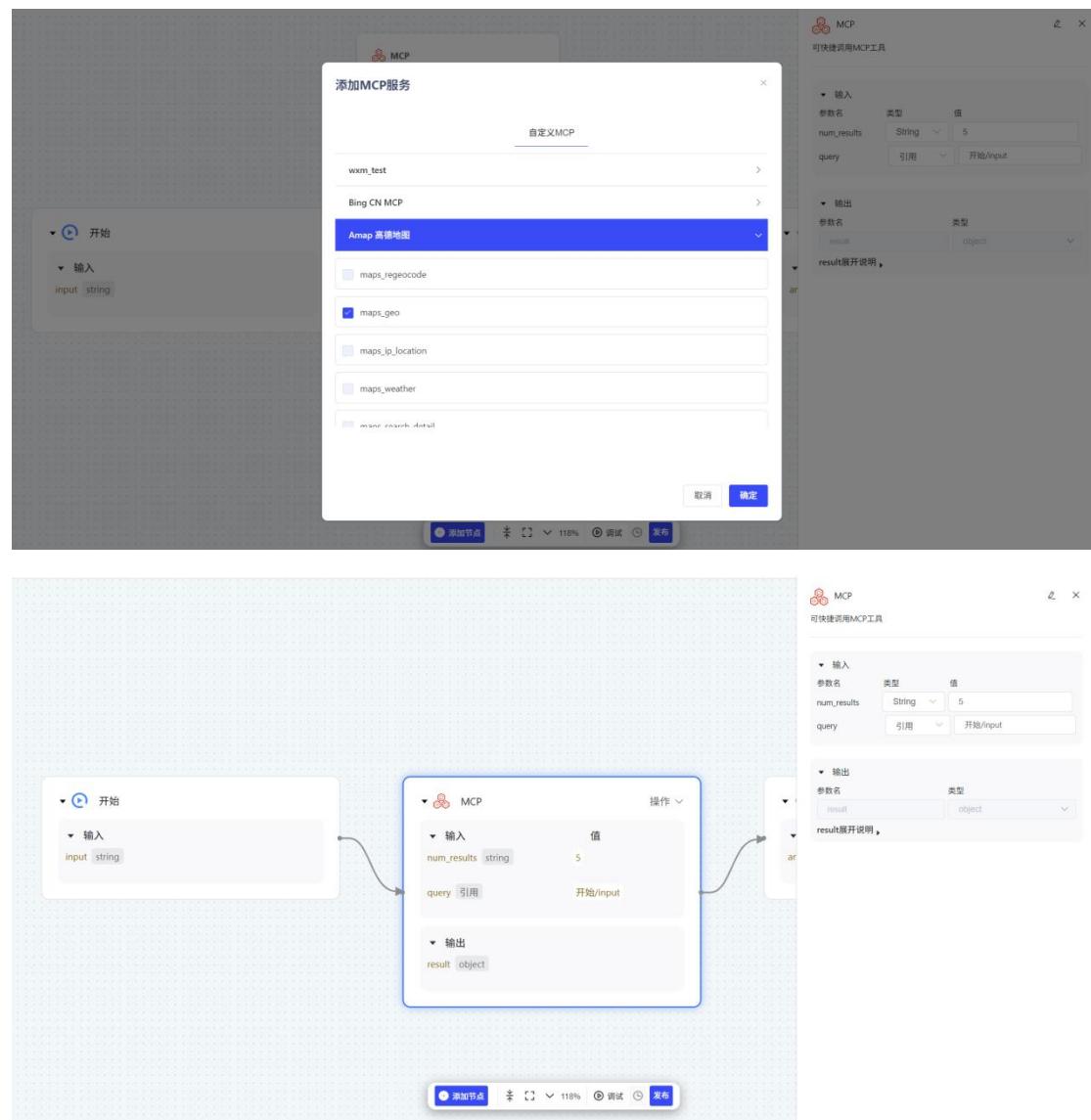
开始

“开始”节点不可以被复制或删除。定义所需的输入参数，会在插件被应用调

用时，由思考模型根据参数描述从用户输入的原始内容中抽取并传入。

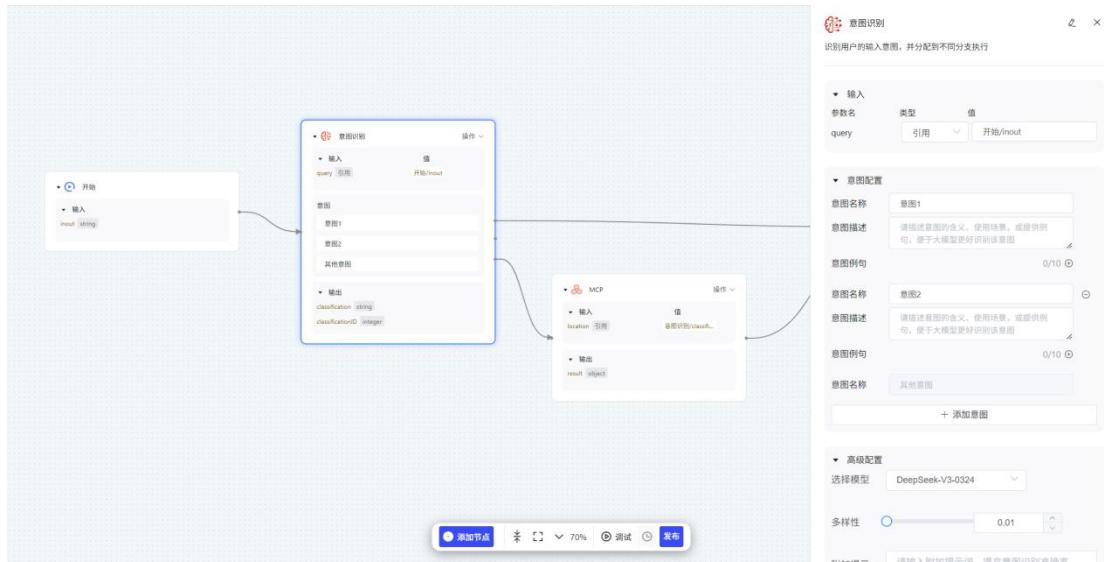
MCP

调用 MCP server 的能力。可添加已发布在 MCP 广场中的 MCP server，并关联对应接口生成结果。



意图识别

识别用户的输入意图，并分配到不同分支执行。



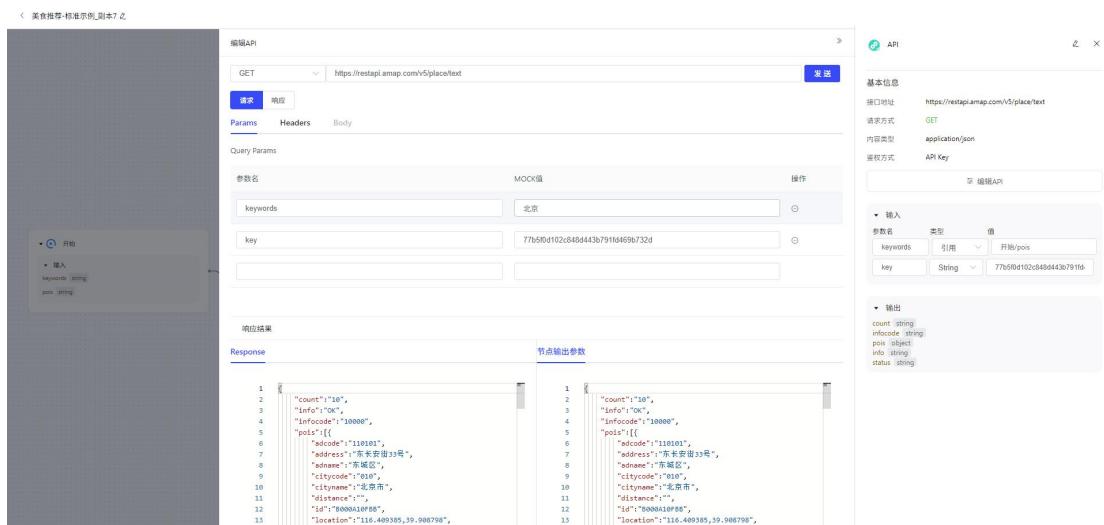
API

支持接入遵守 RESTful 架构规范并采用标准认证机制的 API。API 接入前需要已经完成服务化，确保所有接入的 API 均遵循 RESTful 架构规范，确保使用无状态的请求模型并通过 HTTP 标准方法（如 GET、POST）进行资源操作，同时接口使用标准的认证机制。此外，接口设计符合 OpenAPI 规范，接入的 API 已经过性能测试、安全审查和兼容性检测，以确保 API 的稳定性和可靠性。

使用说明：

I. 添加 API 节点，可以手动逐步配置 API 基本信息。配置信息包括：请求参数和响应参数。您可不配置响应参数，直接根据 API 请求结果的返回信息自动解析输出参数，平台解析规则说明：当前支持接口返回内容类型有：JSON、XML、HTML、Plain Text、YAML、CSV。其中 JSON、XML、YAML 可被解析为 JSON 输出，便于您在后续节点引用某个参数或属性，其余类型支持以整个 string 类型输出。

ii. 配置完成请求参数和响应参数后，点击【发送】，进行接口请求，下方将展示对应接口的原始 API 返回信息，和按照响应参数配置获得的参数返回信息，您可以检查返回信息是否符合预期。



iii. 在完成 API 原始信息编辑后，您需要继续在右侧面板中对输入参数进行值配置，可以引用前序节点输出，或手动输入对应参数值。需要确保引用类型与设置类型一致。

API 节点的 http 请求发送成功即为接口调试通过，在整体流程调试前，API 节点需要为调试通过状态。

代码

在代码节点中，可通过编写代码实现自定义的处理功能。引入代码节点到画布中，可在编辑器（IDE）内看到默认的示例代码。

The screenshot shows a Python code editor with the following code:

```

1 # 定义一个 main 函数，用户只能在 main 函数里嵌套代码开发。
2 # 其中，固定传入 params 字典（字典格式），它包含了节点配置的所有输入变量。
3 # 其中，固定返回 output_params 字典（字典格式），它包含了节点配置的所有输出变量。
4 # 运行环境 Python3。
5
6 # main 函数，固定传入 params 参数
7 def main(params):
8     # 用户自定义部分.....
9
10     # 固定返回 output_params 参数
11     output_params = {
12         # 用户自定义部分.....
13         "key0": params["poi0"][0][("location")],
14     }
15
16     return output_params
17

```

The interface includes sections for '输入' (Input) and '输出' (Output), both with dropdown menus for '参数名' (Parameter Name) and '类型' (Type). The '输入' section has 'pos' selected as '引用' (Reference) type. The '输出' section has 'key0' selected as 'String' type.

编辑器的使用：

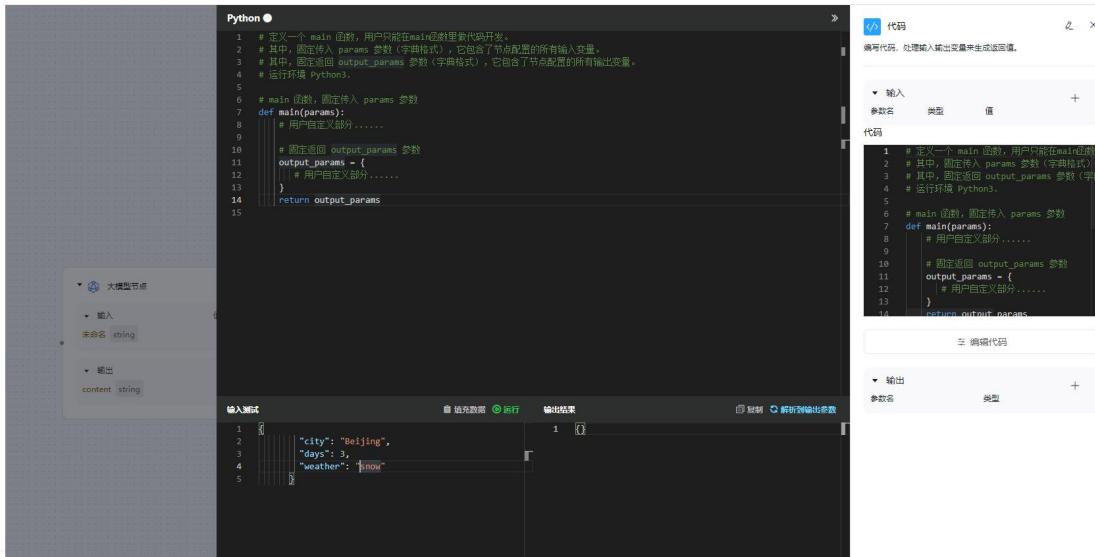
引用输入：配置了输入参数名和参数值后，可以在编辑器中引用输入参数。编辑器引入输入参数时，需要通过字典变量 `params` 引入代码节点的输入参数。代码节点支持不设置输入参数，但是设置的输出参数需要配置对应的默认值或参数引用关系。

返回输出：需要在编辑器中定义一个字典变量，作为编辑器中函数的输出。代码节点的输出参数是该字典变量的键（key）。

运行时环境：编辑器支持的运行时环境为 Python 3。运行环境预制了 NumPy、bs4 包。当前运行环境暂不支持进行 request 请求或获取文件。

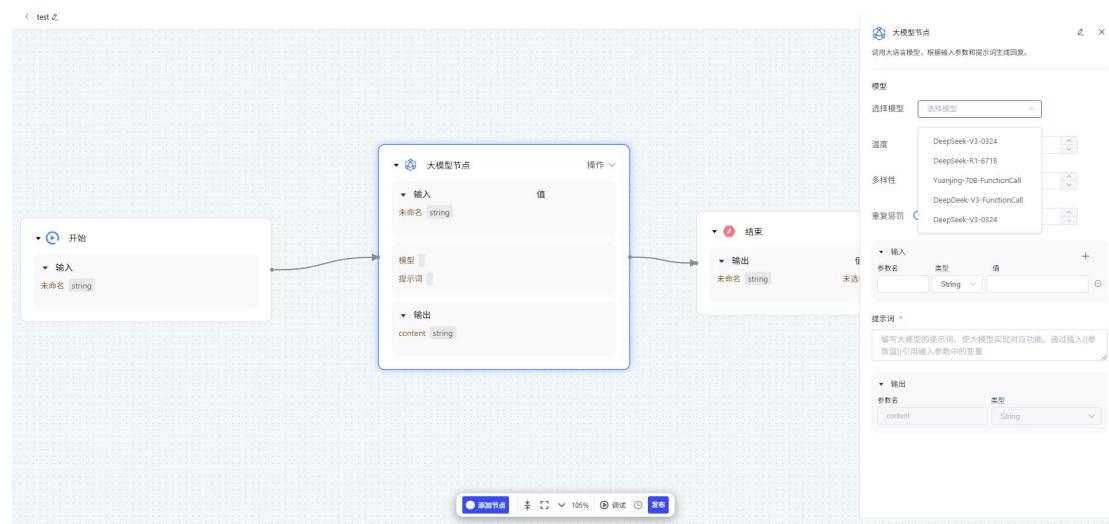
编辑器测试功能：

输入测试：在输入测试区域可以输入测试数据，并进行调试运行。“填充数据”功能可以根据当前输入参数类型，生成输入数据。



大模型

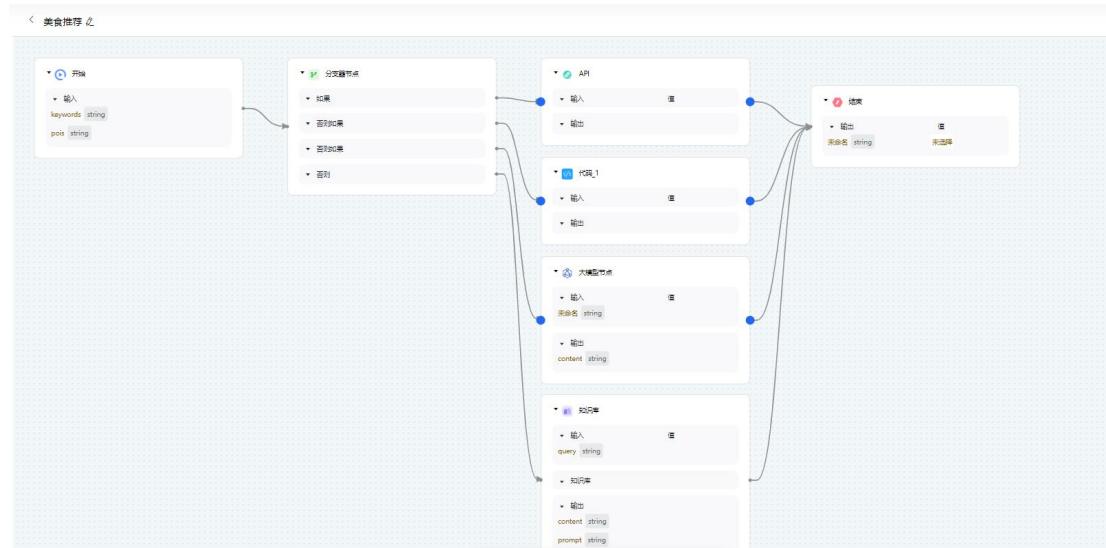
大模型节点可调用大语言模型，根据输入参数和提示词生成回复。



分支器

分支器节点用于设计工作流内的分支流程，可以连接多个下游节点。当向该节点输入参数时，节点会从上到下依次判断是否符合条件，若设定条件成立则运行对

应的条件分支，若均不成立则运行“否则”分支。可通过拖拽分支条件配置面板来设定分支条件的优先级。在每个分支条件中，支持选择判断关系（且/或），以及同时添加多个条件。



知识库

知识库节点支持根据输入的 query，在选定的知识库中检索相关片段并召回，返回切片列表。你可以上传文件并建立知识库，在知识库节点中勾选想要使用的知识库进行检索。知识库节点需选择一个 Rerank 模型后，才能生效。

输入参数：参数名不可修改，参数类型为 string，上级节点的输出参数会强制转换为 string 类型作为知识库节点的输入，输入参数有两种类型：1) 引用类型为引用上一个节点的输出变量，2) 常量类型，可以输入一个 string 类型的入参。

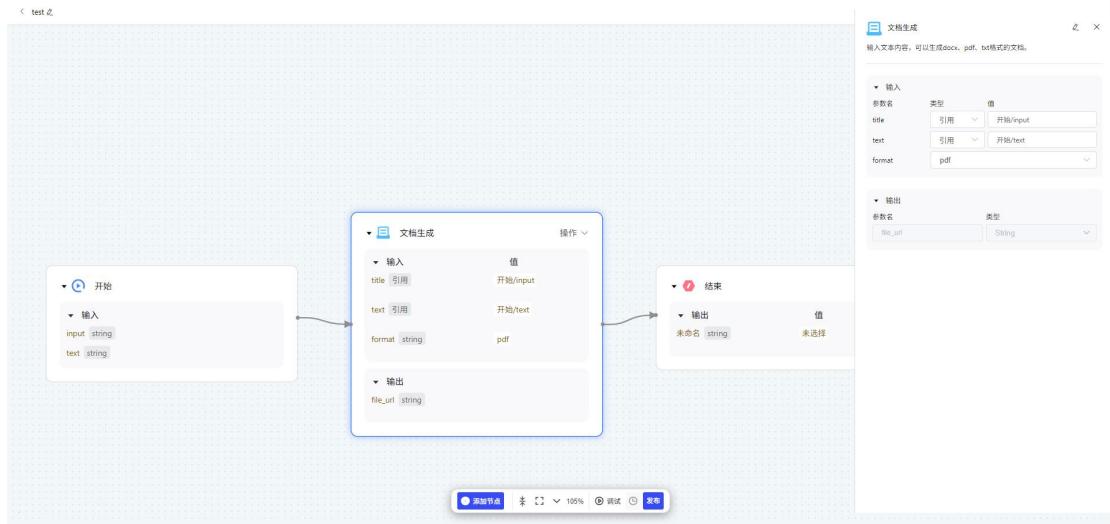
选择知识库：选择需要检索的知识库，支持选择多个知识库。

输出参数：在知识库中检索输出的变量信息及变量类型。



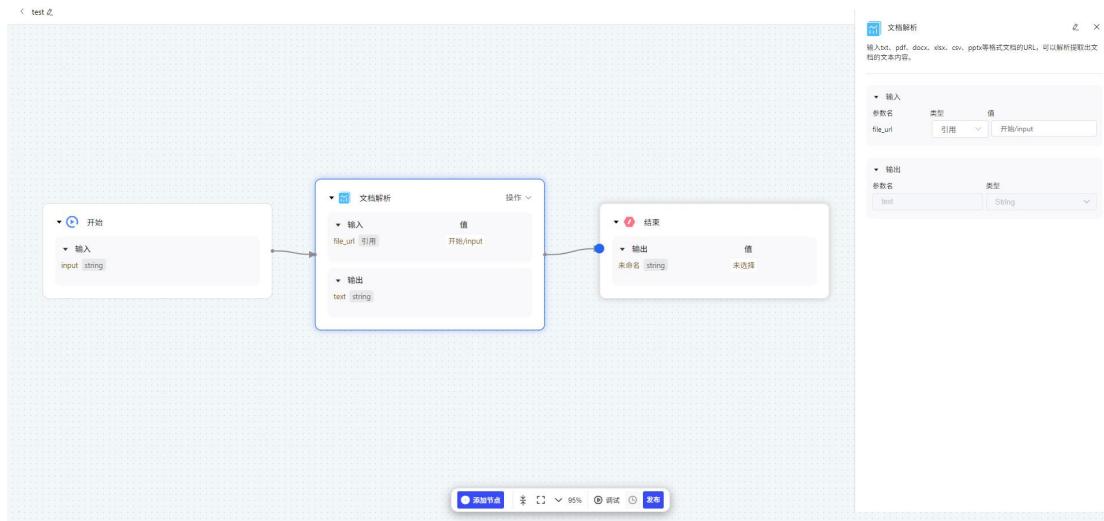
文档生成

输入文本内容，可以生成 docx、pdf、txt 格式的文档。



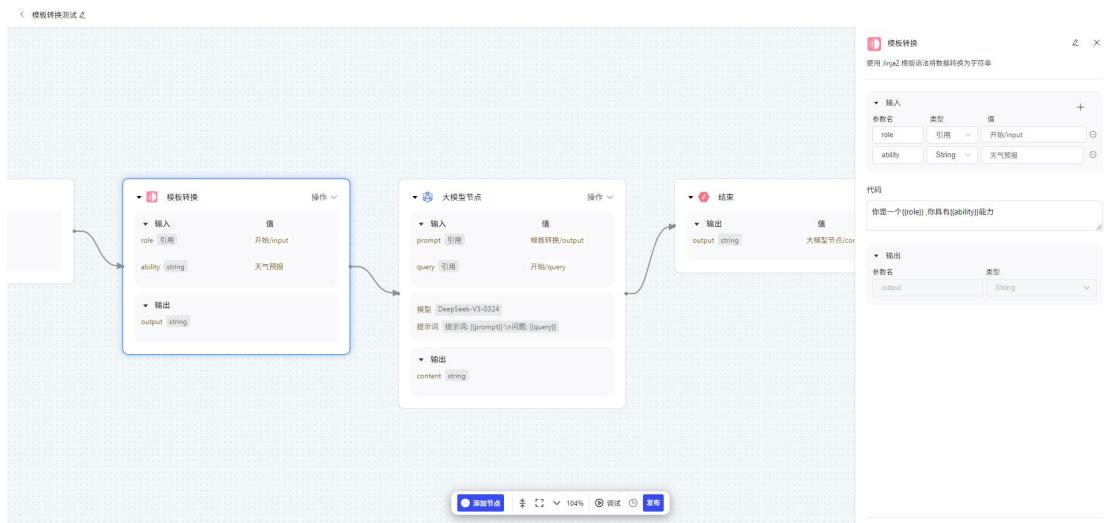
文档解析节点

输入 txt、pdf、docx、xlsx、csv、pptx 等格式文档的 URL，可以解析提取出文档的文本内容。



模板转换

使用 Jinja2 模版语法将数据转换为字符串的工具节点。它适合将前序步骤的多个变量，按需格式化并整合为单一文本，以便为后续步骤提供统一的数据输入。



结束

结束节点是组件的最终节点，输出组件运行后的最终结果。该结果将输出给调用此工具的应用或下游组件。可以配置输出参数：定义组件需要输出的参数。

3、工作流调试与发布

编辑完毕的工作流，点击“调试”，运行成功后，即可进行发布。点击“发布”可进行发布方式选择，用户可进行私密发布，也可进行公开发布。发布完成的工作流可作为工具，被智能体调用。

私密发布：发布后仅对自己可见，可在“探索广场” - “私密发布的”查看。

公开发布：发布后可对全部用户进行共享，所有用户可在“探索广场” - “全部”查看。



已发布的工作流也可取消发布后，重新进行编辑。



七、智能体

1、智能体创建

点击“创建智能体”即可创建智能体。用户可自行设定智能体图标、智能体名称、智能体描述。

The screenshot shows a list of intelligent agents on a platform. A red box highlights the 'Create Intelligent Agent' button at the top left. The list includes various agents with their names, descriptions, and creation dates:

- 生活小助手 (SUSU的生活小助手) - 智能体, created 2025-07-07 15:01:32
- testwxk (testwxk) - 智能体, created 2025-07-03 17:17:57
- 没有rerank (123) - 智能体, created 2025-07-08 11:32:47
- 综合智能体 (综合智能体) - 智能体, created 2025-06-30 17:06:12
- 踏云测试用 (test) - 智能体, created 2025-07-01 20:25:01
- wuxk (wuxk) - 智能体, created 2025-07-03 15:40:43
- fastgptmc (fastgptmc) - 智能体, created 2025-06-27 18:59:20
- wyy (wyy) - 智能体, created 2025-06-30 10:44:48
- 测试 (测试) - 智能体, created 2025-07-07 14:59:28
- 测试联网搜索0708 (测试联网搜索0708) - 智能体, created 2025-07-08 12:53:48
- 测试智能体 (测试智能体) - 智能体, created 2025-07-03 17:25:42
- 抓取网页转换为markdown (抓取网页转换为markdown) - 智能体, created 2025-06-27 14:58:22
- 测试qwen3 (测试qwen3) - 智能体, created 2025-06-27 14:58:22
- 天气查询 (天气查询) - 智能体, created 2025-06-27 14:58:22

2、智能体编辑

智能体可使用以下几类功能进行应用功能 0 代码开发：

选择模型服务： 用户可选择平台中已经纳管的模型，创建智能体。

开场白： 用于编辑开场问候语

系统提示词： 填写应用功能描述、应用处理流程描述、以及对生成结果的要求。

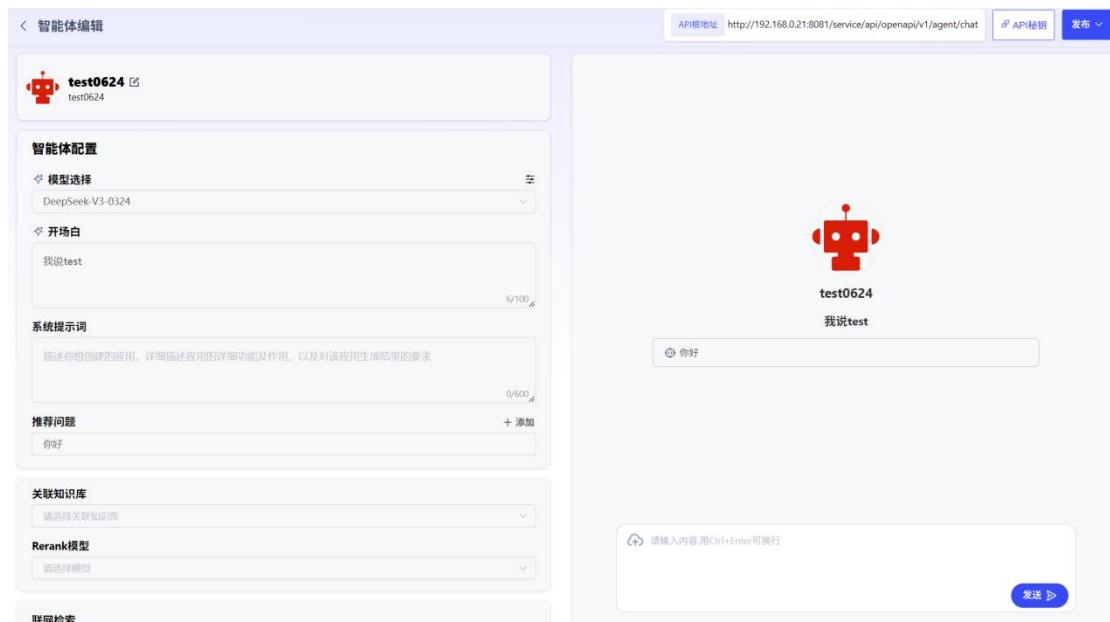
推荐问题： 可设置引导问题

知识库： 用户通过上传文档为大模型进行知识库外挂。外挂知识库后，可与大模型交互文档中的内容。知识库需在“工作室” - “知识库”中提前添加。

联网检索： 通过配置联网检索的 url 和 key，可启用“博查”网络搜索辅助

问答。

工具：用户可添加关联已发布的工作流。用户通过点击“添加”，选择“工作流”，即可添加已发布的工作流。

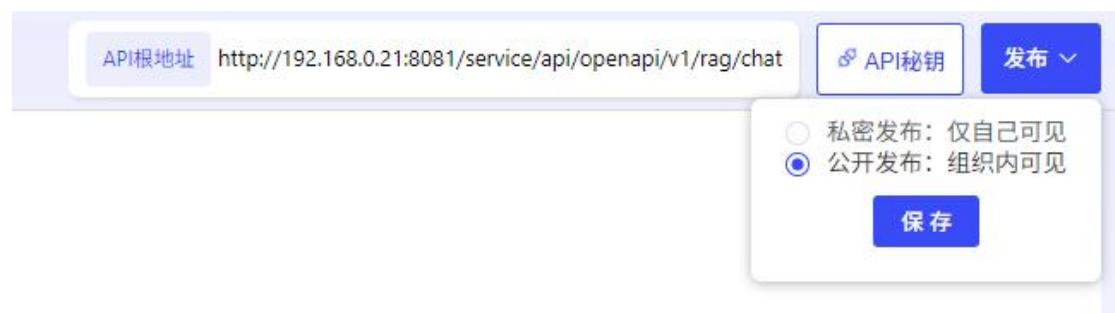


3、智能体发布

编辑完毕的智能体，点击“发布”可进行发布方式选择，用户可进行私密发布，也可进行公开发布。

私密发布：发布后仅对自己可见，可在“探索广场” - “私密发布的”查看。

公开发布：发布后可对全部用户进行共享，所有用户可在“探索广场” - “全部”查看。



已发布的智能体也可取消发布后，重新进行编辑。



4、生成 API

平台已针对应用封装了 API，可点击“API 秘钥”生成应用的专属 API-Key 授权，供用户进行调用。



八、应用广场

1、应用体验

支持用户在探索广场上使用已发布的应用。包括其他用户公开发布的应用以及自己私密发布的应用。对于常用应用，可以点击 进行收藏，方便下次使用。

The screenshot shows a list of intelligent bodies in a grid format. Each card includes the name, icon, status, and creation date.

智能体	文本问答	工作流					
zxc测试admin美食工具-公开	智能体	lwb_test 测试	智能体	选模型测试文件上传 选模型测试文件上传	智能体	测试工具调用 测试工具调用	智能体
2025-07-16 14:57:29	2025-07-15 14:36:06	2025-07-10 18:47:40	2025-07-10 15:53:03				
选模型测试	智能体	高考填报专家.副本2	智能体	测试config 测试config	智能体	高考填报专家	智能体
2025-07-10 14:48:25	2025-07-10 11:27:52	2025-07-10 10:56:10	2025-07-09 22:31:39				
生活小助手	智能体	综合智能体	智能体	测试文件上传问答 测试文件上传问答	智能体	测试知识库 测试知识库	智能体
SUSU的生活小助手	2025-07-07 15:02:47	2025-06-30 17:08:08	2025-06-30 16:39:59	2025-06-30 16:38:20			
智能体0630	智能体	tayun_test 跑云测试	智能体	测试工作流 测试工作流	智能体	627wyw	智能体

文本问答体验

用户点击文本问答卡片，可在线使用文本问答，并可根据 API 接口进行调用。

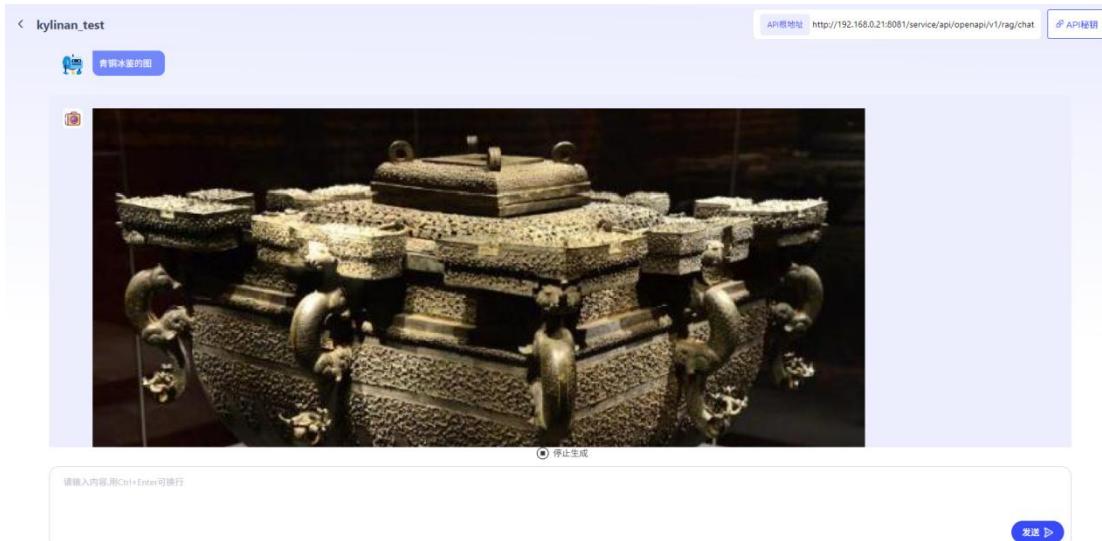
具体调用方式详见接口文档。

The screenshot shows the text question-and-answer interface for the 'kylinan_test' intelligent body. It includes a camera icon, user information, a message input field, and a send button.

kylinan_test
kylinan_test

请输入内容,用Ctrl+Enter可换行

发送 >



工作流体验

用户点击工作流卡片，可在线体验工作流。

The screenshot shows a workflow interface with a search bar and a results panel. The search bar contains '烤鸭' and '北京'. The results panel displays three items:

- 输出:** [{"answer": [{"adcode": "110101", "address": "春悦购物中心中心层春悦新首豪烤鸭(王府井地铁站B1口旁)", "citycode": "010", "cityname": "北京市", "distance": "100m", "id": "E0XFPBGIJY", "location": "116.407625,39.909825", "name": "春悦新首豪烤鸭(王府井店)", "parent": "100000000", "pcodes": "110000", "psname": "北京市", "type": "餐饮服务_中餐厅_中餐厅", "typecode": "050000"}], {"adcode": "110104", "address": "春悦购物中心中心层(王府井地铁站B1口旁)", "citycode": "010", "cityname": "北京市", "distance": "100m", "id": "E0XFPBGIJY", "location": "116.407625,39.909825", "name": "春悦新首豪·烤制(王府井店)", "parent": "100000000", "pcodes": "110000", "psname": "北京市", "type": "餐饮服务_中餐厅_中餐厅", "typecode": "050000"}, {"adcode": "110101", "address": "王府井大街301号春悦购物中心4层", "citycode": "010", "cityname": "北京市", "distance": "170m", "id": "E0XFPBGIJY", "location": "116.410003,39.909977", "name": "大董(北京王府井店)", "parent": "100000000", "pcodes": "110000", "psname": "北京市", "type": "餐饮服务_中餐厅_中餐厅", "typecode": "050000"}]}

智能体体验

用户点击智能体卡片，可在线使用智能体，并可根据 API 接口进行调用。具体调用方式详见接口文档。

The figure consists of three vertically stacked screenshots of a web-based intelligent agent interface. Each screenshot shows a conversation window with a history tab and a detailed response from the AI.

- Screenshot 1:** Shows a general introduction from the AI. It features a blue circular logo with a white bird icon and the text "高考填报专家_副本2". Below it, a message reads: "您可以提供高考生地区、批次、排名、选科、期望就读地区、专业偏好、学费预算等信息，我将提供定制化的志愿填报建议。".
- Screenshot 2:** Shows a detailed response for a user query about Beijing's first batch of 2541 students. The AI provides a comprehensive analysis including:
 - 综合评估 (基于2023年数据参考):** 北京市第一批次理科排名第2541 (2023年北京理科一本线约518分，排名2541对应的分数约630-640分，建议以2025年实际排名和分数线为准)。
 - 建议策略:**
 - 冲: 排名2000名左右的院校 (录取概率40%-50%)
 - 稳: 排名2500名左右的院校 (录取概率60%-80%)
 - 保: 排名3000名以后的院校 (录取概率90%以上)
 - 院校及专业推荐:**
 - 1. 北京交通大学:**
 - 推荐专业:** 计算机类、电子信息类、交通运输类
 - 理由:** 2023年最低录取排名约2000名，计算机专业排名约1800名，北交大工科实力强，就业率高 (央企、国企对口招聘多)。
 - 风险提示:** 计算机专业可能需更高排名，建议服从调剂。
 - 2. 北京科技大学:**
 - 推荐专业:** 材料科学与工程 (A+学科)、人工智能、自动化
 - 理由:** 2023年最低录取排名约2200名，材料学科全国顶尖，理工科保障率高 (20%+)。
 - 风险提示:** 热门专业竞争激烈。
 - 3. 中国政法大学:**
 - 推荐专业:** 法学 (实验班)、经济学 (法律经济学方向)
 - 理由:** 2023年法学实验班录取排名约1900名，普通法学约2300名，适合对政法感兴趣的考生，校友资源丰富。
 - 风险提示:** 热门专业竞争激烈。
- Screenshot 3:** Shows a new conversation window for a user query about Beijing's first batch of 2541 students. The AI is generating a response, indicated by the "正在生成" (Generating) status at the top right.

2、历史应用

可通过历史应用查看历史使用过的全部应用。智能体可查看历史对话内容，可在历史对话窗口下继续交互提问，也可以新建对话再次使用。对于文本问答和工作流应用，不记录历史对话内容，仅支持在线使用。

九、账号

1、个人信息

用户可在个人信息界面，修改登录密码。



2、组织管理

此模块可进行组织、用户、角色的统一管理。

1) 组织管理

组织查看

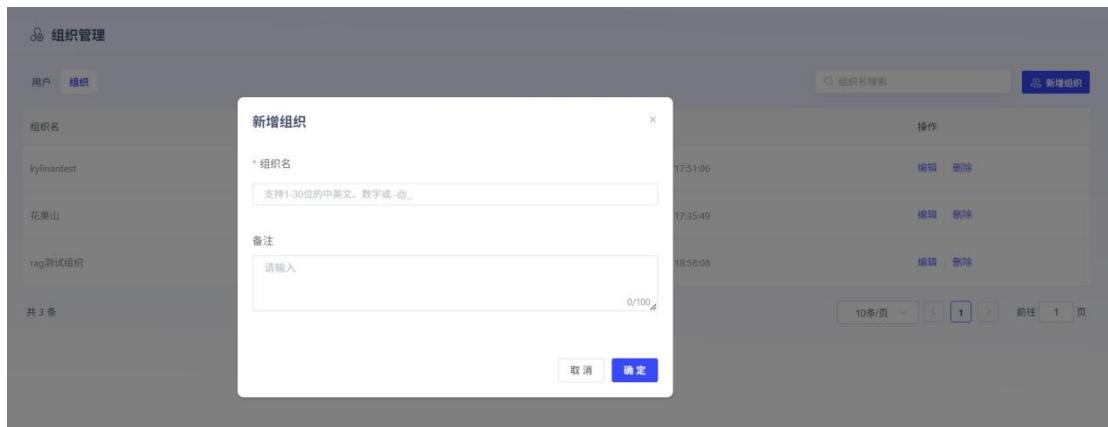
管理员可对该组织内所有子组织情况进行查看，并可编辑、删除、停用和新增子组织。

点击“编辑”，即更改子组织名称。

组织管理			
组织名	创建人	创建时间	操作
kylinanitest	admin	2025-06-09 17:51:06	编辑 删除
花果山	admin	2025-06-05 17:35:49	编辑 删除
rag测试组织	admin	2025-06-04 18:58:08	编辑 删除

新增组织

点击“新增组织”，可对组织名进行设置。点击“确认”，即可完成新增子组织。用户创建下级组织时，系统默认在新组织中创建一个例如“组织管理员”角色，默认拥有新组织的所有权限；同时该用户默认加入新组织，并对应该角色。



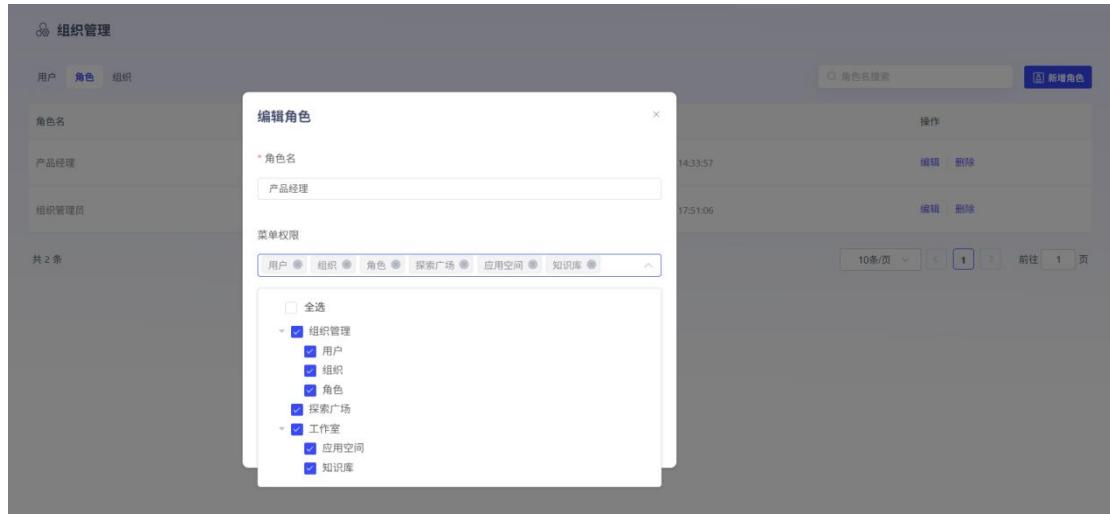
2) 角色管理

角色查看

管理员可对该组织内所有角色情况进行查看，并可编辑、删除、停用和新增角色。其中“组织管理员”角色不可被删除和停用。同一个用户可以在多个组织中。

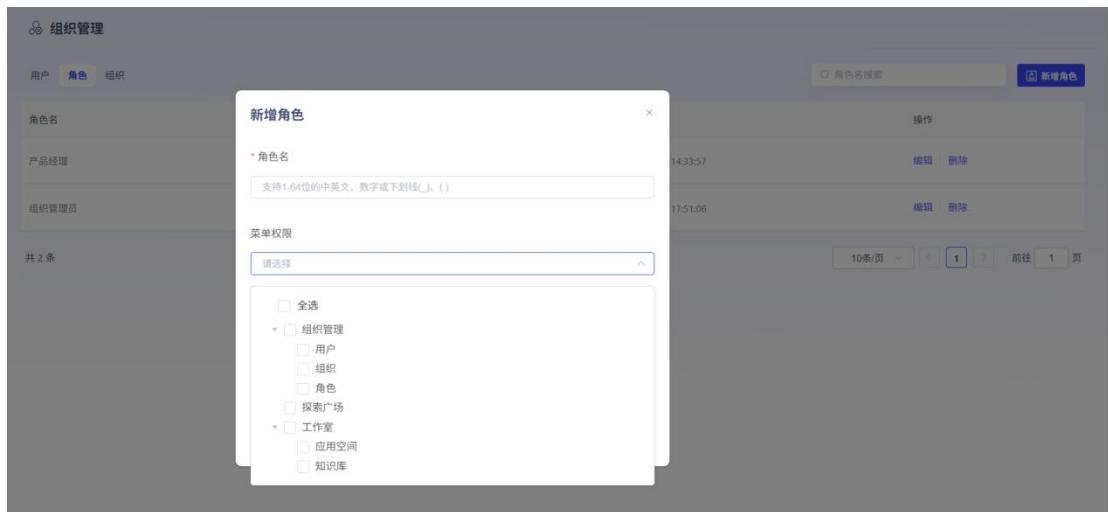
点击“编辑”，即更改角色名称以及菜单权限。

组织管理			
角色名	创建人	创建时间	操作
组织管理员	admin	2025-06-09 17:51:06	编辑 删除
共1条			



新增角色

点击“新增角色”，可对角色名和菜单权限进行设置。点击“确认”，即可完成新增角色。用户创建的新角色，权限集合不超过用户当前对应角色的权限集合。



3) 用户管理

用户查看

管理员可对该组织内所有用户情况进行查看，并可编辑、删除、重置密码、新增账户和邀请用户。

点击“编辑”，即更改用户单位、电话、角色、邮箱。

The screenshot shows the 'Organization Management' section of a software application. At the top, there are tabs for 'User', 'Role', and 'Organization'. Below the tabs, there is a search bar labeled '用户名搜索' and buttons for '新增用户' (Add User) and '邀请用户' (Invite User). A table lists two users: 'kylinan' and 'admin'. Each user row includes columns for '用户名' (Username), '角色' (Role), '创建时间' (Creation Time), and '操作' (Operations). The '操作' column contains links for '编辑' (Edit), '删除' (Delete), and '重置密码' (Reset Password). Below the table, it says '共 2 条' (2 items total). At the bottom right, there are pagination controls for '10条/页' (10 items per page), page numbers 1 and 2, and navigation buttons for '前往' (Go to) and '页' (page).

编辑用户

* 用户名: kylinan

密码: ****

* 单位: 联通数智

* 电话: 17600967982

角色: 组织管理员

邮箱: 请输入

备注: 请输入

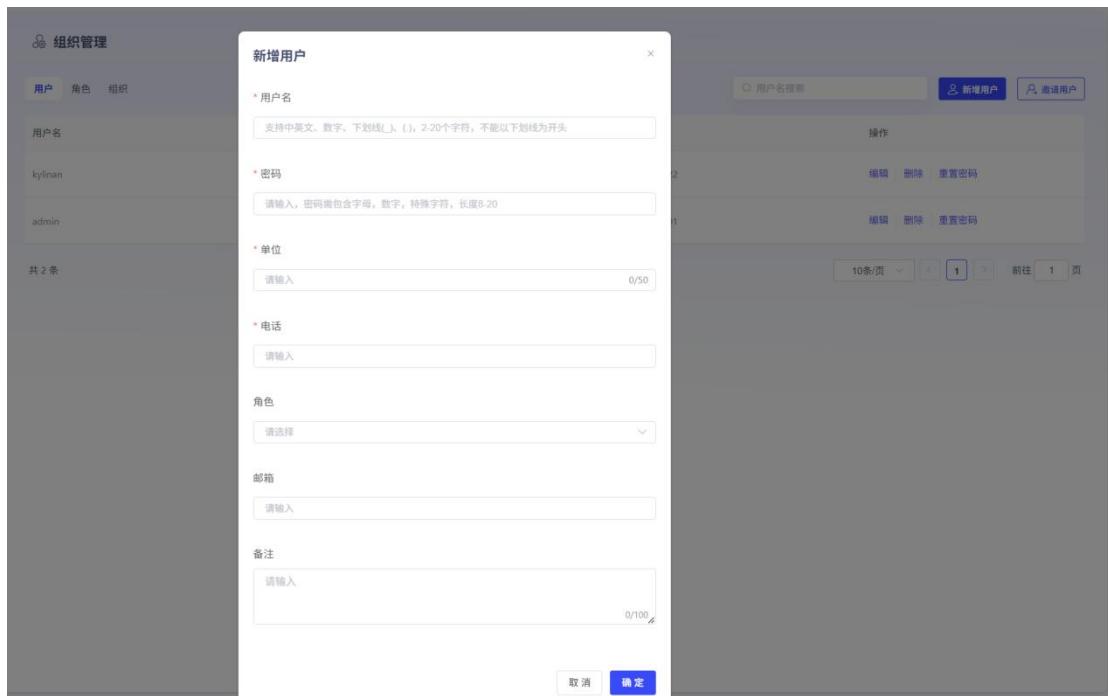
取消 确定

新增账户

点击“新增账户”，可对用户名、密码、单位、电话、角色、邮箱进行设置。

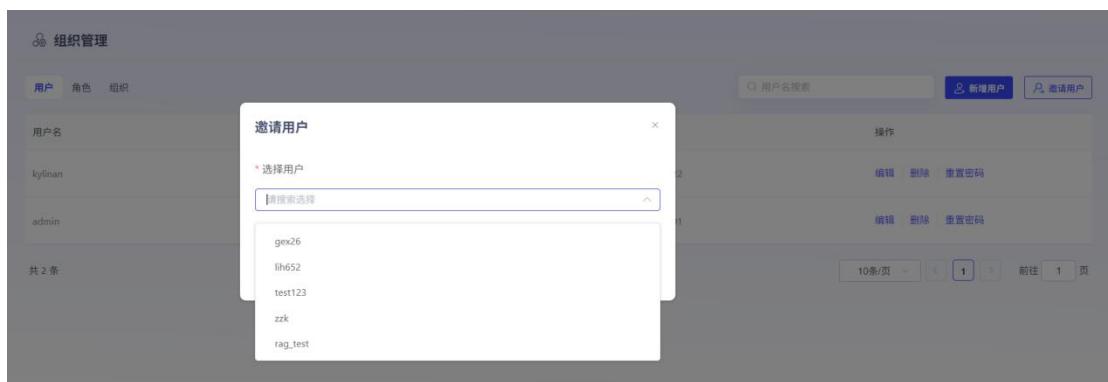
点击“确认”，即可完成新增用户。用户被分配到某个组织，可不对应任何角色。

用户所在组织中，对应的角色有创建用户的权限，则该用户创建的新用户必须默认属于该组织。



邀请用户

点击“邀请用户”，可对特定用户进行邀请。管理员可邀请该系统下任意不在该组织的用户，被邀请的用户自动加入该组织。



3、组织切换

平台支持多租户，用户可进行组织切换，查看不同组织下创建的应用，并进行编辑和使用。



Open API

文本问答API

文本生成应用无会话支持，适合用于翻译/文章写作/总结 AI 等等。

请求接口

Type	Instructions
方法	Http
请求URL	按照应用实际API根地址，例如： <code>http://localhost:8081/service/api/openapi/v1/rag/chat</code>
字符编码	UTF-8
请求类型	POST
鉴权方式(header参数)	<code>Authorization: Bearer {API Key}</code>
请求格式	<code>Content-Type: application/json</code>
响应格式	非流式： <code>Content-Type: application/json</code> 流式： <code>Content-Type: text/event-stream</code>

请求参数

Parameter	Required	Type	Instructions
stream	否	bool	是否以流式接口的形式返回数据，默认为非流式。true为流式，false为非流式。
query	是	string	用户提出的问题或提示语

响应参数

Parameter	Type	Instructions
code	int	状态码，用于表示请求成功或具体的错误类型
message	string	提示信息，通常用于提供关于code的详细解释或请求成功的确认信息
msg_id	string	提示信息ID
data	array	当前响应文本，包含了根据用户输入和知识库搜索得到的答案
history	array	包含之前对话历史的数组，用于上下文管理
finish	int	仅在流式返回中有该字段。表示流是否结束，0：未结束，1：正常结束，2：生成长度导致结束，3：异常结束，4：命中安全护栏结束

data

Parameter	Type	Instructions
output	string	当前响应文本内容片段
searchList	array	知识增强搜索结果

searchList

Parameter	Type	Instructions
kb_name	string	知识库名字
snippet	string	知识内容片段
title	string	文件标题

history

Parameter	Type	Instructions
query	string	请求文本
response	string	模型响应文本

调用示例

流式

```
curl -k --location 'http://localhost:8081/service/openapi/v1/rag/chat' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer <API Key>' \
--data '{
    "stream": true,
    "query": "请一句话介绍DeepSeek"
}'
```

非流式

```
curl -k --location 'http://localhost:8081/service/api/openapi/v1/rag/chat' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer <API Key>' \
--data '{
    "stream": false,
    "query": "请一句话介绍DeepSeek"
}'
```

响应示例

流式

```
data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "DeepSeek", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "是", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "深度求索", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "推出的", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "AI", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "工程化", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "平台", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "提供", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "模型", "searchList": []}, "history": [], "finish": 0}
```

```
data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "纳", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "管", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "、", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "工作", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "流", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "编排", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "、", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "知识", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "库", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "管理等", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "全套", "searchList": []}, "history": [], "finish": 0}
```

```
data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "功能", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": " ", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "支持", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "企业", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "高效", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "构建", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "智能化", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "应用", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "并", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "降低", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "AI", "searchList": []}, "history": [], "finish": 0}
```

```

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "技术", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "门槛", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": ". ", "searchList": []}, "history": [], "finish": 0}

data: {"code": 0, "message": "success", "msg_id": "bf10ca0882a09ec6a4e1e26190a841b2", "data": {"output": "", "searchList": []}, "history": [], "finish": 1}

```

非流式

```

{
  "code": 0,
  "message": "success",
  "msg_id": "f89327a67dc05fec500f009b2c605401",
  "data": {
    "output": "DeepSeek是深度求索推出的模块化AI工程化平台，提供从模型纳管到应用落地的完整工具链，支持多租户架构和企业级知识库等功能，帮助企业降低AI应用门槛并加速数字化转型。",
    "searchList": [
      {
        "kb_name": "DeepSeek",
        "title": "README.pdf",
        "snippet": "库建设、复杂工作流编排等完整功能体系的AI工程化平台。平台采用模块化架构设计，支持灵活的功能扩展和二次开发，在确保企业数据安全和隐私保护的同时，大幅降低了AI技术的应用门槛。无论是中小型企业快速构建智能化应用，还是大型企业实现复杂业务场景的智能化改造，DeepSeek都能提供强有力的技术支撑，助力企业加速数字化转型进程，实现降本增效和业务创新。 🔥 平台核心优势 ✅ 企业级工程化：提供从模型纳管到应用落地的完整工具链，解决LLM技术落地\"最后一公里\"问题 ✅ 开放开源生态：采用宽松友好的 Apache 2.0 License，支持开发者自由扩展与二次开发 ✅ 全栈技术支持：配备专业团队为生态伙伴提供 架构咨询、性能优化 全周期赋能 ✅ 多租户架构：提供多租户账号体系，满足用户成本控制、数据安全隔离、业务弹性扩展、行业定制\n化、快速上线及生态协同等核心需求 ► 核心功能模块 1. 模型纳管 (Model Hub) ► 支持 数百种专有/开源大模型（包括GPT、Claude、Llama等系列）的统一接入与生命周期管理 ► 深度适配 OpenAI API 标准及 DeepSeek 生态模型，实现异构模型的无缝切换 ► 提供 多推理后端支持 (VLLM、TGI等) 与 自托管解决方案，满足不同规模企业的算力需求 2. 可视化工作流 (Workflow Studio) ► 通过 低代码拖拽画布 快速构建复杂AI业务流程 ► 内置 条件分支、API、大模型、知识库、代码 等多种节点，支持端到端流程调试与性能分析 3. 企业级知识库、RAG Pipeline ► 提供 知识库创建→文档解析→向量化→检索→精排 的全流程知识管理能力，支持\npdf/docx/txt/xlsx/csv/pptx等 多种格式 文档，还支持网页资源的抓取和接入 ► 集成 多模态检索、级联切分 与 自适应切分，显著提升问答准确率 4. 智能体开发框架 (Agent Framework) ► 可基于 函数调用 (Function Calling) 的Agent构建范式，支持工具扩展、私域知识库关联与多轮对话 ► 支持 在线调试 5. 后端即服务 (BaaS) ► 提供 RESTful API，支持与企业现有系统 (OA/CRM/ERP等) 深度集成 ► 提供 细粒度权限控制，保障生产环境稳定运行"
    },
    {
      "kb_name": "DeepSeek",
      "title": "README.pdf",
      ...
    }
  ]
}

```

"snippet": "API + 应用程序导向 API + 应用程序导向 编程方法 ✅ ✅ 支持的
LLMs ✅ ✅ RAG引擎 ✅ ✅ Agent ✅ ✅ 工作流 ✅ ✅ 可观测性 ✅ ✅ 本地部署 ✅ ✕
license友好 ✅ ✕ 多租户 🚀 快速开始 Docker安装 从源码安装 ⚙️ 典型应用场景 智能客服：基于
RAG+Agent实现高准确率的业务咨询与工单处理知识管理；构建企业专属知识库，支持语义搜索与智能摘要生
成。流程自动化：通过工作流引擎实现合同审核、报销审批等业务的AI辅助决策平台已成功应用于 金融、工
业、政务 等多个行业，助力企业将LLM技术的理论价值转化为实际业务收益。我们诚邀开发者加入开源社区，
共同推动AI技术的民主化进程。许可证 DeepSeek根据Apache License 2.0发布。"
}
]
},
"history": [
{
"query": "请一句话介绍DeepSeek",
"response": "DeepSeek是深度求索推出的模块化AI工程化平台，提供从模型纳管到应用落
地的完整工具链，支持多租户架构和企业级知识库等功能，帮助企业降低AI应用门槛并加速数字化转型。"
}
],
"finish": 1
}

智能体创建对话API

智能体应用支持会话持久化，可将之前的聊天记录作为上下文进行回答，可适用于聊天/客服 AI 等。

请求接口

Type	Instructions
方法	Http
请求URL	按照应用实际API根地址，例如： <code>http://localhost:8081/service/api/openapi/v1/agent/conversation</code>
字符编码	UTF-8
请求类型	POST
鉴权方式(header 参数)	<code>Authorization: Bearer {API Key}</code>
请求格式	<code>Content-Type: application/json</code>
响应格式	非流式： <code>Content-Type: application/json</code> 流式： <code>Content-Type: text/event-stream</code>

请求参数

Parameter	Required	Type	Instructions
title	是	string	对话标题

响应参数

Parameter	Type	Instructions
code	int	状态码，用于表示请求成功或具体的错误类型
msg	string	提示信息，通常用于提供关于code的详细解释或请求成功的确认信息
data	array	当前响应文本，包含了根据用户输入和知识库搜索得到的答案

data

Parameter	Type	Instructions
conversation_id	string	当前响应文本片段ID

调用示例

```
curl -k --location
'http://localhost:8081/service/api/openapi/v1/agent/conversation' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer <API Key>' \
--data '{
    "title": "你好，DeepSeek"
}'
```

响应示例

```
{
  "code": 0,
  "data": {
    "conversation_id": "56"
  },
  "msg": ""
}
```

智能体对话API

智能体应用支持会话持久化，可将之前的聊天记录作为上下文进行回答，可适用于聊天/客服AI等。

请求接口

Type	Instructions
方法	Http
请求URL	按照应用实际API根地址，例如： <code>http://localhost:8081/service/api/openapi/v1/agent/chat</code>
字符编码	UTF-8

Type	Instructions
请求类型	POST
鉴权方式(header参数)	Authorization: Bearer {API Key}
请求格式	Content-Type: application/json
响应格式	非流式: Content-Type: application/json 流式: Content-Type: text/event-stream

请求参数

Parameter	Required	Type	Instructions
conversation_id	是	string	历史响应文本片段ID
stream	否	bool	是否以流式接口的形式返回数据，默認為非流式。 true为流式，false为非流式
query	是	string	用户提出的问题或提示语

响应参数

Parameter	Type	Instructions
code	int	状态码，用于表示请求成功或具体的错误类型
message	string	提示信息，通常用于提供关于code的详细解释或请求成功的确认信息
gen_file_url_list	array	模型生成输出的文件url列表
response	string	当前响应文本，包含了根据用户输入和知识库搜索得到的答案
search_list	array	知识增强搜索结果
history	array	包含之前对话历史的数组，用于上下文管理
usage	array	token使用量
finish	int	仅在流式返回中有该字段。表示流是否结束，0: 未结束，1: 正常结束，2: 生成长度导致结束，3: 异常结束，4: 命中安全护栏结束

gen_file_url_list

Parameter	Type	Instructions
output_file_url	string	输出文件url

history

Parameter	Type	Instructions
query	string	请求文本
response	string	模型响应文本

search_list

Parameter	Type	Instructions
kb_name	string	知识库名字
snippet	string	知识内容片段
title	string	文件标题

usage

Parameter	Type	Instructions
prompt_tokens	int	输入提示词token数
completion_tokens	int	输出文本token数
total_tokens	int	输入加输出总的token数

调用示例

流式

```
curl -k --location 'http://localhost:8081/service/api/openapi/v1/agent/chat' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer <API Key>' \
--data '{
    "stream": true,
    "conversation_id": "56",
    "query": "请一句话介绍DeepSeek"
}'
```

非流式

```
curl -k --location 'http://localhost:8081/service/api/openapi/v1/agent/chat' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer <API Key>' \
--data '{
    "stream": false,
    "conversation_id": "56",
    "query": "请一句话介绍DeepSeek"
}'
```

响应示例

流式

```
data: {"code": 0, "message": "success", "response": "DeepSeek",  
"gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens":  
0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "是", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "深度求索",  
"gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens":  
0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "推出", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "的", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "AI", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "工程", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "化", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "平台", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "", "gen_file_url_list":  
[], "history": [], "finish": 0, "usage": {"prompt_tokens": 0,  
"completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "提供", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "从", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "模型", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "纳", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "管", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "到", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "应用", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "落", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "地的", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "完整", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "工具", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "链", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "支持", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "企业", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "级", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "AI", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "应用的", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "快速", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "构建", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "与", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "智能化", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

```
data: {"code": 0, "message": "success", "response": "改造", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "【", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "1", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "^\n", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "】", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": ".\n", "gen_file_url_list": [], "history": [], "finish": 0, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}

data: {"code": 0, "message": "success", "response": "", "gen_file_url_list": [], "history": [], "finish": 1, "usage": {"prompt_tokens": 0, "completion_tokens": 0, "total_tokens": 0}, "search_list": [], "qa_type": [1]}
```

非流式

```
{
  "code": 0,
  "message": "success",
  "response": "DeepSeek是深度求索推出的AI工程化平台，提供从模型纳管到应用落地的完整工具链，支持企业级知识库、RAG Pipeline、智能体开发等功能，助力企业数字化转型【1^】【2^】。",
  "gen_file_url_list": [
    ],
  "search_list": [
    {
      "kb_name": "DeepSeek",
      "title": "README.pdf",
      "content": "DeepSeek是一个由深度求索公司开发的AI工程化平台。它旨在帮助企业将复杂的AI模型纳入其业务流程中，并实现从模型部署到实际应用的无缝衔接。该平台集成了多种先进功能，包括但不限于RAG Pipeline（结合语义搜索和生成模型）以及智能体开发工具，从而帮助企业快速响应市场变化，提升运营效率。"
    }
  ]
}
```

"snippet": "库建设、复杂工作流编排等完整功能体系的AI工程化平台。平台采用模块化架构设计，支持灵活的功能扩展和二次开发，在确保企业数据安全和隐私保护的同时，大幅降低了AI技术的应用门槛。无论是中小型企 业快速构建智能化应用，还是大型企业实现复杂业务场景的智能化改造，DeepSeek都能提供强有力的技术支撑，助力企业加速数字化转型进程，实现降本增效和业务创新。 🔥 平台核心优势 ✓ 企业级工程化：提供从模型纳管到应用落地的完整工具链，解决LLM技术落地\"最后一公里\"问题 ✓ 开放开源生态：采用宽松友好的 Apache 2.0 License，支持开发者自由扩展与二次开发 ✓ 全栈技术支持：配备专业团队为生态伙伴提供 架构咨询、性能优化 全周期赋能 ✓ 多租户架构：提供多租户账号体系，满足用户成本控制、数据安全隔离、业务弹性扩展、行业定制\n化、快速上线及生态协同等核心需求 ► 核心功能模块 1. 模型纳管（Model Hub） ► 支持 数百种专有/开源大模型（包括GPT、Claude、Llama等系列）的统一接入与生命周期管理 ► 深度适配 OpenAI API 标准 及 DeepSeek 生态模型，实现异构模型的无缝切换 ► 提供 多推理后端支持（vLLM、TGI等）与 自托管解决方案，满足不同规模企业的算力需求

2. 可视化工作流（Workflow Studio） ► 通过 低代码拖拽画布 快速构建复杂AI业务流程 ► 内置条件分支、API、大模型、知识库、代码 等多种节点，支持端到端流程调试与性能分析 3. 企业级知识库、RAG Pipeline ► 提供 知识库创建→文档解析→向量化→检索→精排 的全流程知识管理能力，支持\npdf/docx/txt/xlsx/csv/pptx等 多种格式 文档，还支持网页资源的抓取和接入 ► 集成 多模态检索、级联切分 与 自适应切分，显著提升问答准确率 4. 智能体开发框架（Agent Framework） ► 可基于 函数调用（Function calling） 的Agent构建范式，支持工具扩展、私域知识库关联与多轮对话 ► 支持 在线调试 5. 后端即服务（BaaS） ► 提供 RESTful API ，支持与企业现有系统（OA/CRM/ERP等）深度集成 ► 提供 细粒度权限控制，保障生产环境稳定运行 "

},
{
 "kb_name": "DeepSeek",
 "title": "README.pdf",
 "snippet": "API + 应用程序导向 API + 应用程序导向 编程方法 ✓ ✓ 支持的LLMs
✓ ✓ RAG引擎 ✓ ✓ Agent ✓ ✓ 工作流 ✓ ✓ 可观测性 ✓ ✓ 本地部署 ✓ ✗ License
友好 ✓ ✗ 多租户 ✏ 快速开始 Docker安装 从源码安装 ⚡ 典型应用场景 智能客服：基于
RAG+Agent实现高准确率的业务咨询与工单处理 知识管理：构建企业专属知识库，支持语义搜索与智能摘
要生成 流程自动化：通过工作流引擎实现合同审核、报销审批等业务的AI辅助决策 平台已成功应用于
金融、工业、政务 等多个行业，助力企业将LLM技术的理论价值转化为实际业务收 益。我们诚邀开发者加入开
源社区，共同推动AI技术的民主化进程。 🔑 许可证 DeepSeek根据Apache License 2.0发布。"
}
],
"history": [
{
 "query": "请一句话介绍DeepSeek",
 "response": "DeepSeek是深度求索推出的AI工程化平台，提供从模型纳管到应用落地的完
整工具链，支持企业级知识库、RAG Pipeline、智能体开发等功能，助力企业数字化转型【1】【2】。"
}
],
"usage": {
 "completion_tokens": 0,
 "prompt_tokens": 0,
 "total_tokens": 0
},
"finish": 1
}