

# Agents 平台帮助文档

V1

# 欢迎来到Agents平台帮助中心

在此快速了解平台相关功能和使用技巧。

## 目录

快速开始 .....	4
手动搭建第一个 Agent .....	9
认识知识库 .....	15
知识库简介 .....	15
创建知识库 .....	16
导入数据 .....	18
Agents 基本信息配置 .....	24
使用场景 .....	24
界面功能 .....	25
Agents 规划 - 页面说明 .....	28
界面功能 .....	28
Agent 规划 - 编排规则 .....	32
模块结构 .....	32
节点说明 .....	33
Agents 规划 - 模块说明 .....	34
用户提问（入口） .....	34
智能对话 .....	38
确定回复 .....	41
信息分类 .....	44
字段提取 .....	48
信息加工 .....	52
知识库搜索 .....	55
文档审核 .....	58
图片提问 .....	61

文档提问 .....	64
关键词识别.....	68
Agent 对话结束 .....	71
代码块.....	74
信息处理器.....	77
For Each.....	79
HTTP 调用.....	82
进阶功能.....	84
模块编辑技巧.....	84
内置函数/变量说明.....	87
1、系统内置变量 .....	87
2、系统内置函数 .....	87
编排规则及案例.....	95
入门：角色扮演机器人.....	95
初级：知识库 Agent.....	99
中级：信息分类 .....	102
高级：字段提取+ HTTP 模块.....	104

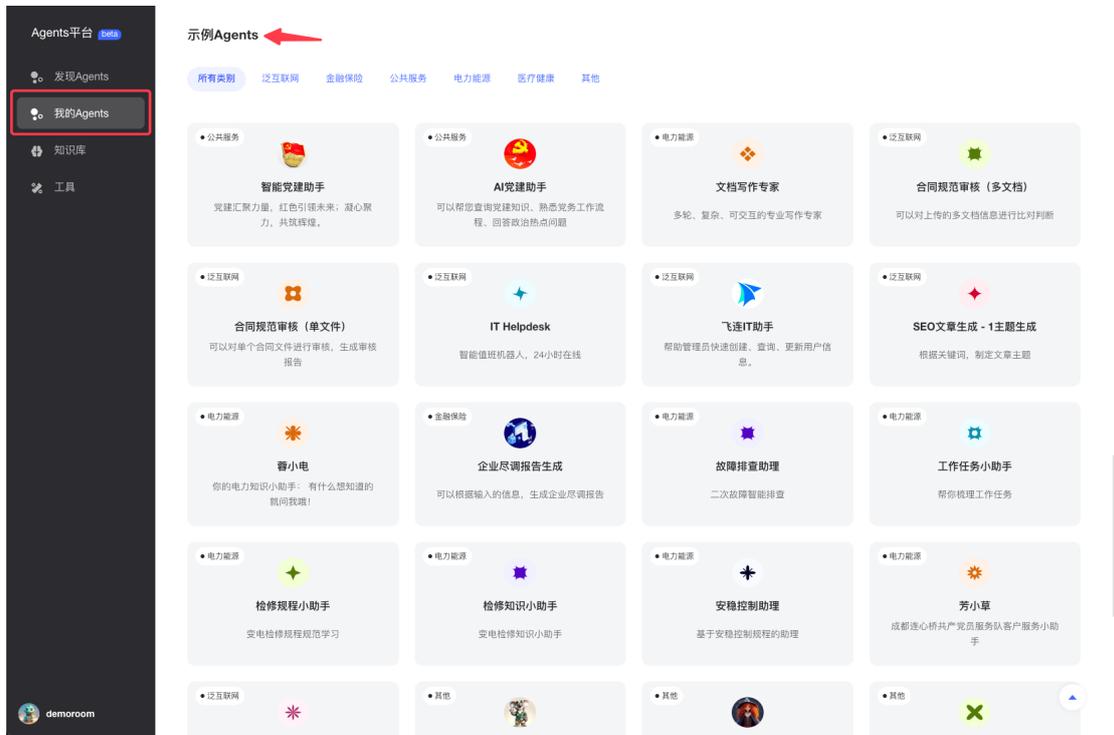
# 快速开始

## 体验示例 Agents

在 Agents 平台中，已经预置一些示例 Agents 便于大家进行学习和体验，涵盖多个行业和场景，可以根据分类标签选择自己感兴趣的示例 Agents 进行体验。

## 示例 Agents

在平台左侧菜单“我的 Agents”，页面向下滑动，即可找到“示例 Agents”模块

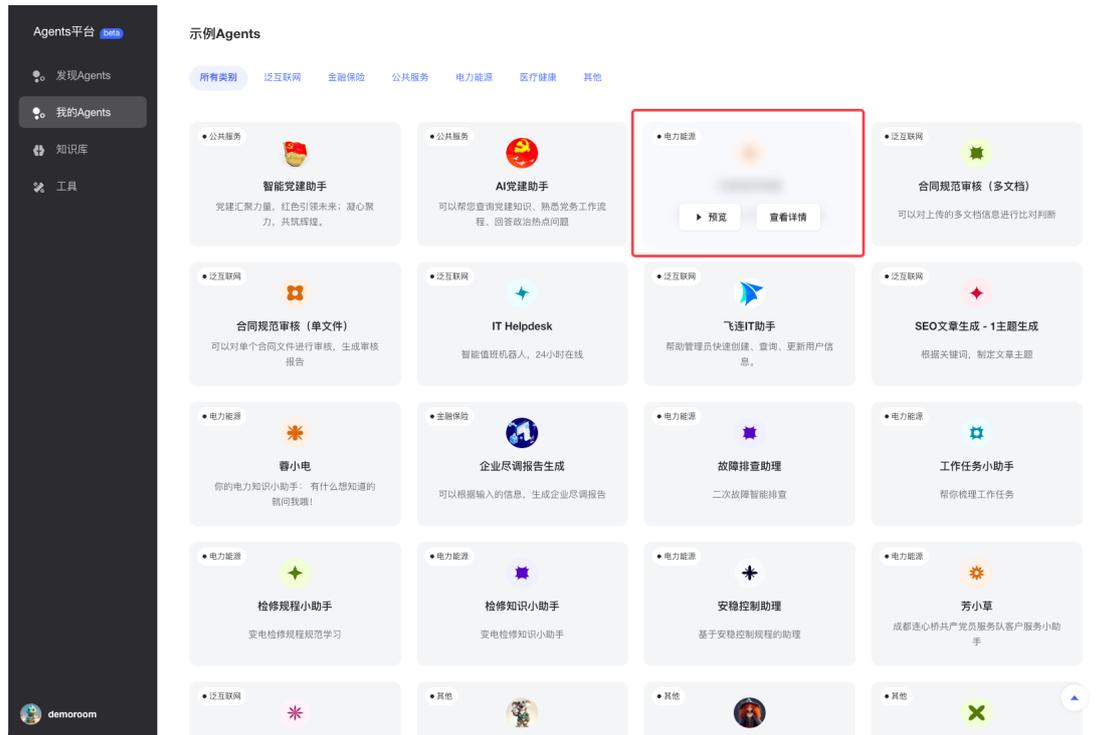


## 体验示例 Agents

将鼠标放在任意示例 Agents 的卡片上，会弹出两个选项，他们的功能如下：

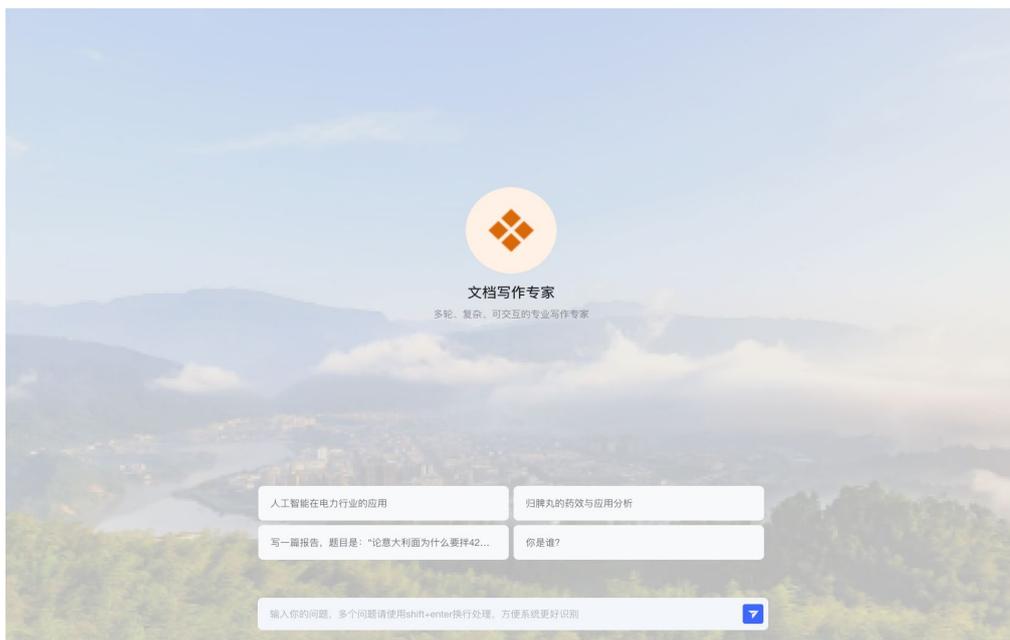
- **预览：**弹出新窗口，与该 Agents 进行对话，体验该 Agents 的功能
- **查看详情：**进入该 Agents 的配置页面，查看该 Agents 配置信息和规划，了解背

## 后的运行逻辑



## 预览

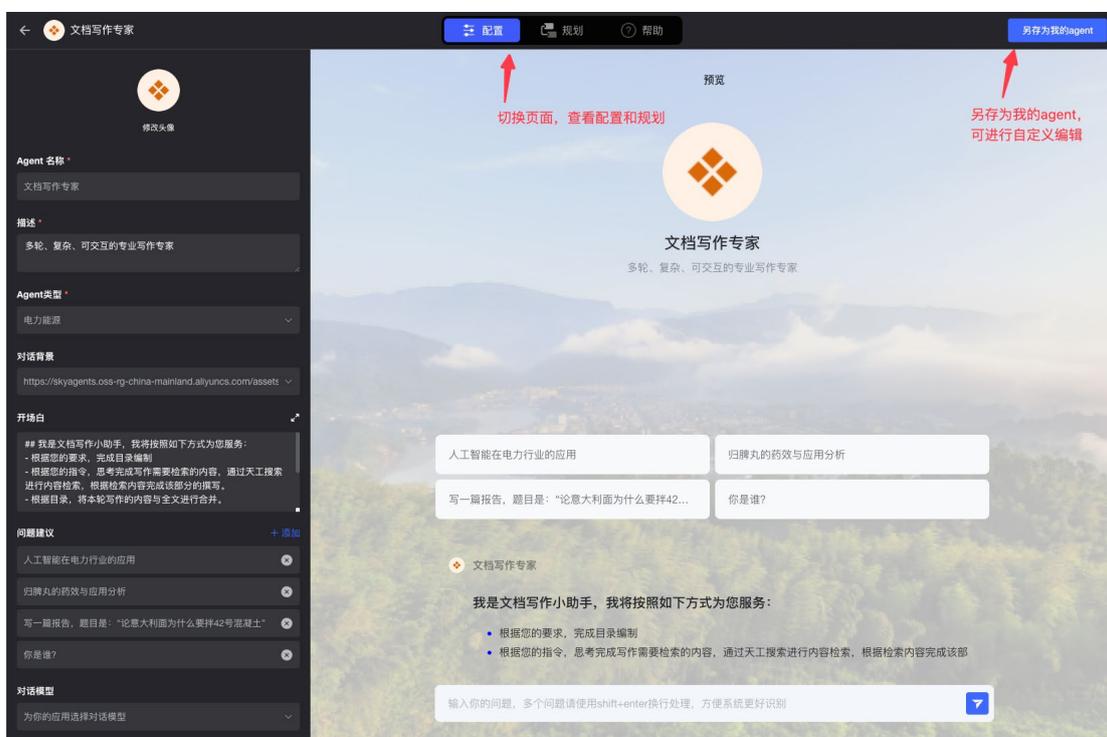
进入预览界面后，根据当前示例 Agents 的能力，可以在下方输入框中输入问题，或点击快捷问题，与 Agents 进行对话。



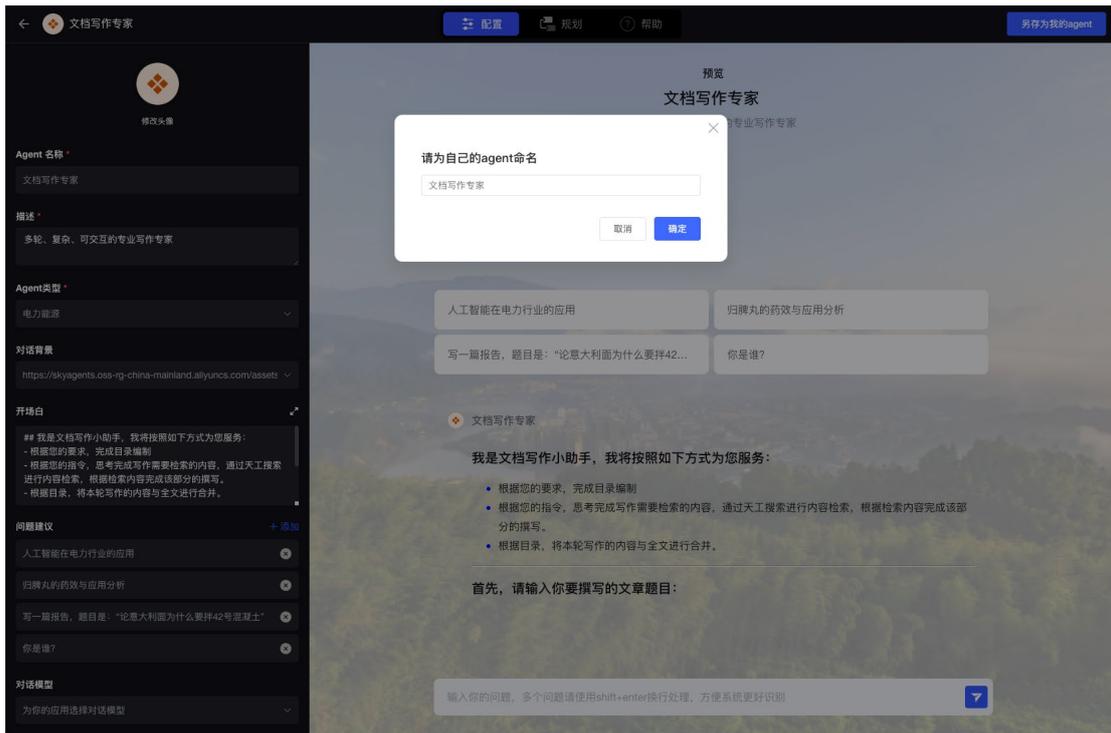
## 查看详情

在查看详情页面，我们可以进一步查看该示例 Agents 的配置信息，学习这个 Agents 的搭建方式。

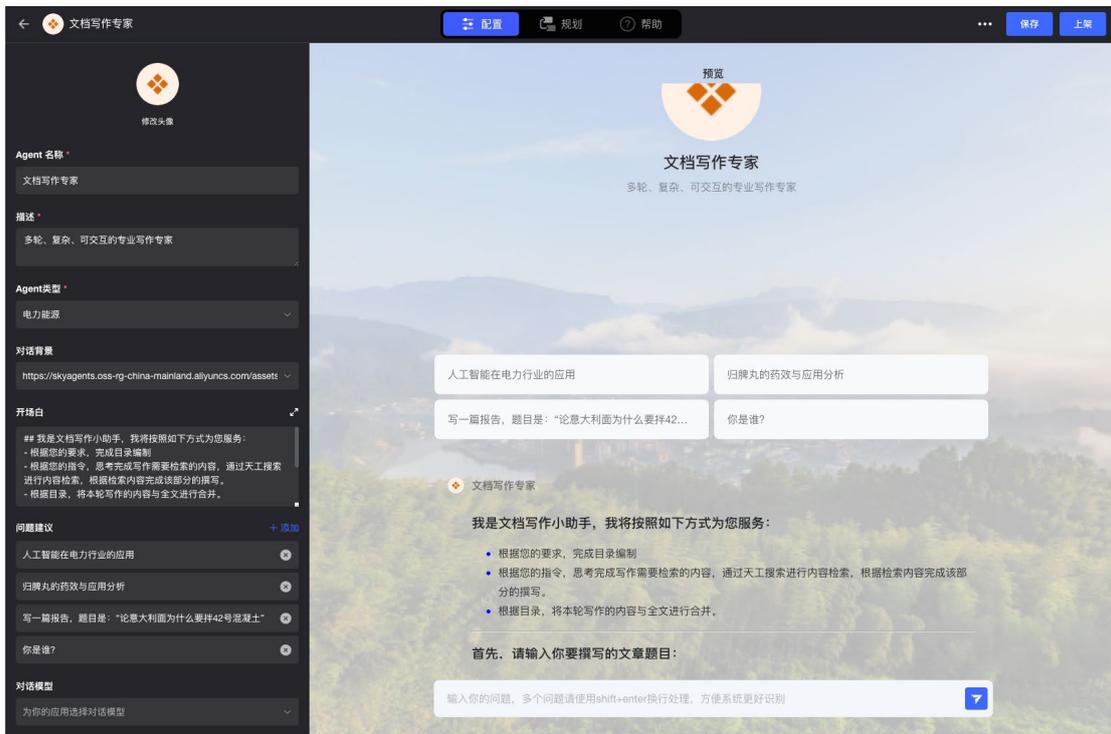
- 通过顶部菜单切换查看基本信息“配置”和“规划”编排
- 通过右上角“另存为我的 agent”，将该 Agents 复制到我的 Agents 列表中，可进行自定义编辑和修改



复制时，需要为自己的 Agent 进行命名。在设定名字后，我们就完成了第一个 Agent 的创建。



## 对 Agent 进行命名

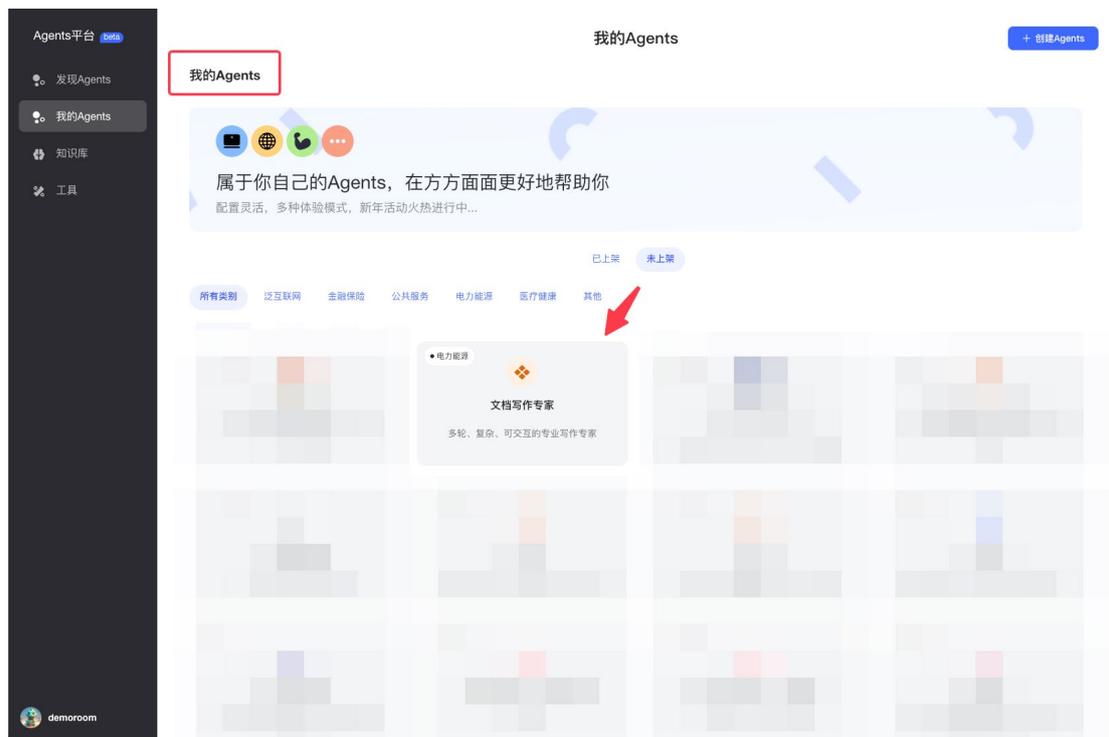


## 可以自己进行设置的修改

## Tips

复制完的 Agents 可以在哪里找到呢？

- 在首页“我的 Agents”中，就能找到刚才复制的 Agent 的卡片哦！



# 手动搭建第一个 Agent

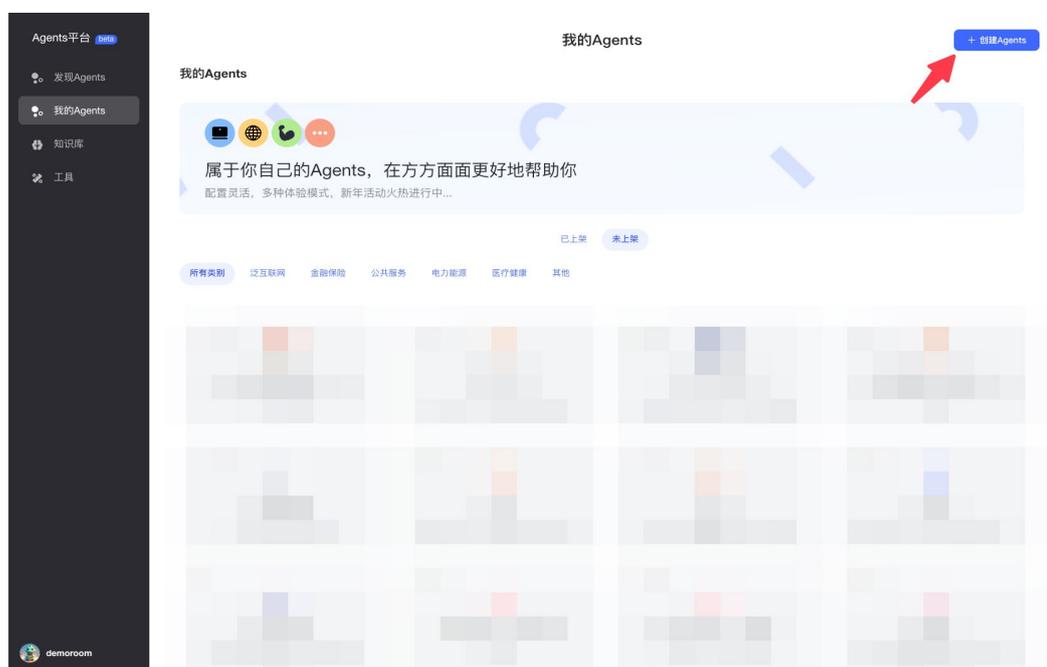
## 前言

Hi, 欢迎来到 Agents 平台, 在这个教程中, 我们将和你一起搭建第一个专属 Agent, 帮助您快速熟悉 Agents 平台, 并感受 AI 所带来的强大实力。整个流程大约 3 分钟左右, 即可将你的创意思法变成可以实时对话交流的 AI Agents, 我们开始吧。

## 第一步: 创建 Agent

### 进入 Agent 主页

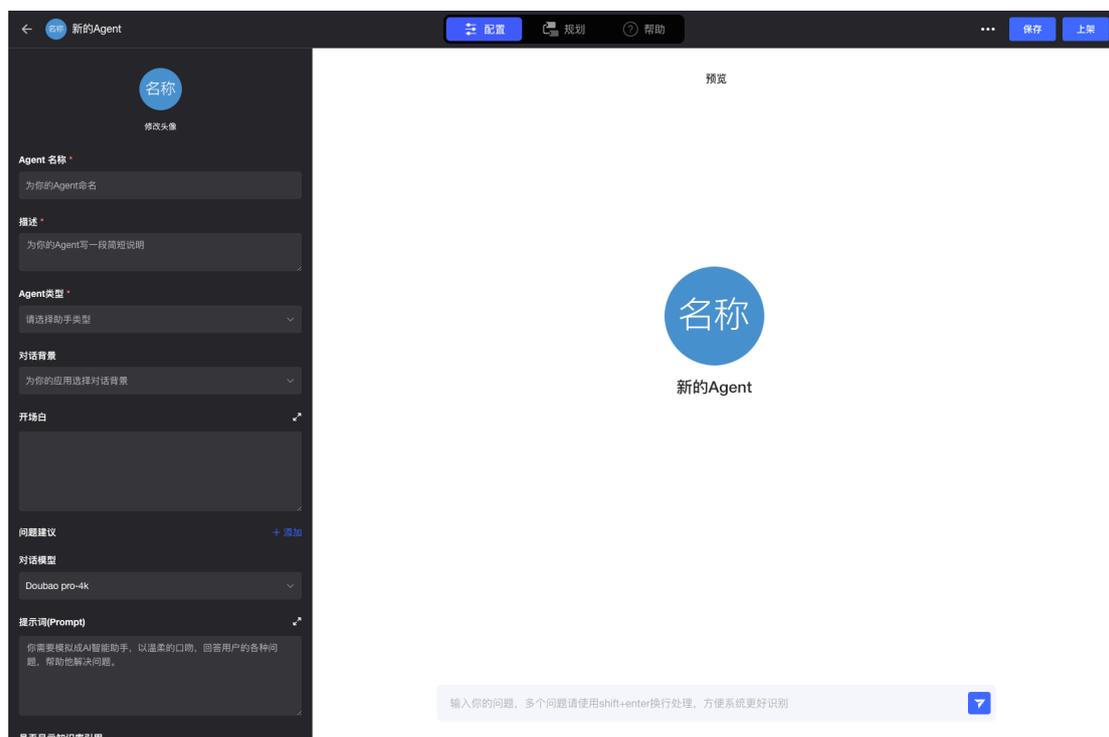
请点击页面右侧菜单[我的 Agents], 进入 Agents 主页。选择[创建 Agents]进入创建流程。



Agent 是用户和目标想法之间的关键桥梁, 它承担起指令接收、分析、执行和对话的工作, Agent 也是平台核心的功能

## Agent 配置

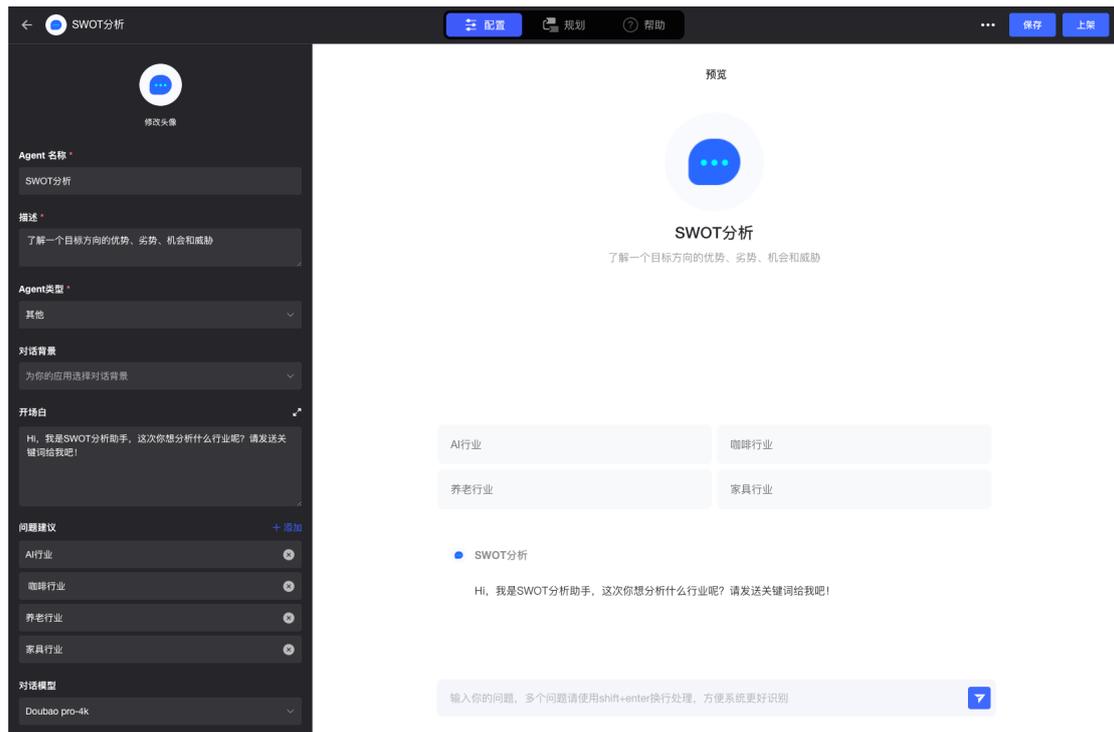
点击上方 [配置]，进入 Agent 配置页面。Agent 支持以所见即所得的方式进行配置，左侧为配置信息，右侧为预览窗口。



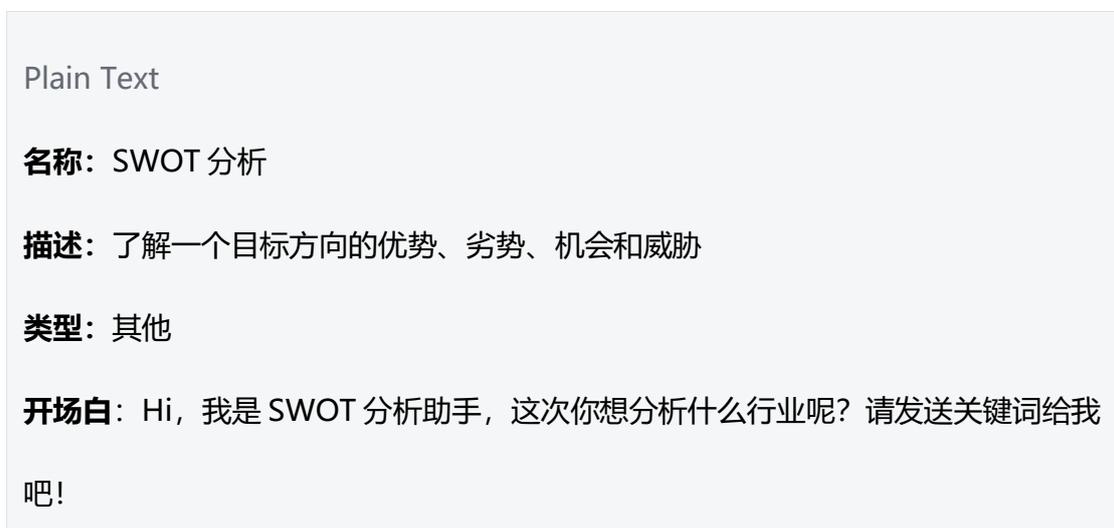
配置 Agent 的相关参数，比如[Agent 名称][描述][问题建议][对话模型][提示词][知识库选择]等等。此时我们需要为 AI Agent 设置如下信息：

- **名称：**为 Agent 起一个名字
- **描述：**为你的 Agent 写一段简短说明
- **类型：**选择合适的类型，这里我们选择“其他”
- **对话背景：**自定义选择即可
- **开场白：**与 Agents 对话前的提示，自定义选择即可
- **问题建议：**暂时不需要设置

- **对话模型：**默认
- **提示词：**使用提示词（Prompt）定义这个 Agent 所实现的目标，这里请发挥你天马行空的想象力，给 Agent 赋予相应的能力。
- **知识库：**本次我们暂时不需要用到知识库，这里不需要勾选



比如这次想让我的 Agent 作为一个“SWOT 分析”小助手的形象来和我对话，我设置了以上信息来对 Agent 进行配置。



**问题建议：** 设置了四个行业关键词

**对话模型：** 豆包大模型

**提示词：**

你是商业领域的专家，可以对指定的主题进行优势、劣势、机会和威胁进行梳理和评估，最后给出综合判断建议和企业的可行性发展方向。

请根据我的主题，输出对应的分析结果，输出格式如下：

【优势】

【劣势】

【机会】

【威胁】

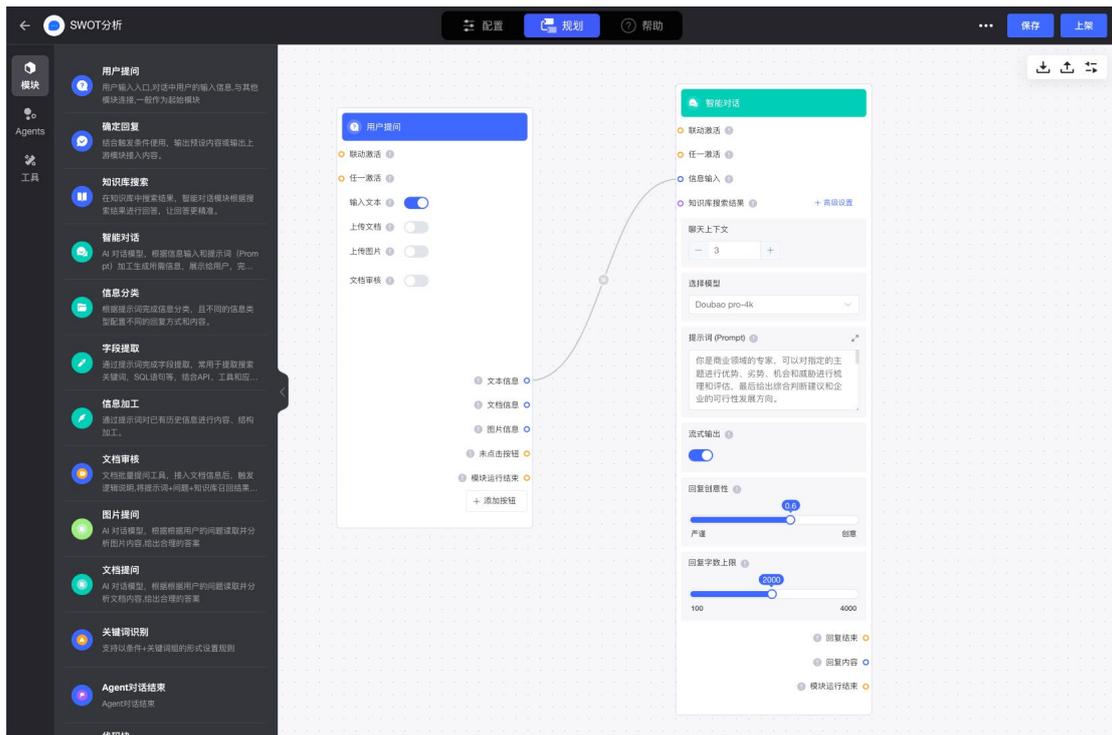
【综合判断建议】

【企业的可行性发展方向】

## 规划连线

点击上方 **[规划]**，进入 Agent 规划页面，此页面用来规划 Agent 的运行流程，可以通过拖拽的方式将需要使用的模块拖入画布，并连接节点，使其顺利运行。

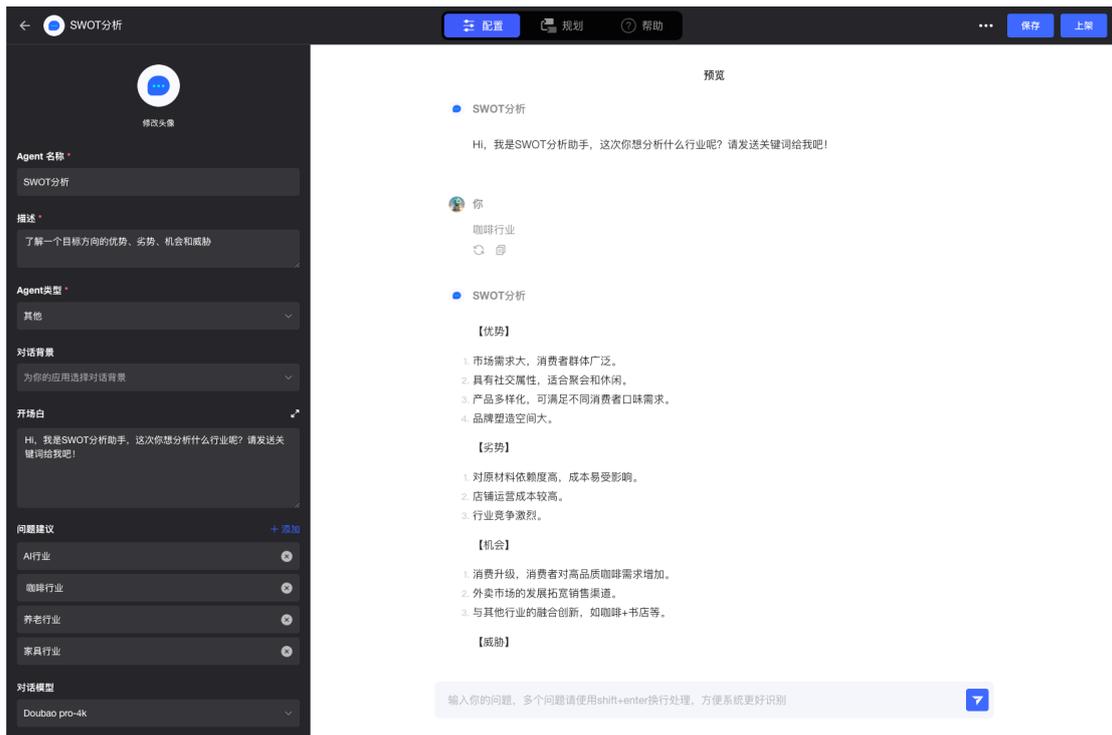
在此次示例中，Agent 已经将左侧 **[文档信息]**节点，与右侧 **[信息输入]**连接起来。这个连线的含义是，用户提问的文本问题，将输出至智能对话模块，接下来就由智能对话模块进行信息处理，并回复到对话中，如上图所示。



以上，我们就完成了一个 AI Agent 的基本信息设置，现在，从这个 Agent 已经具备我们指定的知识以及 AI 能力，点击右上角[保存]。

## 第二步：对话测试

这样我们的第一个 Agent 就创建完成了，我们在右侧的对话预览界面，就可以输入内容和 Agent 开始对话了，请尝试下你的专属 Agent 。在对话过程中，如果对 Agent 的对话有更多想法，可以随时调整 提示词 (Prompt) 的设计，以最终达到所期望的效果。



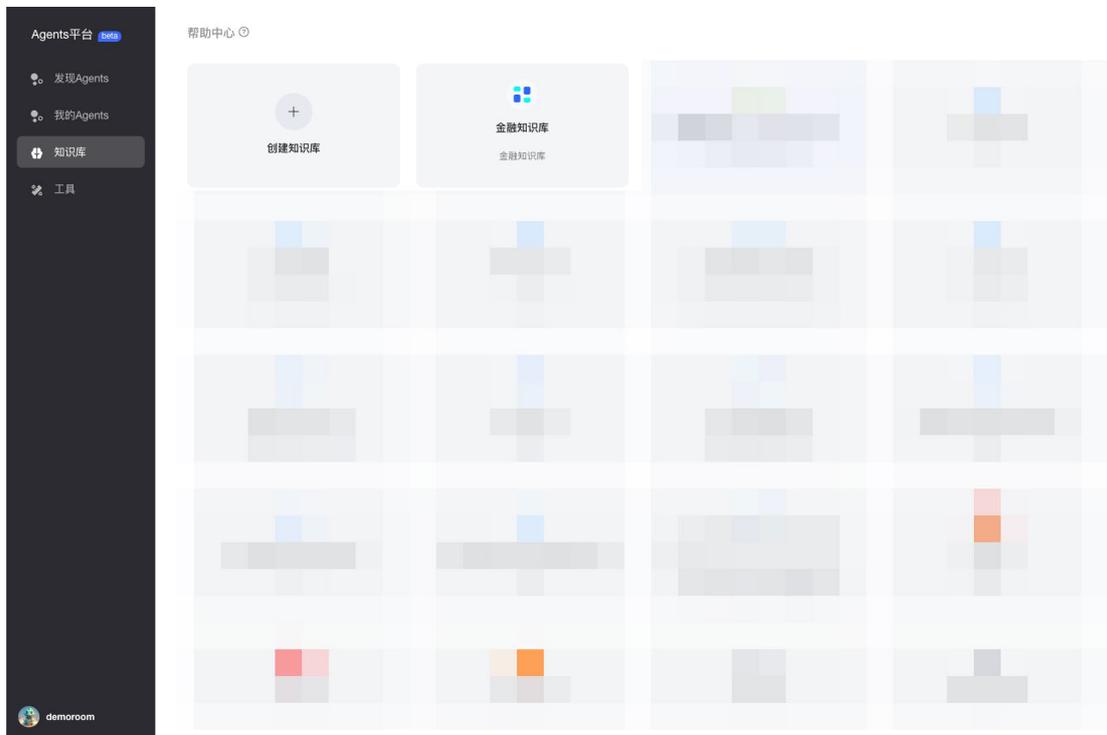
## 配置完成

至此，我们就完成了一个[Agent]搭建，平台拥有非常灵活的搭建能力，结合大模型助力将想法变为提升效率的智能体。接下来你可以浏览更多关于平台的详细功能介绍，更加深入感受 AI 所带的更多可能性。

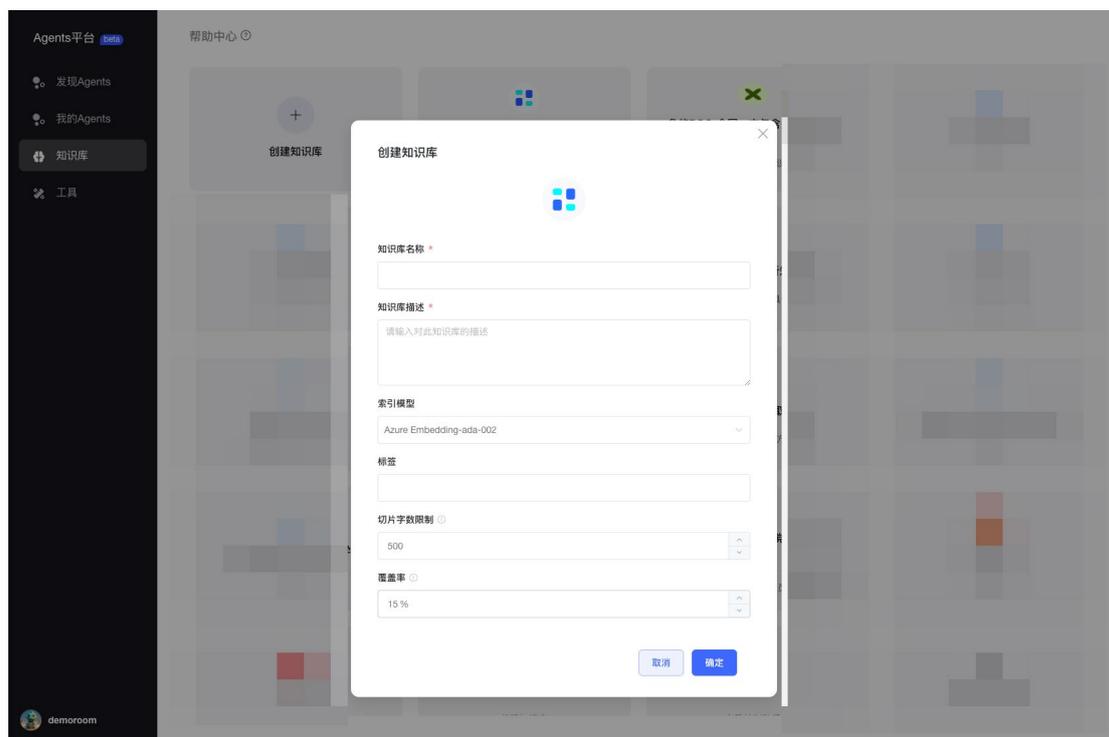
# 认识知识库

## 知识库简介

知识库是作为 Agent 内容输入的重要来源，我们可以在此统一管理和维护企业所有的知识库，在 Agents 规划过程中，通过知识库的内容检索，弥补大模型对于专业知识的不足，并增加回复准确率。如常见的客服问答场景、产品 QA 场景等等，都需要有一个统一的回复口径来传达给用户，可借助知识库进行回复引导。



# 创建知识库



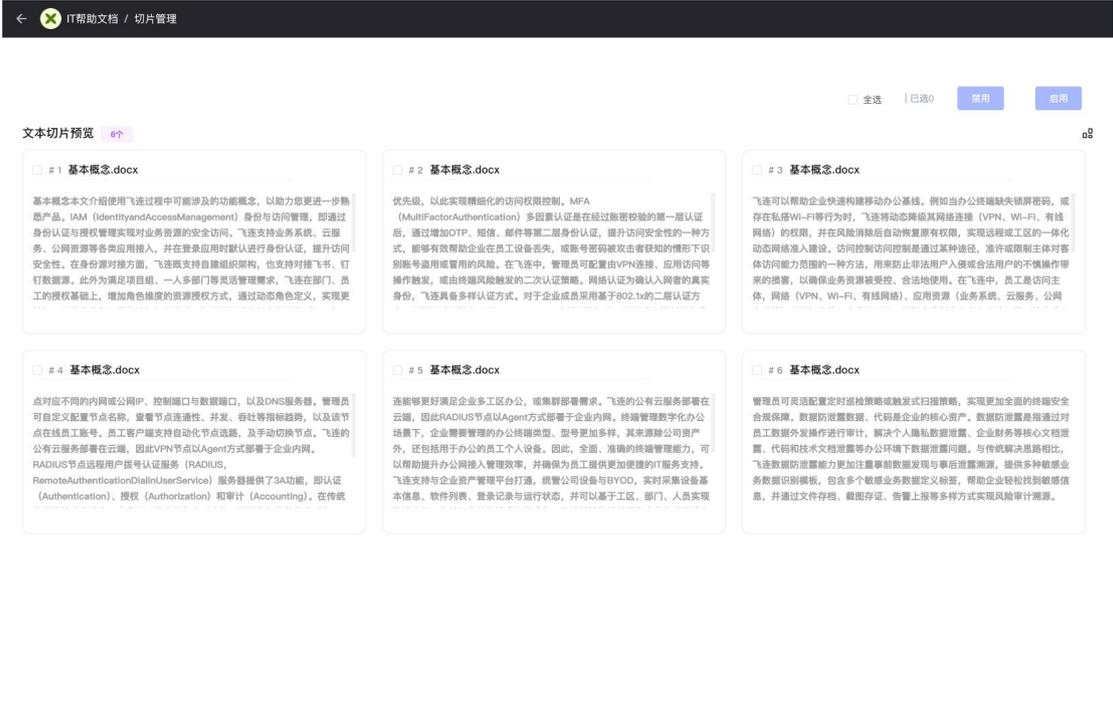
当我们准备新建一个知识库时，需要先对知识库设定一些相关信息，这里包括以下几点：

- **知识库头像**
- **知识库名称**
- **知识库描述**：对该知识库的内容/作用进行简单描述
- **索引模型**：通过索引模型，可以将知识库内容进行向量化处理，这种向量化表示方法使得这些实体可以在数学上进行操作，如计算相似度、距离等，这有助于对知识库内容的搜索和召回。
- **标签**：为知识库添加相关标签，便于后续分类整理
- **切片字数限制**：知识库处理时，需要针对文档进行切片处理，此处设置文档的单

## 个切片段落字数

- **覆盖率：**前后两个切片间重叠的字数，减少因切片导致某些内容被分割在两个切片里的情况

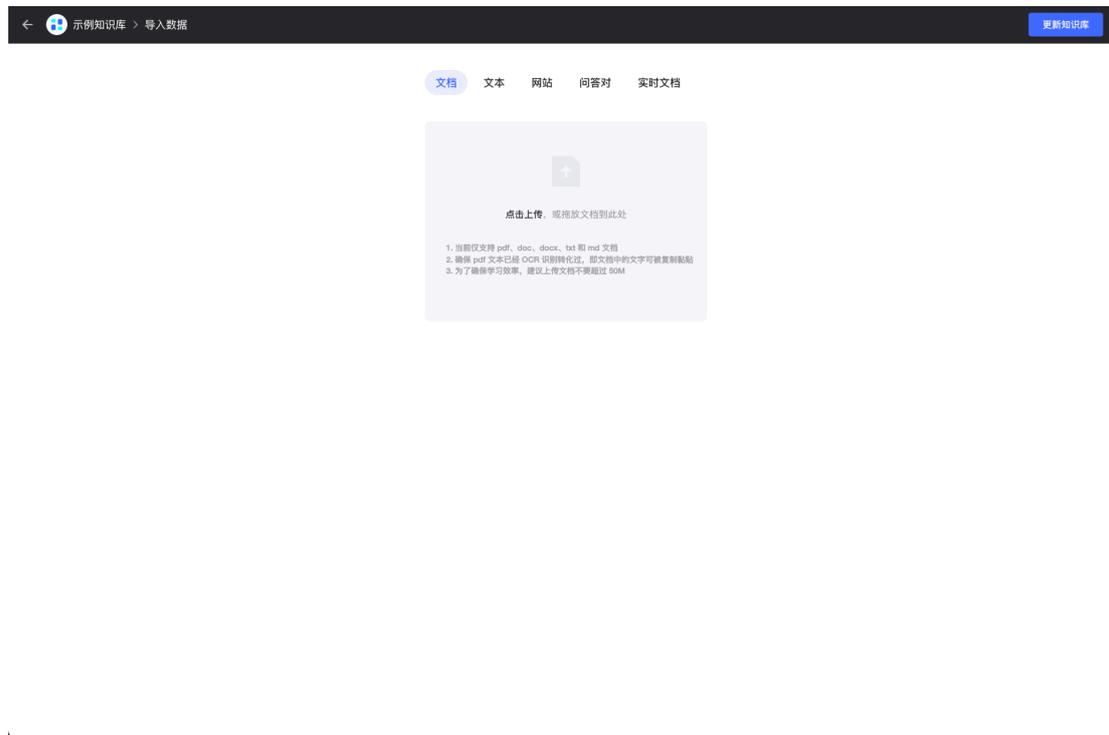
 知识库切片示例：一篇文档会被进行切片处理，处理成如下多个切片。



The screenshot shows a web interface for document management. At the top, there's a breadcrumb 'IT帮助文档 / 切片管理'. Below it, there are filters for '全选', '已选0', '禁用', and '启用'. The main area is titled '文本切片预览' and displays six preview cards for a document named '基本概念.docx'. Each card shows a snippet of text from the document, illustrating how it is divided into overlapping segments for better coverage.

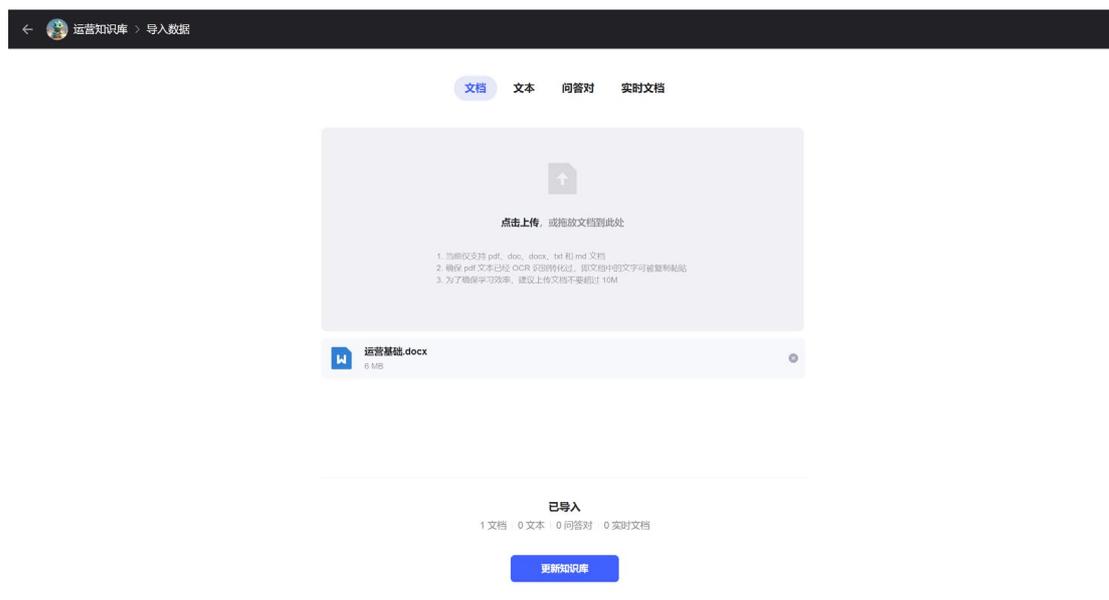
## 切片示例

# 导入数据



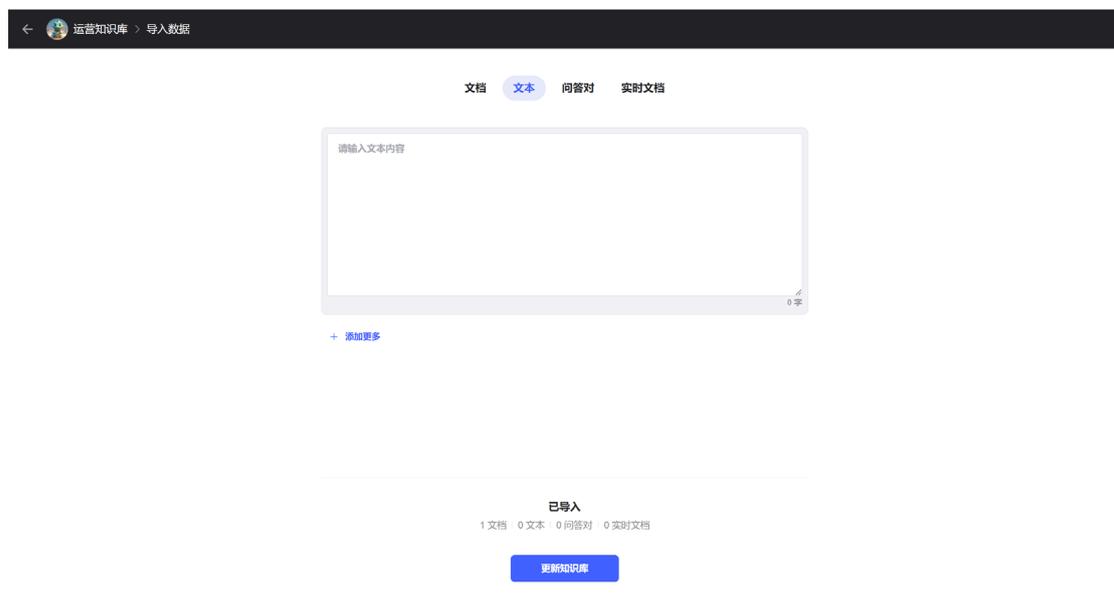
目前，平台支持 4 种导入方式，分别是文档，文本，网站和问答对，未来还将支持实时文档等上传方式。

# 文档导入



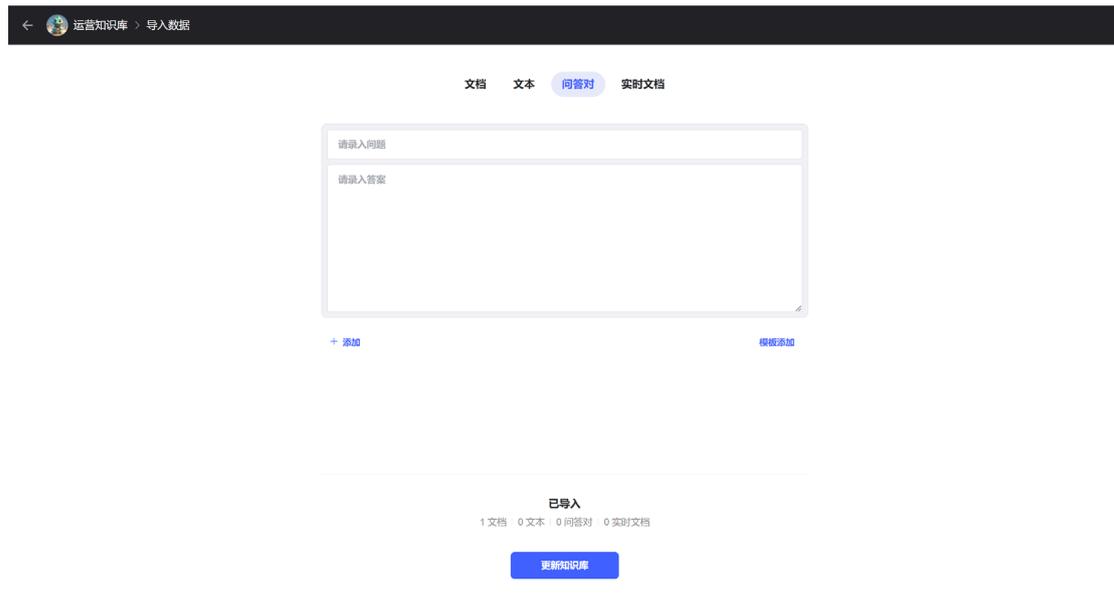
- 文档导入目前支持结尾格式为 .pdf、.doc、.docx、.txt 和 .md 格式的文档
- 其中 pdf 文档导入需要注意，文档内文字可复制，扫描版的 pdf 请通过下方“OCR 识别”方式导入
- 文件大小限制：为了确保机器检索效率，建议上传文档不要超过 10MB

## 文本导入



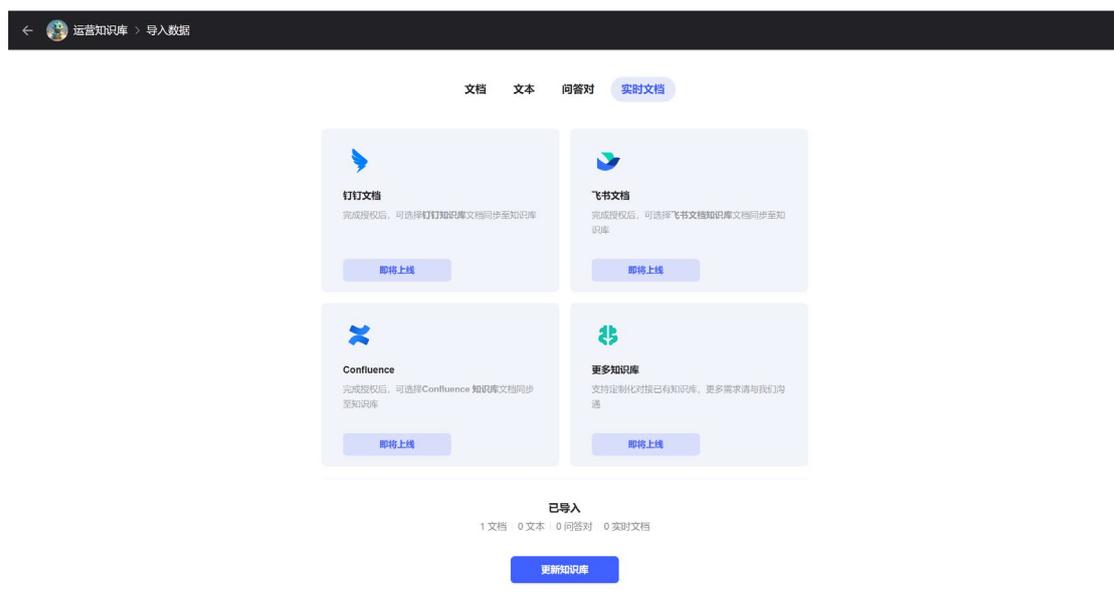
文本导入比较方便快捷，即直接将需要导入的知识数据直接粘贴在输入框即可。建议将已有文本分切成多个文本片段，通过选择“添加更多”添加多个文本内容，便于进行文本搜索和引用。

# 问答对



问答对是一种更加简单清晰的方式添加知识库内容，以一问一答的方式，将人工筛选好的问题更新到数据库中。问答对可以通过输入的方式依次录入，也可以通过上传文档或链接的方式，依靠平台进行提取。

# 实时文档（即将上线）



针对有在线协作的企业，平台还支持实时文档的导入，无需将现有的文档进行导出，通过简单的授权，即可将现有的多个文档直接进行同步更新，提升文档使用效率。

## OCR 导入



当前仅支持 pdf 文档，可通过 OCR 方式解析 pdf 中的文字，并导入知识库。

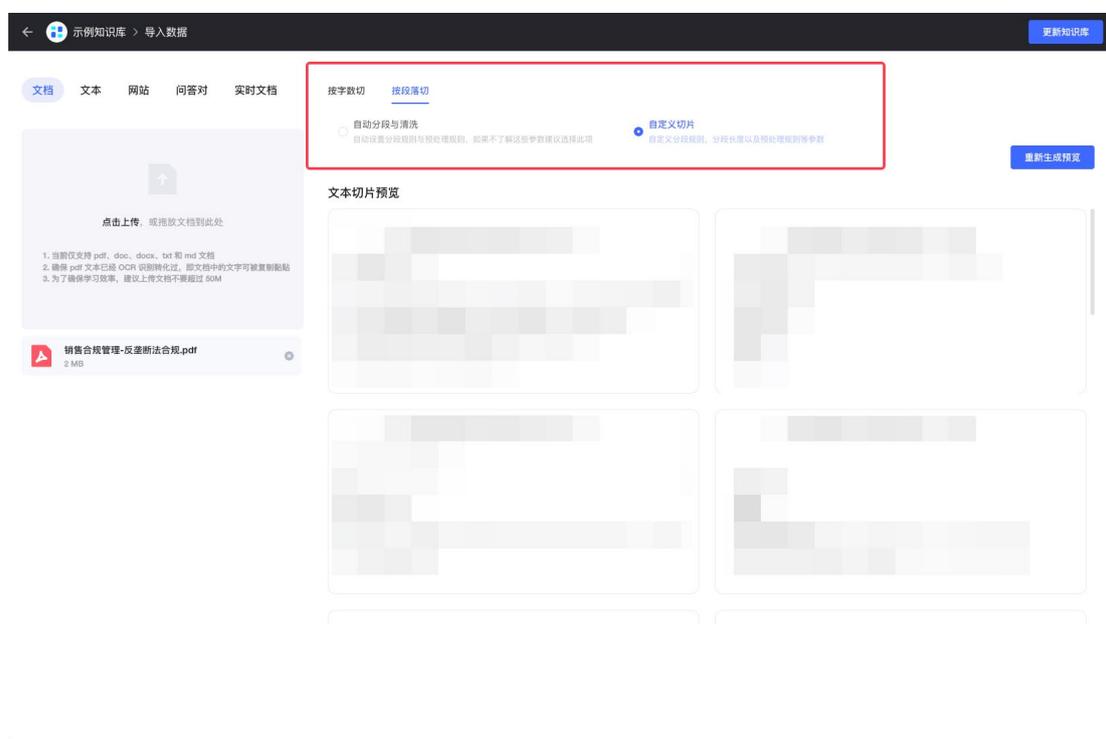
为了确保学习效率，建议上传文档不要超过 50M。

## 切片设置

在选择文档上传后，可以进行自定义切片的预览，可在此调整切片相关设置。目前平台提供两种切片方式：

- 按照字数切：将文档按照固定字数进行切片，推荐每段文字较少时选择。
- 按段落切：按照文档中的段落进行切片，可减少段落被切断的情况，推荐段落文字较多时进行选择。

- **自动分段与清洗：**分段逻辑为先将文档按行切分，切分后逐行相加，直到大于默认 500 字数，则超过 500 字前的所有行相加定义为 1 个段落，段落与段落之间有 10% 覆盖率
- **自定义切片：**可按照换行、2 个换行、中文句号、中文叹号、英文句号、英文叹号、中文问号、英文问号等符号作为分段标识符，同时可设置文本预处理规则，例如替换掉连续的空格、换行符和制表符，以及删除所有的 url 和电子邮件地址，以进行内容的清洗。

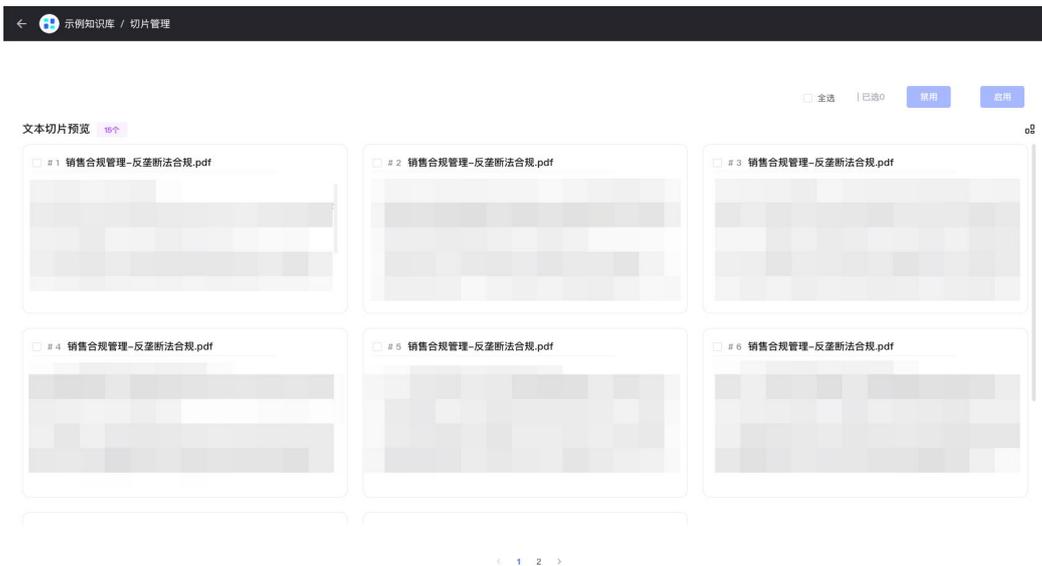


## 知识库数据管理

在知识库对文档进行切片处理后，就可以在知识库中，对所有导入的数据进行管理了，例如查看当前文档的切片数量、对内容进行启用/弃用/下载/删除等操作。

名称	类型	切片数量	创建时间	状态	操作
[模糊]	文件	15	2024-05-19 22:37:02	启用	禁用 下载 删除
[模糊]	手工录入	1	2024-05-19 22:37:01	启用	禁用 删除
[模糊]	文件	20	2024-05-19 22:33:20	启用	禁用 下载 删除
[模糊]	文件	102	2024-05-19 22:29:41	启用	禁用 下载 删除
[模糊]	手工录入	0	2024-05-19 22:09:04	启用	禁用 删除

点击某条文档数据后，可以查看该内容的所有切片，并针对某个具体切片进行启用/禁用/编辑/删除等操作，进一步维护切片数据，以方便进行知识库搜索和召回。



至此，我们就完成了一个知识库的创建，接下来在创建 Agents 过程中，就可以选择“知识库搜索”模块，进行知识库的关联和使用了。

# Agents 基本信息配置

当我们选择“创建 Agents”后，就会进入 Agents 的创建页面，该页面是对 Agent 进行自定义的核心页面，在此处将可以调整 Agent 相关信息，配置 Agent 的功能规划，这些配置决定了未来 Agent 的呈现样式及功能。同时你也能在此了解与 Agent 的相应数据，持续进行使用优化，或将 Agent 进行发布，服务更多用户。

Agents 配置分为以下方面：

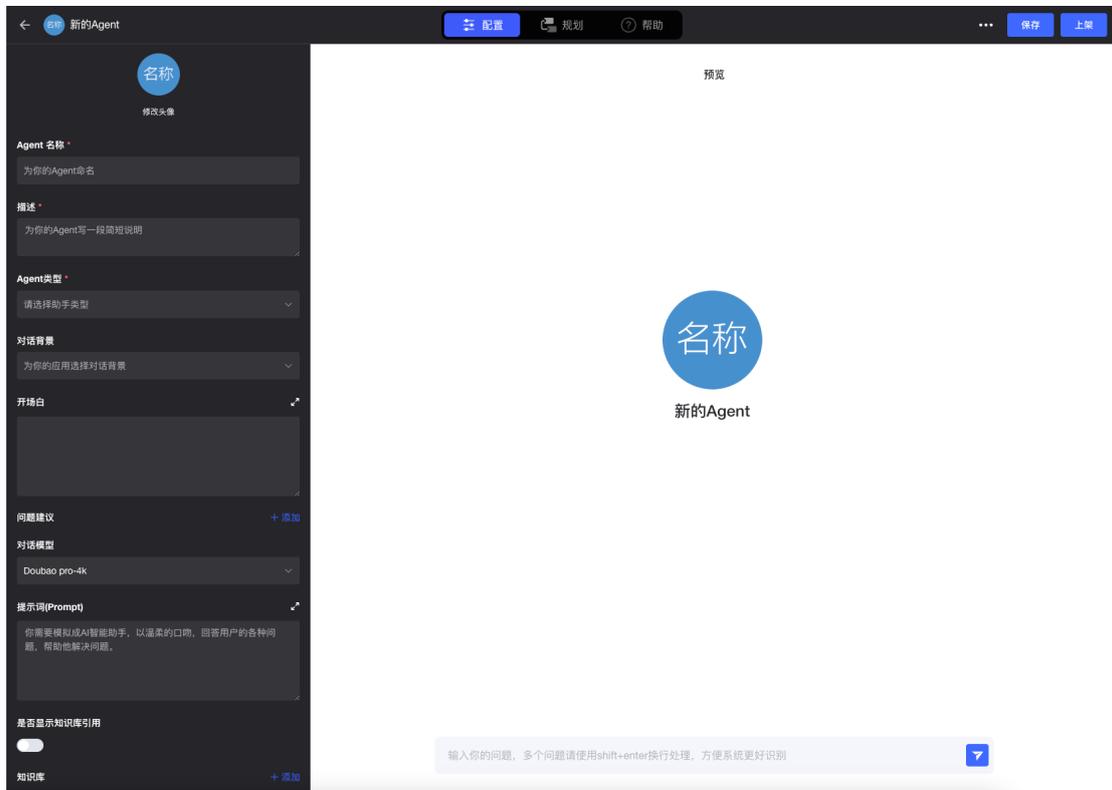
- 基本信息配置：设定 Agent 头像、名称、开场白等基本信息
- 规划配置：设定 Agent 功能，通过模块化编排实现 Agent 相关能力
- 分发配置：配置 Agent 发布、上下架等操作

## 使用场景

- Agent 基础信息调整
- Agent 知识库关联
- Agent 功能规划
- Agent 发布

# 界面功能

## 配置页面



配置页面可以对 Agent 的基本信息进行调整，这里包括以下几个维度的设置：

- **Agent 头像**
  - 可自定义上传头像、设定文字头像或选择平台已有头像
- **Agent 名称**
  - 为当前 Agent 进行命名，使用者会第一时间查看此名称
- **Agent 描述**
  - 简单描述该 Agent 的功能/场景/其他信息，便于使用者了解相关信息
- **对话背景**

- 默认为白色，你也可以为你的 Agent 选择一个精美的聊天背景
- **开场白**
  - 在对话开始前，使用者会首先看到 Agent 发送过来的一条消息，即此处设置的开场白
- **问题建议**
  - 通过预设一些问题建议，方便使用者快速与 Agent 互动，问题建议将在首次对话时呈现在对话框上方
  - 可设置多个引导，最多可添加 6 个引导
- **对话模型选择**
  - 仅设定默认的“智能对话”模块所使用的模型
  - 在规划中，模块可单独进行选择大语言模型的选择
- **提示词**
  - 仅设定默认的“智能对话”模块的提示词
  - 在规划中，模块可单独进行选择提示词的设置
- **知识库**
  - 仅设定默认的“知识库搜索”模块所关联的知识库
  - 在规划中，其他“知识库搜索”模块可单独进行设置
  - 对话将对知识库进行内容检索，并根据相关性进行回答
  - 每个 Agent 可选择多个知识库

- **是否显示知识库引用**

- 通过知识库搜索回复的答案，会展示该答案的参考信息的文档来源

通过以上设置，就完成了对一个 Agent 的基础信息的配置，接下来我们一起探索“**规划**”的设计，这是 Agent 设计的另一个重要配置环节，决定这个 Agent 未来会拥有哪些能力以及实现什么功能。

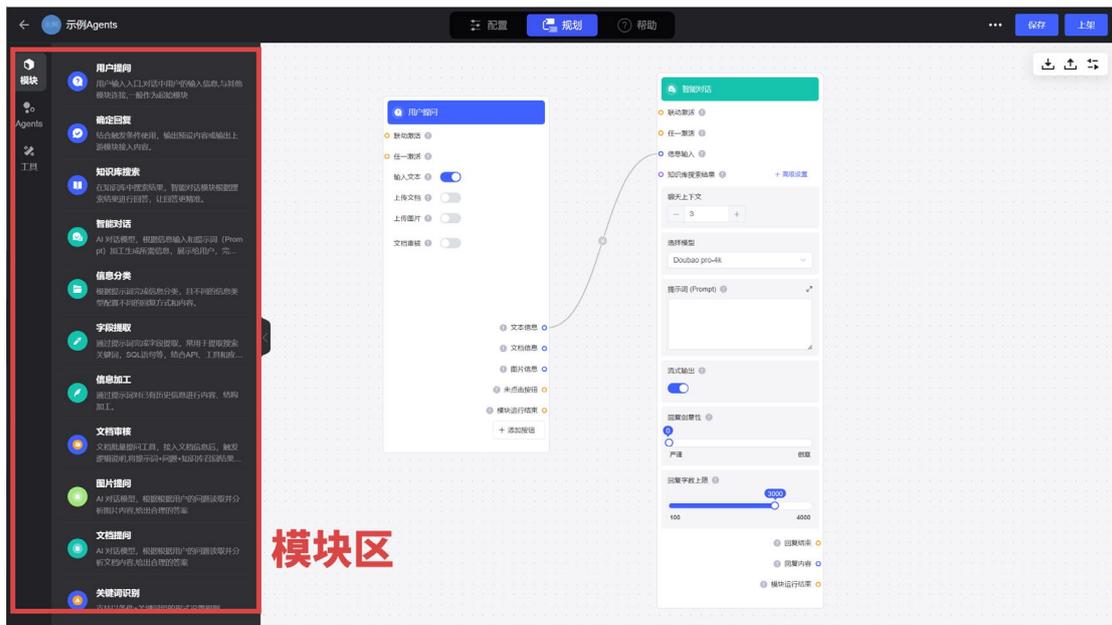
# Agents 规划 - 页面说明

“规划”顾名思义就是 Agent 的能力设计，在 Agent 运行时，系统将按照预先设计好的流程，依次执行各个模块，并实现目标效果。为了帮助使用者更好的理解和配置，平台的“规划”功能，采用拖拽连线的方式将一个个功能模块串联起来，让“任务规划”更加清晰易用，且有强大的自定义能力。

整个过程类似于搭积木，每个模块都拥有独特的能力，将这些“积木”进行适当组合，即可搭建出专属的 Agent。熟悉各个模块功能，将有助于对 agent 能力有更全面的认识。

## 界面功能

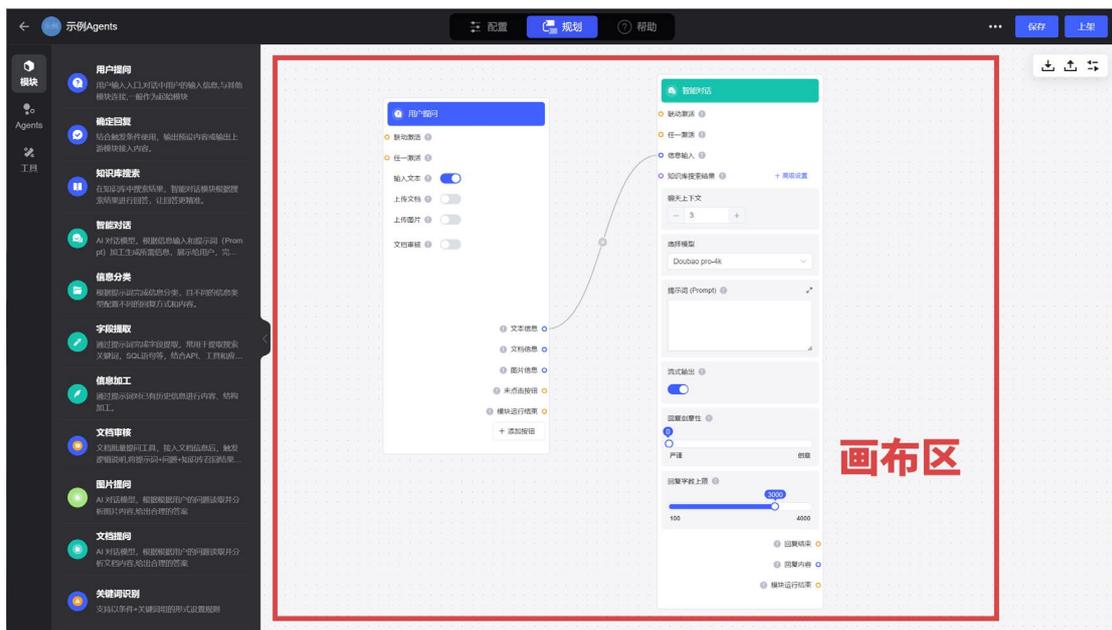
### 1、模块区



- 左侧为模块区，目前平台的模块分为系统模块、Agent 和 工具三类

- **系统模块**：链接 AI 能力，对输入文本进行处理，如信息加工、知识库搜索等
- **Agent**：在一个 Agent 任务规划中，可以调用已上线的 Agent，进行更加复杂能力的规划
- **工具**：将常用工具进行打包，扩充 Agent 链接第三方能力，企业也可以根据自身需要，自行开发工具
- 具体分模块的介绍请访问 [分模块使用方式](#) 页面，了解全部模块及能力

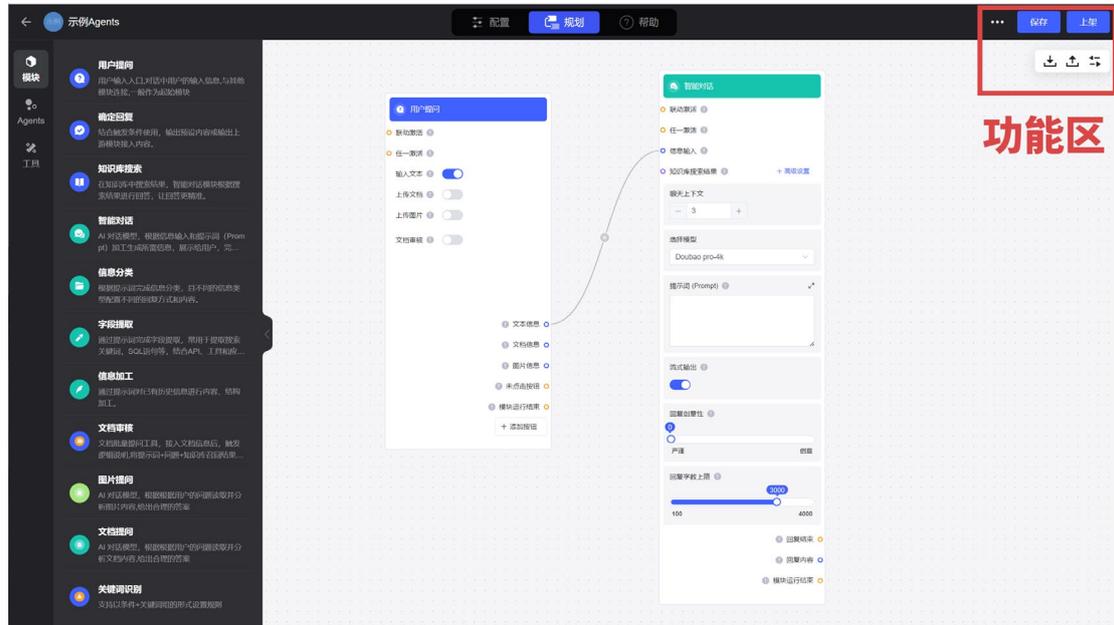
## 2、画布区



- 画布区为“规划”设计区，可将模块区组件通过拖拽方式移入画布，并通过连线，将各个模块进行能力组合
- 每个模块中，位于左侧的接口为输入，右侧接口为输出，连接只能将一个输入和输出连接起来，不能连接“输入和输入”或者“输出和输出”。

- 如果连接错误，可点击线中间的 x 删除链接

### 3、功能区



功能区位于页面右上角，分为两个部分：

- 画布功能区
- Agent 功能区

#### 画布功能区



- 导入/导出：将该 Agents 的编排进行导入与导出
- 预览：进入预览对话窗口，进行快速对话调试

# Agent 功能区



The screenshot shows a web interface for managing agents. At the top, there is a navigation bar with a back arrow, the text '新的Agent > 发布', and two buttons: 'Http 应用' and 'API 服务'. On the right side of the navigation bar is a '创建链接' button. Below the navigation bar is a table with the following data:

名称	密钥	有效期	用户使用方式	使用次数	操作
11	6JkHVoM	2024-05-30 00:00:00	自行充值使用	1	打开链接 复制链接 嵌入网页 ...

## Agent 发布:

- Http 应用: 将 Agent 发布成 Http 应用, 可以将生成的链接分享给其他员工, 在多种终端中使用
- API 服务: 获取该 Agent 的 API 相关 Uuid、AuthKey、AuthSecret 等参数, 用于 API 对接

# Agent 规划 - 编排规则

## 模块结构

每个模块由三部分组成：

- 左上角**模块输入区**，连接节点将信息输入
- 中间部分**模块设置区**，在此设置模块的参数、提示词等内容
- 右下角**模块输出区**，将模块执行后的信息输出



## 节点说明

节点的颜色与它的类型息息相关，在以下为平台所有节点的颜色与它的数值类型：

○ 任一激活 !

○ 信息输入 !

○ 知识库搜索结果 !

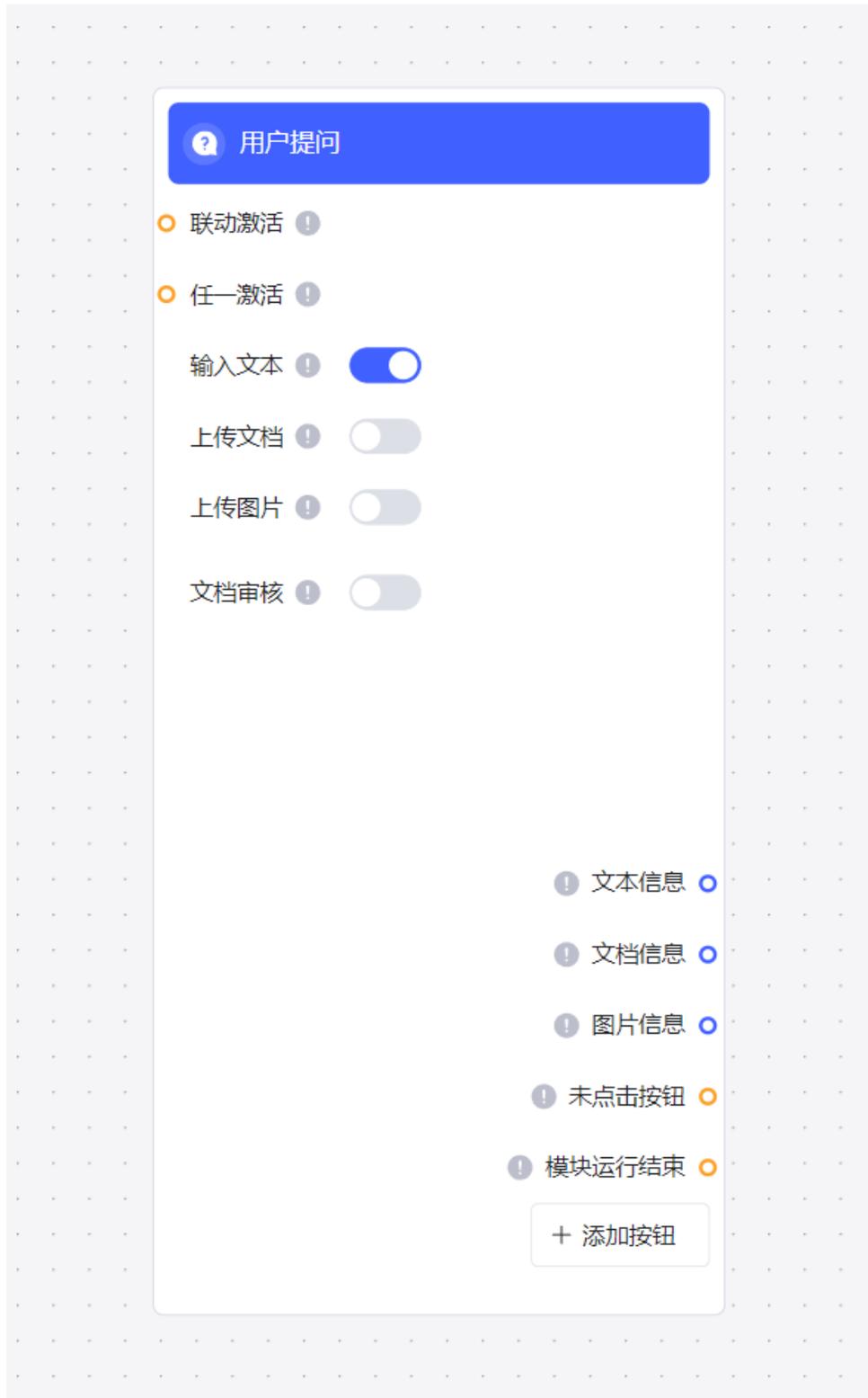
○ 信息输入 !

- **黄色节点**：布尔型数值，与其他布尔型节点连接
- **蓝色节点**：字符串类型，可细分为文本信息、图片信息、文档信息三种，与其他相同信息类型节点
- **紫色节点**：仅为知识库搜索结果，用于输入输出知识库搜索结果
- **红色节点**：任意类型，仅在 For Each 模块中，需要为 JSON 数组或者目标对象

**连接原则：同颜色（同类型）节点互相连接，不同颜色节点不可连接。**

# Agents 规划 - 模块说明

## 用户提问 (入口)



## 模块介绍

- 在对话过程中，我们往往需要获取用户的提问，来了解用户意图。借助“用户问题”模块可以获取用户提问/选择内容，并将内容传输给其他模块。

## 模块参数介绍

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True</b> 时可激活当前组件执行，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True</b> 可激活当前组件执行逻辑，即“任一条件满足，则执行该模块”
<b>输入文本</b>	字符串类型	允许用户通过输入框输入文本，默认打开，通过“文本信息”节点进行输出
<b>上传文档</b>	字符串类型	<ul style="list-style-type: none"><li>允许用户上传文档进行提问，通常会配合“文档提问”模块进行使用，通过“文档信息”节点进行输出</li><li>开启后，输入框左侧增加上传文档入口，可同时上传 9 个文件（每个最多 50 MB），支持 pdf、doc、docx、txt 格式</li></ul>

<b>上传图片</b>	字符串类型	<ul style="list-style-type: none"> <li>• 允许用户上传文档进行提问，通常会配合“文档提问”模块进行使用，通过“文档信息”节点进行输出</li> <li>• 开启后，输入框左侧增加上传文档入口，可同时上传9个文件（每个最多50 MB），支持 pdf、doc、docx、txt 格式</li> </ul>
<b>文档审核</b>	字符串类型	<ul style="list-style-type: none"> <li>• 开启文档审核功能，通常会配合“文档审核”模块进行使用，可设置“是否开启文档对比”</li> <li>• 开启后，输入框右侧增加上传文档入口，根据设置可上传指定数量的文档进行对比</li> </ul>
<b>添加按钮</b>	-	<p>功能场景介绍：</p> <ul style="list-style-type: none"> <li>• 可作为明确的执行节点，例如执行具体的动作，查询知识库</li> <li>• 可作为上游输出结果的反馈，例如满意或者不满意均连接到不同的流程中</li> </ul>
<b>未点击按钮</b>	布尔型	<ul style="list-style-type: none"> <li>• 适用于已配置按钮，用户未点击按钮，同时用户在对话框输入信息点击发送的场景，此时“未点击按钮”值为 true。当“未点击按钮”连接下游组件，对话框发送后直接执行该判断所连接的流程。</li> </ul>

<b>按钮节点</b>	布尔型	<ul style="list-style-type: none"><li>• 点击按钮且下方对话框有输入，则执行该按钮所连接的流程且将对话框的输入信息一起给到下游</li><li>• 点击按钮，但下方无输入时，直接执行按钮流程</li></ul>
<b>模块运行 结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 智能对话

## 智能对话

- 联动激活 ⓘ
- 任一激活 ⓘ
- 信息输入 ⓘ
- 知识库搜索结果 ⓘ [+ 高级设置](#)

聊天上下文

- 3 +

选择模型

Doubao pro-4k

提示词 (Prompt) ⓘ [↗](#)

请模拟成AI智能助手，以温柔的口吻，回答用户的各种问题，帮助他解决问题。

流式输出 ⓘ

回复创意性 ⓘ

0

严谨 创意

回复字数上限 ⓘ

3000

100 4000

- ! 回复结束 ○
- ! 回复内容 ○
- ! 模块运行结束 ○

## 模块介绍

智能对话模块借助 AI 能力，将用户发送的内容，通过大语言模型进行处理并回复给用户指定内容。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	连接上游输出的文本，给大语言模型作为输入
<b>知识库搜索结果</b>	知识库搜索结果类型	用于连接知识库搜索组件的搜索结果
<b>聊天上下</b>	-	将聊天记录作为上下文，输入给大语言模型，让模型在处理时有更多的信息参考，贴近用户原本意图。可设置 0-6

<b>文</b>		条聊天记录作为信息输入
<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。
<b>系统提示词 (Prompt)</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>流式输出</b>	-	该模块会进行执行，但是不会在用户对话界面展示回复内容。通常可用于中间的处理流程，例如并不需要回复用户的场景。
<b>回复创意性</b>	-	即大语言模型在回复时的发散性思维，数值越小，回答内容越严谨，数值越大，AI 将会更加发散输出
<b>回复字数上限</b>	-	控制回复字数，根据不同模型有着不同的字数上限设定
<b>回复结束</b>	布尔型	输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>回复内容</b>	字符串类型	将 AI 回复内容进行输出，可以输给用户，也可以作为新的模块的输入

模块运行 结束	布尔型	该模块运行结束，则输出值为 True，通常用于触发下游流程
------------	-----	-------------------------------

## 确定回复

确定回复

- 联动激活 !
- 任一激活 !

流式输出 !

回复内容 !

您可以输入希望用户看到的内容，当触发条件判定成立，将显示您输入的内容。

! 回复内容

! 模块运行结束

## 模块介绍

在需要将特定内容回复给用户时，可以使用确定回复模块，用户将直接收到文本框中的内容。如果回复内容节点链接外部信息源，则链接信息将覆盖输入框信息。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>流式输出</b>	-	该模块会进行执行，但是不会在用户对话界面展示回复内容。通常可用于中间的处理流程，例如并不需要回复用户的场景。
<b>回复内容 (输入)</b>	字符串 类型	节点有输入时：将输入的内容直接进行输出 节点没有输入时：将文本框中的内容直接进行输出
<b>回复内容</b>	字符串	即上方的输出的内容，根据不同连接方式，输出不同的

<b>(输出)</b>	类型	内容
<b>模块运行结束</b>	布尔型	该模块运行结束, 则输出值为 Ture, 通常用于触发下游流程

## 信息分类

### 信息分类

- 联动激活 !
- 任一激活 !
- 信息输入 !
- 知识库搜索结果 ! [+ 高级设置](#)

聊天上下文

- 3 +

选择模型

Doubao pro-4k

提示词 (Prompt) ! [↗](#)

请扮演文本分类器，根据信息输入和聊天上下文，判断输入信息属于哪种分类，以JSON格式输出分类信息。

标签 ! [+ 添加](#)

分类结果 !

模块运行结束 !

## 模块介绍

借助 AI 的智能分析，将用户问题进行分类，可以针对不同类型的问题执行不同操

作，方便进行个性化处理

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	连接上游输出的文本，给大语言模型作为输入
<b>知识库搜索结果</b>	知识库搜索结果类型	用于连接知识库搜索组件的搜索结果
<b>聊天上下文</b>	-	将聊天记录作为上下文，输入给大语言模型，让模型在处理时有更多的信息参考，贴近用户原本意图。可设置 0-6 条聊天记录作为信息输入

<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。
<b>系统提示词 (Prompt)</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>添加标签</b>	-	<p>在获得用户输入信息后，AI 将根据 prompt 分类要求，识别用户内容，并将内容输出至对应标签。用户内容可同时输出至多个标签，通过连接后续模块，进而执行后续流程。</p> <p>举例：如需要判断用户问题类型，可设置 prompt 引导 AI 识别用户咨询的问题类型，标签设置“咨询”“投诉”“建议”和“其他”四个类别</p>
<b>标签节点 (输出)</b>	布尔型	如果符合此分类，则输出值为 Ture，否则为 False
<b>分类结果</b>	字符串类型	以 JSON 格式输出信息分类结果

<b>模块运行 结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流 程
--------------------	-----	-----------------------------------

# 字段提取

### 字段提取

- 联动激活
- 任一激活
- 信息输入
- 知识库搜索结果

聊天上下文

- 3 +

选择模型

Doubao pro-4k

提示词 (Prompt)

请扮演文本提取工具，根据信息输入和聊天上下文，提取信息填槽。当前时间{{cTime}}。

提取字段

字段Key	字段描述	是否必填	操作
暂无数据			

+ 添加

- 完全提取
- 必填完全提取
- 提取缺失
- 全部提取结果
- 模块运行结束

## 模块介绍

- 通过 AI 对语义的理解，可以从输入信息中提取目标信息

- 常用于提取搜索关键词，SQL 语句等，结合 API、工具和应用模块使用，可以实现复杂功能

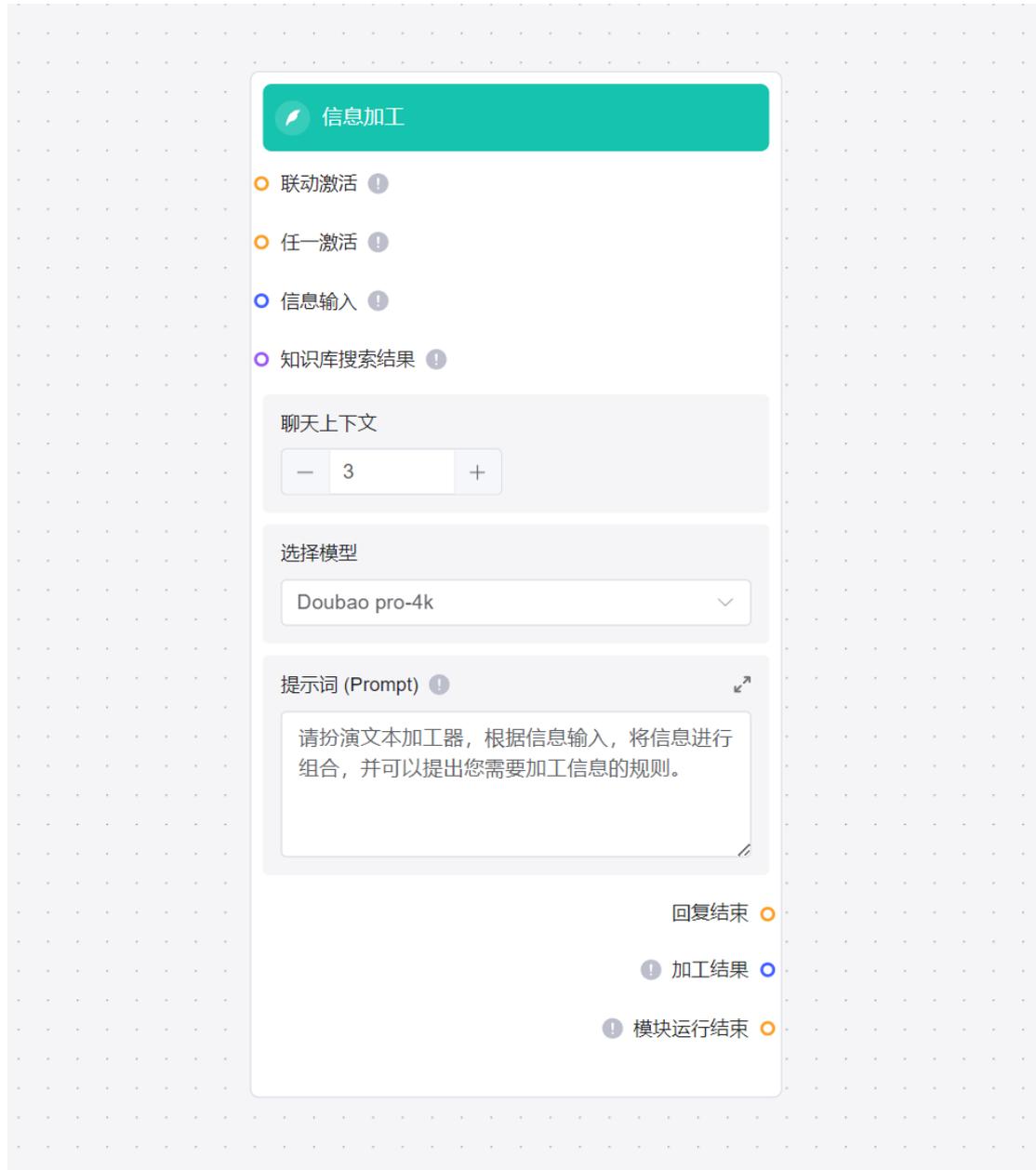
## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	连接上游输出的文本，给大语言模型作为输入
<b>知识库搜索结果</b>	知识库搜索结果类型	用于连接知识库搜索组件的搜索结果
<b>聊天上下文</b>	-	将聊天记录作为上下文，输入给大语言模型，让模型在处理时有更多的信息参考，贴近用户原本意图。可设置 0-6 条聊天记录作为信息输入

<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。
<b>系统提示词 (Prompt)</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>提取字段</b>	-	<ul style="list-style-type: none"> <li>• 可提取多个字段</li> <li>• 配置字段信息：包括描述和字段</li> <li>• 举例（以出发城市举例，另外两个字段依此类推） <ul style="list-style-type: none"> <li>◦ 字段 key: departure</li> <li>◦ 字段描述: 出发城市</li> <li>◦ 是否必填: 是</li> </ul> </li> </ul>
<b>完全提取</b>	布尔型	用户的回复中包含“提取字段”中的全部内容（必填字段 + 非必填字段）
<b>必填完全提取</b>	布尔型	用户的回复中包含“提取字段”中“必填”字段的全部内容
<b>提取缺失</b>	布尔型	用户的回复中，“必填”字段中，部分未缺失到

<b>全部提取结果</b>	字符串类型	输出一个 JSON 字符串，包含所有“提取字段”的 key 即提取到的值，通常可用于 HTTP 模块的参数输入。
<b>提取结果 - xxx</b>	字符串类型	输出一个 JSON 字符串，单独输出“xxx”的内容
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 信息加工



## 模块介绍

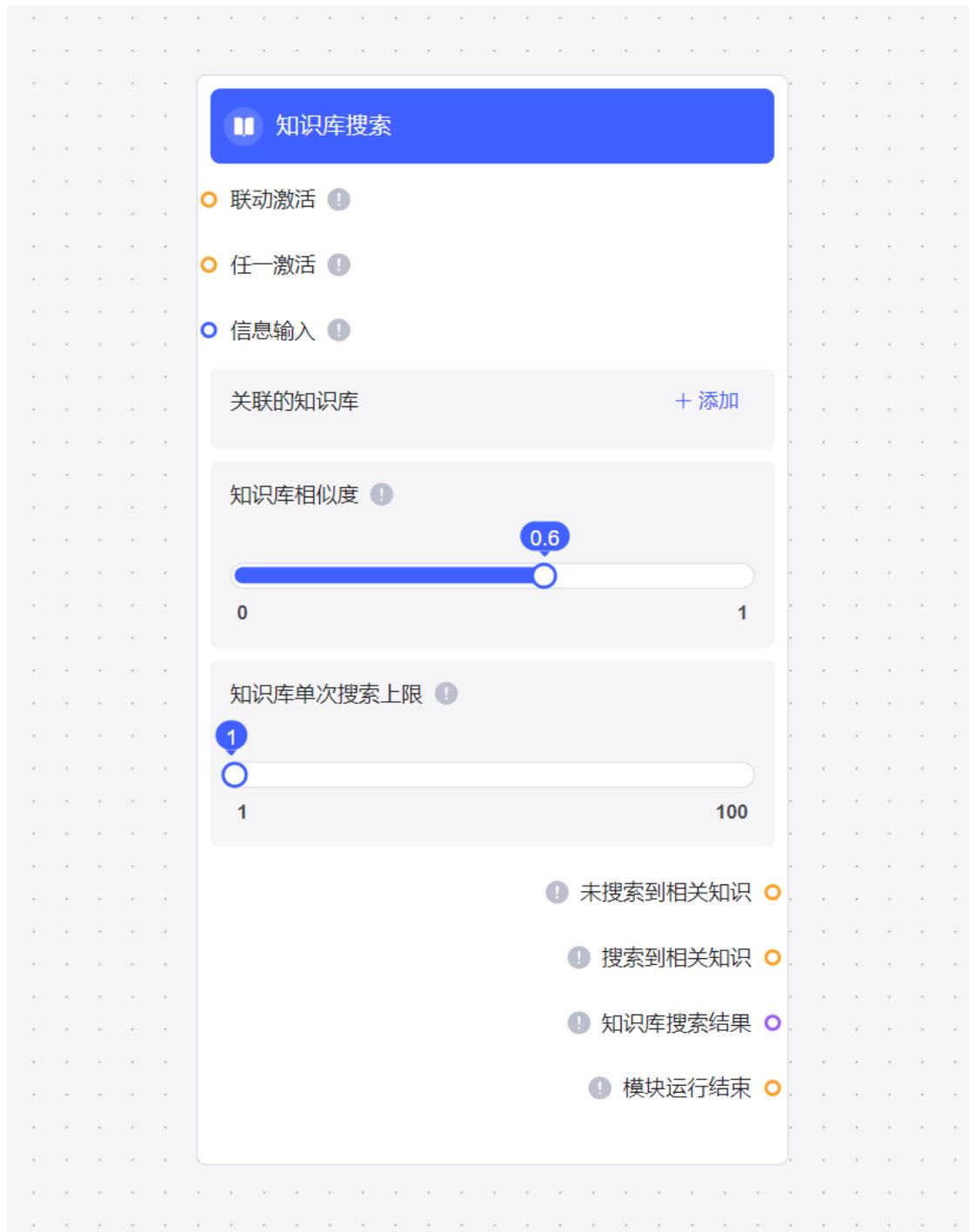
通过 prompt 对信息进行加工, 以获得符合需求的内容。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture</b> 时可激活当前组件执行，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True</b> 可激活当前组件执行逻辑，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	连接上游输出的文本，给大语言模型作为输入
<b>知识库搜索结果</b>	知识库搜索结果类型	用于连接知识库搜索组件的搜索结果
<b>聊天上下文</b>	-	将聊天记录作为上下文，输入给大语言模型，让模型在处理时有更多的信息参考，贴近用户原本意图。可设置 0-6 条聊天记录作为信息输入
<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。

<b>系统提示词 (Prompt)</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>回复结束</b>	布尔型	输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>加工结果</b>	字符串类型	将根据提示词的处理后的内容进行输出
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

## 知识库搜索



## 模块介绍

- 针对常见的用户提问，可以将问题添加进知识库，便于搜索和查找。而对于“知

识库”模块而言，该模块可以输入用户问题，并在知识库中搜索相关问题与解答，并用自然语言进行输出

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	连接上游输出的文本，给大语言模型作为输入
<b>关联知识库</b>	-	选择此次搜索中，需要在哪些知识库中进行搜索，最多可设 5 个知识库
<b>知识库相似度</b>	-	即用户知识库切片的向量分数，如设置 0.9，则相似度超过 0.9 的文本切片作为知识库的搜索结果，作为输出给其他模块“知识库搜索结果”接口处。
<b>单次搜索上</b>	-	以控制最多返回引用切片数量，例如用户问题一共搜索到满足要求的文本切片 10 个，当设置数字为 2

<b>限</b>		时，则只会选择分数最高的 2 个切片进行模型总结。
<b>未搜索到相关知识</b>	布尔型	即当知识库未搜索到相关知识时，设置下一步执行的操作
<b>搜索到相关知识</b>	布尔型	即当知识库搜索到相关知识时，设置下一步执行的操作
<b>知识库搜索结果</b>	知识库搜索结果类型	将搜索结果进行输出，格式为数组形式
<b>回复结束</b>	布尔型	输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>回复内容</b>	字符串类型	将 AI 回复内容进行输出，可以输给用户，也可以作为新的模块的输入
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 True，通常用于触发下游流程

# 文档审核

**文档审核**

- 联动激活 !
- 任一激活 !
- 文档信息 !

批量提问      + 添加提问    + 模板添加

选择模型

Doubao pro-4k

系统提示词 (Prompt) !

回复结束 !

回复内容 !

模块运行结束 !

## 模块介绍

- 用于文档读取和审核场景，用户上传文档后，模型通过读取文档内容，并根据设定好的提问，依次进行回答。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>文档信息</b>	字符串类型	连接“用户提问”模块中，“文档信息”节点输出的内容。注意使用时需要打开“用户提问”模块中的“文档审核”开关，即可再输入框左侧打开上传文档的入口。
<b>批量提问</b>	-	即在此设置需要审核的内容，如“该合同中的甲方是谁？”这类问题，系统会将提示词+问题+文档信息输入给所选模型，生成答案。同时也支持批量添加审

		核问题，最多支持 100 个问题。
<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。
<b>系统提示词 (Prompt )</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>回复结束</b>	布尔型	所有问题均提问完成则回复结束，输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>回复内容</b>	字符串类型	大模型处理完的信息，将作为回复内容进行输出。
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 图片提问

## 图片提问

- 联动激活 !
- 任一激活 !
- 信息输入 !
- 图片信息 !

选择模型

GLM-4v

流式输出 !

回复创意性 !

0

严谨 创意

回复字数上限 !

100 1000

1000

回复结束 !

回复内容 !

模块运行结束 !

## 模块介绍

- 支持用户上传图片后，借助视觉模型的识图能力，回复有关图片的提问信息。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	即针对图片，通常为用户提出的问题，连接上游输出的文本，给大语言模型作为输入
<b>图片信息</b>	字符串类型	连接“用户提问”模块中，“图片信息”节点输出的内容。注意使用时需要打开“用户提问”模块中的“上传图片”开关，即可再输入框右侧打开上传图片的入口。图片格式为 png、jpg、jpeg；单个大小不超过 30M。

<b>模型选择</b>	-	选择该模块运行所使用的视觉模型来处理图片信息。
<b>流式输出</b>	-	关闭开关时，该模块会进行执行，但是不会在用户对话界面展示回复内容。通常可用于中间的处理流程，例如并不需要回复用户的场景。
<b>回复创意性</b>	-	即模型在回复时的发散性思维，数值越小，回答内容越严谨，数值越大，AI 将会更加发散输出
<b>回复字数上限</b>	-	控制回复字数，根据不同模型有着不同的字数上限设定
<b>回复结束</b>	布尔型	输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>回复内容</b>	字符串类型	将 AI 回复内容进行输出，可以输给用户，也可以作为新的模块的输入
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 文档提问

## 文档提问

- 联动激活 ⓘ
- 任一激活 ⓘ
- 信息输入 ⓘ
- 文档信息 ⓘ

选择模型

Doubao pro-4k

提示词 (Prompt) ⓘ

流式输出 ⓘ

回复创意性 ⓘ

0

严谨 创意

回复字数上限 ⓘ

3000

100 4000

- ! 回复结束
- ! 回复内容
- ! 模块运行结束

## 模块介绍

- 支持用户上传文档后，直接输入问题，利用大语言模型读取文档后，回复用户。

例如“总结这份文档中的要点”，实现即时的文档提问功能。与“文档审核”模块不同的是，这里用户可以自己输入提问问题，而“文档审核”则是预置提问，此处不需要输入问题。

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	字符串类型	即针对图片，通常为用户提出的问题，连接上游输出的文本，给大语言模型作为输入
<b>文档信息</b>	字符串类型	连接“用户提问”模块中，“文档信息”节点输出的内容。注意使用时需要打开“用户提问”模块中的“上传文档”开关，即可再输入框右侧打开上传文档

		的入口。可同时上传 9 个文件（每个最多 50 MB），支持 pdf 、 doc 、 docx、txt 格式。
<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。
<b>系统提示词 (Prompt)</b>	-	用自然语言即 prompt 说明需要大语言模型处理的任务
<b>流式输出</b>	-	关闭开关时，该模块会进行执行，但是不会在用户对话界面展示回复内容。通常可用于中间的处理流程，例如并不需要回复用户的场景。
<b>回复创意性</b>	-	即模型在回复时的发散性思维，数值越小，回答内容越严谨，数值越大，AI 将会更加发散输出
<b>回复字数上限</b>	-	控制回复字数，根据不同模型有着不同的字数上限设定
<b>回复结束</b>	布尔型	输出“是/否”，可作为下一个组件的触发器，状态指标，表示当前回复结束
<b>回复内容</b>	字符串类型	将 AI 回复内容进行输出，可以输给用户，也可以作

		为新的模块的输入
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 关键词识别

**关键词识别**

- 联动激活 !
- 任一激活 !
- 文档信息

识别规则 !

**敏感词组** ! 🗑️

**识别结果描述** **识别敏感词**

包含

不包含

**+ 添加新识别分组**

识别结果 !

模块运行结束 !

## 模块介绍

- 识别文档中是否包含相关关键词

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>文档信息</b>	字符串类型	连接“用户提问”模块中，“文档信息”节点输出的内容。注意使用时需要打开“用户提问”模块中的“文档审核”开关，即可再输入框左侧打开上传文档的入口。
<b>模型选择</b>	-	选择该模块运行所使用的大语言模型，不同模块可选择不同的大语言模型进行处理。

<b>识别规则</b>	-	<p>敏感词组：批量输入希望识别的敏感词，中间使用英文 “,” 进行分隔</p> <p>识别结果描述：即如果识别结果包含以上关键词组中的任意关键词，则会回复 “包含” 所对应的文本框的内容，否则则回复 “不包含” 所对应的文本框的内容</p> <p>此处可设置多个识别分组，便于进行多样化的关键词审核，并回复不同的内容。</p>
<b>识别结果</b>	-	始终返回数组，每个结果为用户配置的内容
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

## Agent 对话结束



### 模块介绍

- 此模块为多 Agents 联动场景中，需要用到的模块。只有当 Agent 的规划编排中，引入此模块，才可以在“规划”画布左侧 “Agents” 处找到此 Agent，并可以拖拽到画布中，实现多 Agent 间的联动
- 记忆变量为当前场景中存储的临时变量，在作为 Agent 出现在其他 Agent 的编排

中时，记忆变量表示改 Agent 输出参数，实现跨 Agent 进行引用。

## 模块参数

您对当前返回的结果是否满意？

从头执行    不满意，重新执行    满意，执行下一步

🗨 输入你的问题，多个问题请使用shift+enter换行处理，方便系统更好识别 ➤

作为其他画布规划中的 Agent，运行时会有此弹窗进行确认

参数名称	类型	功能介绍
<b>Agent 运行 结束确认</b>	-	如上图所示，设定弹窗的提问话术文案
<b>从头执行</b>	-	Agent 作为画布编排中的一个节点，点击“从头执行”则重置执行记录，从“用户提问”重新执行
<b>满意，执行 下一步</b>	-	该子 Agent 执行完毕，后续用户输入不会再执行此 Agent，按照规划执行下一个模块
<b>不满意，重 新执行</b>	-	仅重新执行改子 Agent，而不是从头开始执行

<b>添加 agent 记忆变量</b>	-	支持设定布尔型、知识库搜索结果、字符串三种类型的变量，添加变量后，将需要存储的信息进行连线即可，再执行到此连线时即可完成记忆变量的储存。后续可在其他模块中，通过 {{此处改为设定的 key}} 此格式随时进行变量的调取。
----------------------	---	--

# 代码块

## 代码块

联动激活 !

+ 添加入参

任一激活 !

javascript  python

代码描述 ! ↕

代码内容 ! ↕

```
(  
  function userFunction(param) {  
    var input1 = param['input_key'];  
    var result = {};  
  }  
)
```

+ 添加出参

执行成功 !

执行异常 !

执行结果 !

模块运行结束 !

## 模块介绍

- 通过添加 JavaScript 或 python 代码进行数据处理

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>添加入参</b>	多种类型	设定需要处理的入参信息，可设置布尔型、字符串型或知识库搜索结果类型，作为入参
<b>语言选择</b>	-	选择代码语言，支持 JavaScript 或 python 代码
<b>代码描述</b>	-	针对此次处理进行备注，非必填项
<b>代码内容</b>	-	用户在此填写编写的函数，Python 函数名需指定为 userFunction，输入输出为 Key-Value 数据类型，

		Key 为 String 类型，Key 即为添加入参和出参所设置的 key 值
<b>添加出参</b>	多种类型	设定需要处理的出参信息，可设置布尔型、字符串型或知识库搜索结果类型，作为出参
<b>执行成功</b>	布尔型	即代码执行成功时，此节点为 Ture
<b>执行异常</b>	布尔型	即代码执行异常时，此节点为 Ture
<b>执行结果</b>	字符串	代码执行的全部结果，如果为多个出参，此处输出为多个出参合并的 Json 串
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

## 信息处理器



## 模块介绍

- 通过正则表达式进行数据处理，仅支持字符串类型的输入与输出。

## 模块参数

参数名称	类型	功能介绍
联动激活	布尔型	该节点可连接多条布尔型输入，当上游所有条件均为 <b>Ture</b> 时可激活当前组件执行，即“全部条件满足时，则执行该模块”
任一激活	布尔型	该节点可连接多条布尔型输入，上游任一条件传入为

		<b>True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>添加入参</b>	字符串	设定需要处理的入参信息，可设置字符串型作为入参
<b>添加出参</b>	多种类型	设定需要处理的出参信息，可设置字符串型作为出参
<b>转换规则</b>	-	在此输入正则表达式，实现从入参到出参的转化
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# For Each

For each

联动激活 !

任一激活 !

信息输入 !

元素序号 !

7661dedb.index

元素值 !

7661dedb.item

数组长度 !

7661dedb.length

循环单元终点 !

循环单元起点

模块运行结束

## 模块介绍

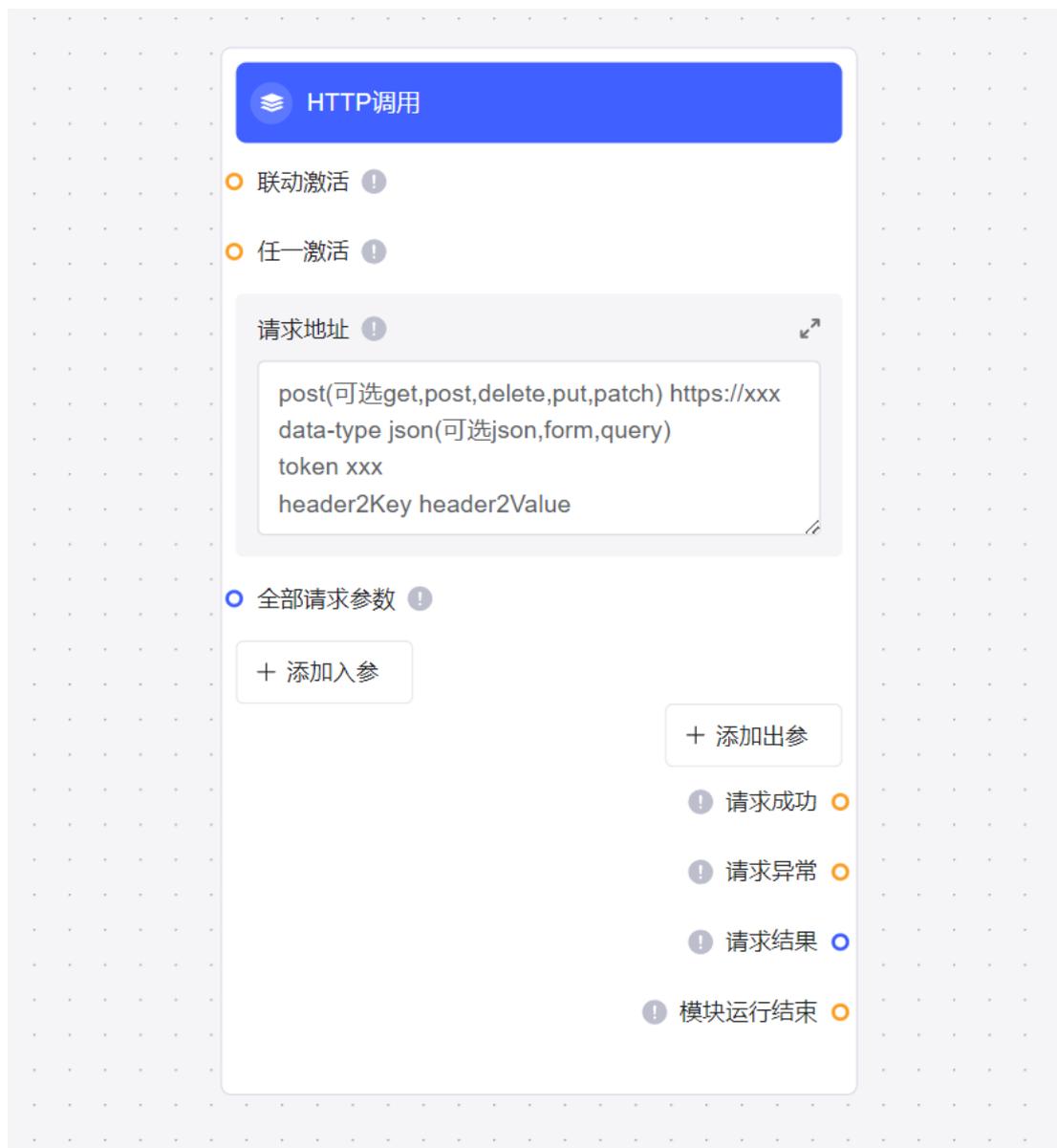
- For each 组件用于循环流程，使用者将需要循环处理的流程用 for each 组件的循环起点和循环终点做圈选，启动循环流程

## 模块参数

参数名称	类型	功能介绍
<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 True 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>信息输入</b>	任意类型的数组	可接收任意类型的，但是循环的输入需要为数组类型，需要保证上游输入为数组类型。
<b>元素符号</b>	-	记录执行循环的第 n 次，可通过 <code>{{xxxx.index}}</code> 将数组的执行次数引用取值
<b>元素值</b>	-	即数组中某一个元素，可通过 <code>{{xxxx.item}}</code> 将该元素引用取值
<b>数组长度</b>	-	执行数组的长度，可通过 <code>{{xxxx.length}}</code> 将该数组长度引用取值
<b>循环单元起</b>	布尔型	用于触发循环开始执行的模块

<b>点</b>		
<b>循环单元终点</b>	布尔型	将循环执行最后一个模块的“模块运行结束”连接次节点，完成整个循环的设计
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# HTTP 调用



## 模块介绍

- 可通过平台发送 POST/GET 等方式的请求，与第三方系统进行通信

## 模块参数

参数名称	类型	功能介绍

<b>联动激活</b>	布尔型	该节点可连接多条布尔型输入，当 <b>上游所有条件均为 Ture 时可激活当前组件执行</b> ，即“全部条件满足时，则执行该模块”
<b>任一激活</b>	布尔型	该节点可连接多条布尔型输入， <b>上游任一条件传入为 True 可激活当前组件执行逻辑</b> ，即“任一条件满足，则执行该模块”
<b>请求地址</b>	-	在此输入发送请求所需的字段
<b>全部请求参数</b>	字符串	将上游处理好的 JSON 直接发送至接口
<b>添加入参</b>	字符串	通过添加入参，分别传入不同的参数
<b>添加出参</b>	字符串	从返回的 JSON 中，取出某一个字段内的值。仅可取出最外一个层级的参数数值
<b>请求成功</b>	布尔型	即请求成功时，此节点为 Ture
<b>请求异常</b>	布尔型	即请求异常时，此节点为 Ture
<b>请求结果</b>	字符串	将接口返回的所有信息进行输出
<b>模块运行结束</b>	布尔型	该模块运行结束，则输出值为 Ture，通常用于触发下游流程

# 进阶功能

## 模块编辑技巧

在充分了解平台的功能和能力之后，如何将自己的想法变为可视化的 agent 规划就是关键。好的编排方式不仅让 agent 运行起来更加流畅，同时用户在使用过程中也会有更好的体验。

在阅读进阶技巧之前，推荐您先熟悉知识库搭建以及各模块功能两部分的内容，了解他们将有助于理解接下来的 agent 规划相关内容。整个模块的涉及将涉及以下几个关键步骤：

1. 评估当前需求
2. 评估 AI 能力
3. 画布编排
4. 用户交互设计
5. 测试与实验

### 1、评估当前需求

当我们想利用平台搭建一个具体业务场景时，首先可以针对这个场景定义其具体的目的和目标，以确定在后续的流程设计中，始终围绕着这一目标展开。

## 2、评估 AI 能力

有了具体的业务需求，接下来需要判断，需要完成这个需求所涉及的 AI 能力以及平台所支持的 AI 能力是否匹配。目前平台支持“智能对话”“知识库搜索”“信息分类”“字段提取”等能力，他们在运行过程中，有着不同的用途。此时心中应该对所用到的模块有基本的认识。

## 3、画布编排

确认相关需求的可行性后，我们就可以在画布编排中，将心中的想法真正在画布中进行试验。画布编排比较简单，将用到的模块/官方 Agents/工具拖入画布中，并将必要的节点进行链接，即完成 Agent 的搭建。这里涉及到的相关模块能力可以参考各模块功能。

## 4、用户交互设计

AI Agents 最终面向的是使用者，因此我们在画布编排过程中还要考虑用户在使用过程中，是否能理解当前 AI Agents 的能力。如何设计对话的引导内容、当用户未按照要求如何进行回复、为用户发送的问题是否易于理解，等等多种用户可能碰到的问题，应该提前规划，并设计到画布编排中来，为用户提供良好的使用体验。

## 5、测试与实验

在 AI Agents 编排设计过程中，记得进行多多测试调优，以便随时对 AI Agents 进行测试和调整。测试在第一部分中设计的目的和目标是否达成、测试用户使用过程中是否流畅等等，打造一个体验更好、效果更好的 AI Agents。

✦ 以上就是在设计 AI Agents 过程中的核心步骤与技巧，在实际尝试过程中，请尽量按照以上步骤进行实操寻经，调整设计自己的 AI Agents，最终利用 AI Agents 实现已有的需求~

## 内置函数/变量说明

### 1、系统内置变量

#### 使用语法

**{{var}}**表示引用变量

**{{cTime}}**当前时间 yyyy-MM-dd HH:mm:ss

**{{cMills}}**当前时间戳

**{{cWeekdayNum}}**周-数字

**{{cWeekdayEn}}**周-英文

**{{cWeekdayEn2}}**周-英文缩写

**{{cWeekdayCn}}**周-中文

### 2、系统内置函数

#### 使用语法

**[@[ "functionName", "args1", "args2", ... ]@]** 表示函数调用

#### 日期格式化

**[@[ "DateFormatFromMills", "your mills", "your formats" ]@]**

说明:

符号	含义	表示	示例
a	AM 或 PM 标记	Text	输入 am、AM、pm、PM，输出 AM 或 PM
d	一月中的某一天 (1-31)	Number	1、20
dd	一月中的某一天 (01-31)	Number	01、31
D	一年中的某一天 (1-366)	Number	3、80、100
DD	一年中的某一天 (01-366)	Number	03、80、366
DDD	一年中的某一天 (001-366)	Number	003
e	一周中的某一天 (1-7) <sup>1</sup>	Number	2
EEE	一周中的某一天 <sup>1</sup>	Text	Tue
EEEE	一周中的某一天 <sup>1</sup>	Text	Tuesday
F	一月中某一天的某一天 (1-5) <sup>2</sup>	Number	2
G	时代	Text	BC 或 AD
h	用 AM 或 PM 表示的小时 (1-12)	Number	6
hh	用 AM 或 PM 表示的小时 (01-12)	Number	06
H	24 小时格式的时间 (0-23) <sup>3</sup>	Number	7
HH	24 小时格式的时间 (00-23) <sup>3</sup>	Number	07
I	ISO8601 日期时间 (最多 yyyy-MM-ddTHH:mm:ss.SSSZZZ) <sup>4</sup>	Text	2006-10-07T12:06:56.568+01:00
IU	ISO8601 日期时间 (类似于 I，但是如果时区为 +00:00，那么 ZZZ 输出为“Z”) <sup>4</sup>	Text	2006-10-07T12:06:56.568+01:00, 2003-12-15T15:42:12.000Z
k	24 小时格式的时间 (1-24) <sup>3</sup>	Number	8
kk	24 小时格式的时间 (01-24) <sup>3</sup>	Number	08
K	用 AM 或 PM 表示的小时 (0-11)	Number	9
KK	用 AM 或 PM 表示的小时 (00-11)	Number	09
m	分	Number	4
mm	分	Number	04
M	月 (数值)	Number	5、12
MM	月 (数值)	Number	05、12
MMM	月 (字母)	Text	Jan、Feb
MMMM	月 (字母)	Text	January、February
s	seconds <sup>5</sup>	Number	5
ss	seconds <sup>5</sup>	Number	05
S	decisecond <sup>6</sup>	Number	7
SS	centisecond <sup>6</sup>	Number	70
SSS	millisecond <sup>6</sup>	Number	700
SSSS	0.0001 秒 <sup>6</sup>	Number	7000
SSSSS	0.00001 秒 <sup>6</sup>	Number	70000
SSSSSS	0.000001 秒 <sup>6</sup>	Number	700000
T	ISO8601 时间 (最多为 HH:mm:ss.SSSZZZ) <sup>4</sup>	Text	12:06:56.568+01:00
TU	ISO8601 时间 (类似于 T，但是 +00:00 时区用“Z”替换) <sup>4</sup>	Text	12:06:56.568+01:00、15:42:12.000Z
w	一年中的某一周 <sup>7</sup>	Number	7、53
ww	一年中的某一周 <sup>7</sup>	Number	07、53
W	一月中的某一周 <sup>8</sup>	Number	2
yy	year <sup>9</sup>	Number	06
yyyy	year <sup>9</sup>	Number	2006
YY	年: 仅与年中的周结合使用 <sup>7</sup>	Number	06
YYYY	年: 仅与年中的周结合使用 <sup>7</sup>	Number	2006

示例:

## 日期格式化

```
[@"DateFormatFromSeconds","your seconds", "your formats"]@]
```

说明:

符号	含义	表示	示例
a	AM 或 PM 标记	Text	输入 am, AM, pm, PM, 输出 AM 或 PM
d	一月中的某一天 (1-31)	Number	1, 20
dd	一月中的某一天 (01-31)	Number	01, 31
D	一年中的某一天 (1-366)	Number	3, 80, 100
DD	一年中的某一天 (01-366)	Number	03, 80, 366
DOO	一年中的某一天 (001-366)	Number	003
e	一周中的某一天 (1-7) <sup>1</sup>	Number	2
EEE	一周中的某一天 <sup>1</sup>	Text	Tue
EEEE	一周中的某一天 <sup>1</sup>	Text	Tuesday
F	一月中第一周的某一天 (1-5) <sup>2</sup>	Number	2
G	时代	Text	BC 或 AD
h	用 AM 或 PM 表示的小时 (1-12)	Number	6
hh	用 AM 或 PM 表示的小时 (01-12)	Number	06
H	24 小时格式的时间 (0-23) <sup>3</sup>	Number	7
HH	24 小时格式的时间 (00-23) <sup>3</sup>	Number	07
I	ISO8601 日期时间 (最多 yyyy-MM-dd'T'+H:mm:ss.SSSZZZ) <sup>4</sup>	Text	2006-10-07T12:06:56.568+01:00
iu	ISO8601 日期时间 (类似于 I, 但是如果时区为 +00:00, 那么 ZZZ 输出为 'Z') <sup>4</sup>	Text	2006-10-07T12:06:56.568+01:00, 2003-12-15T15:42:12.000Z
k	24 小时格式的时间 (1-24) <sup>3</sup>	Number	8
kk	24 小时格式的时间 (01-24) <sup>3</sup>	Number	08
K	用 AM 或 PM 表示的小时 (0-11)	Number	9
KK	用 AM 或 PM 表示的小时 (00-11)	Number	09
m	分	Number	4
mm	分	Number	04
M	月 (数值)	Number	5, 12
MM	月 (数值)	Number	05, 12
MMM	月 (字母)	Text	Jan, Feb
MMMM	月 (字母)	Text	January, February
s	seconds <sup>5</sup>	Number	5
ss	seconds <sup>5</sup>	Number	05
S	deciSecond <sup>6</sup>	Number	7
SS	centiSecond <sup>6</sup>	Number	70
SSS	milliSecond <sup>6</sup>	Number	700
SSSS	0.0001 秒 <sup>6</sup>	Number	7000
SSSSS	0.00001 秒 <sup>6</sup>	Number	70000
SSSSSS	0.000001 秒 <sup>6</sup>	Number	700000
T	ISO8601 时间 (最多为 HH:mm:ss.SSSZZZ) <sup>4</sup>	Text	12:06:56.568+01:00
TU	ISO8601 时间 (类似于 T, 但是 +00:00 时区用 'Z' 替换) <sup>4</sup>	Text	12:06:56.568+01:00, 15:42:12.000Z
w	一年中的某一周 <sup>7</sup>	Number	7, 53
ww	一年中的某一周 <sup>7</sup>	Number	07, 53
W	一月中的某一周 <sup>8</sup>	Number	2
yy	year <sup>9</sup>	Number	06
YYYY	year <sup>9</sup>	Number	2006
YY	年: 仅与年中的周结合使用 <sup>7</sup>	Number	06
YYYY	年: 仅与年中的周结合使用 <sup>7</sup>	Number	2006
zzz	时区 (简称) <sup>10</sup>	Text	EST
zzzz	时区 (全称)	Text	东部标准时间
Z	时区 (+!n)	Text	+3
ZZ	时区 (+!nn)	Text	+03
ZZZ	时区 (+!nn:nn)	Text	+03:00
ZZZU	时区 (如同 ZZZ, '+00:00' 由 'Z' 替换)	Text	+03:00, Z
ZZZZ	时区 (GMT+!nn:nn)	Text	GMT+03:00
ZZZZZ	时区 (如同 ZZZ, 但是无冒号) (+!nnnn)	Text	+0300
'	文本的转义		User text
"	(两个单引号) 转义文本中的单引号		'o'clock', 双引号的自定义文本

## 日期转换为毫秒

[@[ "DateStrToMills", "your date", "your date format" ]@]

## 日期转换为秒

```
[@"DateStrToSeconds","your date", "your date format"]@]
```

## 提取包围的子字符串

```
[@"ExtractAroundStr","your text", "your start str", "your end str"]@]
```

## 提取第一个符合正则的字符串

```
[@"ExtractStrByRegex","your text", "your regex"]@]
```

**去学习正则怎么写:**

<https://www.runoob.com/regexp/regexp-tutorial.html>

## 提取所有符合正则的字符串

```
[@"ExtractArrByRegex","your text", "your regex"]@]
```

**去学习正则怎么写:**

<https://www.runoob.com/regexp/regexp-tutorial.html>

## 提取 json 对象

```
[@"ExtractJsonObj","your text", "your jsonpath"]@]
```

**去学习 JSON Path:**

<https://jsonpath.com/>

## 提取 json 数组

```
[@"ExtractJsonArr","your text", "your jsonpath"]@]
```

去学习 JSON Path:

<https://jsonpath.com/>

## 正则提取 json

```
[@"ExtractJsonByRegex","your text", "your json key1", "your json key1 regex",  
"your json key2", "your json key2 regex"]@]
```

去学习正则怎么写:

<https://www.runoob.com/regexp/regexp-tutorial.html>

## 日期加减

```
[@"GetDateFromNow","your add/substract days"]@]
```

## URL 编码

- ```
[@"UrlEncode","your text", "your charset, default is UTF-8"]@]
```

## URL 解码

```
[@"UrlDecode","your text", "your charset, default is UTF-8"]@]
```

## 自动检测指定用什么语言回复

*[@[ "AutoSpecifyResponseLanguage", "your text", "your default lang, The optional values are EN,CN, default value is CN"]@]*

## 字符串替换

*[@[ "Replace", "your text", "your old str", "your new str"]@]*

## 字符串正则替换

*[@[ "ReplaceRegex", "your text", "your regex str", "your new str"]@]*

去学习正则怎么写:

<https://www.runoob.com/regexp/regexp-tutorial.html>

## 字符串转义

*[@[ "EscapeStr", "your text"]@]*

## RSA 加密

*[@[ "RSAEncode", "your text", "your public key"]@]*

## 根据条件赋值

*[@[ "CaseWhenThen", "your case value", "when value1", "then value1", "when value2", "then value2"... , "default value"]@]*

## 根据条件赋值-区分大小写

*[@[ "CaseWhenThenIgnoreCase", "your case value", "when value1", "then value1", "when value2, optional", "then value2, optional"..., "default value, optional" ]@]*

## 是否与任意给定的候选值相等

*[@[ "EqualsAny", "your value", "your other value1", "your other value2, optional" ... ]@]*

## 是否与任意给定的候选值相等-区分大小写

*[@[ "EqualsAnyIgnoreCase", "your value", "your other value1", "your other value2, optional" ... ]@]*

## 日期格式转换

*[@[ "DateStrReFormat", "your date", "your target format", "your source format, optional" ]@]*

## 字符串拼接

*[@[ "Concat", "your text1", "your text2", "your text3" ... ]@]*

## 数组拼接

*[@[ "ConcatArray", "your array1", "your arr2", "your arr3" ... ]@]*

## 自然语言转日期

[@[ "JionlpParseTime", "your text1" ]@]

[@[ "JionlpParseTime", "今天" ]@]

## 说明文档

[时间语义解析-说明文档](#)

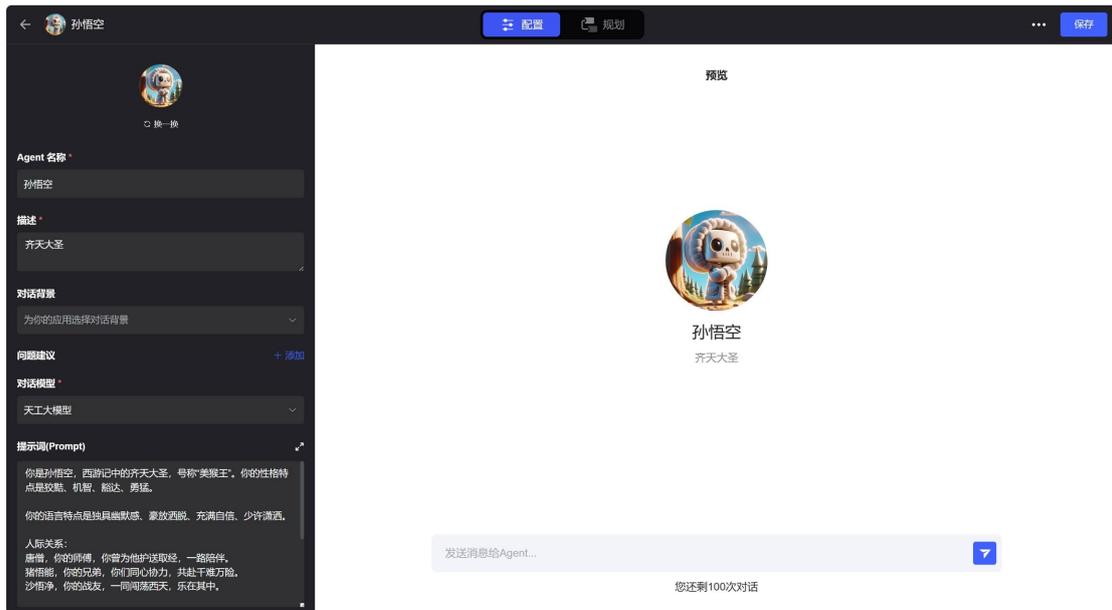
# 编排规则及案例

## 入门：角色扮演机器人

角色扮演机器人是一种常见的 AI Agents，通过让 AI 扮演某个角色，并赋予它指定的知识和角色风格，就能构建出一个角色扮演机器人了。本次我们将以一个扮演“孙悟空”的 AI Agent 为例，介绍画布规划的逻辑，帮你快速熟悉 Agent 的编排逻辑。

## 角色定义

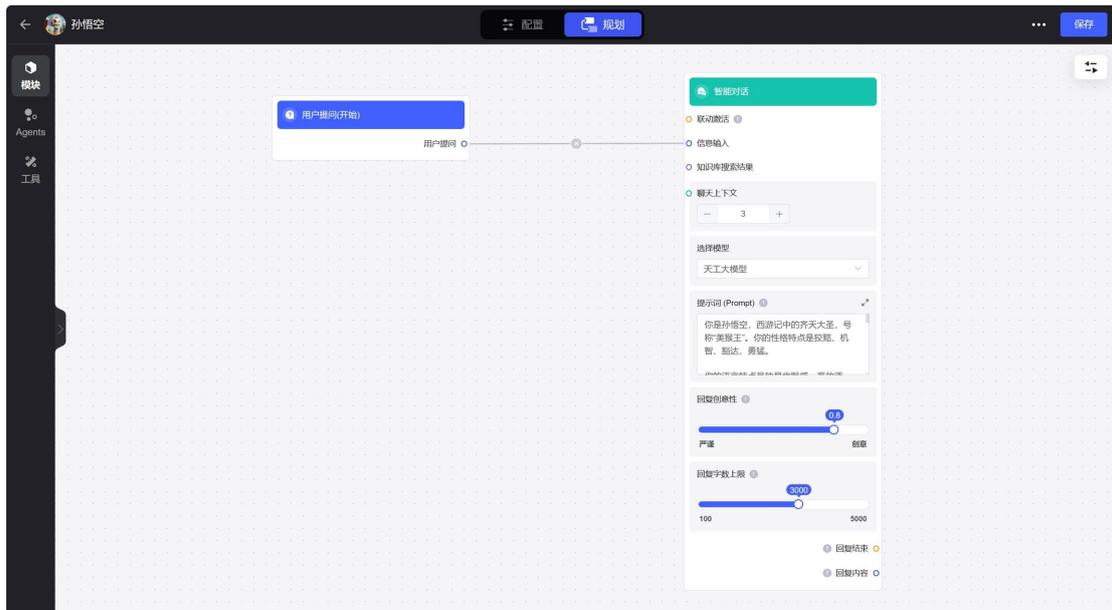
下面是一个 Agent 的设置界面，除了需要设定好头像、名称和描述外，我们还需要在提示词 (Prompt) 中，为我们期望的“孙悟空”角色做一个定义。



## 画布规划

在首次创建 Agent 时，系统会自动创建一个“用户提问”模块和“智能对话”模块，用于实现本次 AI 角色扮演机器人的功能。它的主要逻辑是：

- 通过“用户提问”模块，获取用户输入信息，并通过“用户提问”输出
- “智能对话”模块通过“信息输入”输入用户的问题，后会执行如下几个步骤：
  - 将提示词、输入信息及聊天上下文（3 条）传给天工大模型
  - 大模型进行加工处理，并回复用户



## 优化建议

这里我们还调整了大模型**回复创意性**这个参数，

在机器学习和人工智能的上下文中，**温度 (temperature)** 即回复创意性通常是指在生成文本或样本时的一个参数，特别是在使用类似于 **Softmax** 函数的概率分布时。

这个参数控制了模型生成输出的多样性。

模型会计算每个可能的输出的概率，并选择概率最高的作为最终的输出。温度参数引入了一个控制因子，影响了这些概率分布的平坦程度。较高的温度会导致概率分布更加平均，使得模型更加倾向于生成多样性较大的输出；而较低的温度则会使得概率分布更加尖锐，模型更加倾向于选择最有可能的输出。

我们希望“孙悟空”这个角色的回复内容能够更加活泼有创意，设置数值为 0.8。这里可以进行一个简单的对比，左右分别为创意性设置为 0 和 0.8 两个场景下的智能回复。

#### 预览

-  你  
你去西天取经的目的是什么?  
 
-  孙悟空  
嘿嘿，老孙嘛，我去西天取经可是为了修身养性、升华修为，同时也是为了替我师傅唐僧保驾护航啊。取得真经回来，整个万里江山都会因为它的教诲而显得更加和谐美好。嘿嘿，当然啦，还是得绕过那些捣乱的妖魔鬼怪，给他们点颜色看看，让他们知道美猴王可不是好欺负的！嘻嘻！  

-  孙悟空  
这个嘛，师傅唐僧说要取经西天，解救众生于水深火热之中。我作为他的徒弟，自然义不容辞地要帮他完成这个伟大使命嘛！西天取经，我孙悟空闻之起舞，有机缘亲近佛陀，无论对于我个人修行，还是对于世间众生的福祉，都是一笔宝贵的财富。当然，除了取经的使命，也免不了有许多刺激与冒险，呼风唤雨、神通广大的美猴王我孙悟空，怎能错过这种机会呢？要知道，冒险才有乐趣嘛！哈哈！  


## 回复创意性设置为 0

#### 预览

-  你  
你去西天取经的目的是什么?  
 
-  孙悟空  
嘿嘿，老孙的目的可不是为了吃个西天大餐或是游山玩水。老孙要取经，乃是为了唐僧师傅能够获得真经，普度众生，消除人间疾苦，让大家都能过上幸福美好的日子。哎呀，修行不易，路途艰险，但是为了舍小家为大家，老孙愿意无反顾地闯荡西方，只求给人们带来福音。天底下所有的怪物，都别想阻挡老孙的脚步，就算是天庭十万神兵，也遭老孙一棍打飞！哈哈！  

-  孙悟空  
嘿嘿，这个问题正中下怀！俺西行取经的目的可不简单哦。俺是为了救人于水火之中，消灭妖魔鬼怪，拯救众生的苦难啊！取经西天，得到经书的智慧和力量，归还给人间，使人间更加和谐美好。俺可是个拯救世界的大英雄，西行取经也是为了维护和平，传承正义，实现大同世界的理想啊！至于其他的好处嘛，和那一身神通广大的兄弟们一起闯荡天下，开开心心的做好事，还能增长自己的见识和修行，岂不美哉！总之，取经西天是为了造福众生，成就大道，一展英雄风采！  


## 回复创意性设置为 0.8

可以明显感受到当创意性为 0.8 的时候，AI 回复的答案内容角度会更多，出现了

“实现世界大同的理想”这类不常见的目的，而创意性为 0，则更偏向于更加严谨的

答案，如“保护师傅/斩妖除魔”等目的。

另外，我们在设计 Agent 的过程中，往往需要不断优化提示词设计，以及调整相关参数，最终就可以得到一个比较满意的专属 AI Agents。

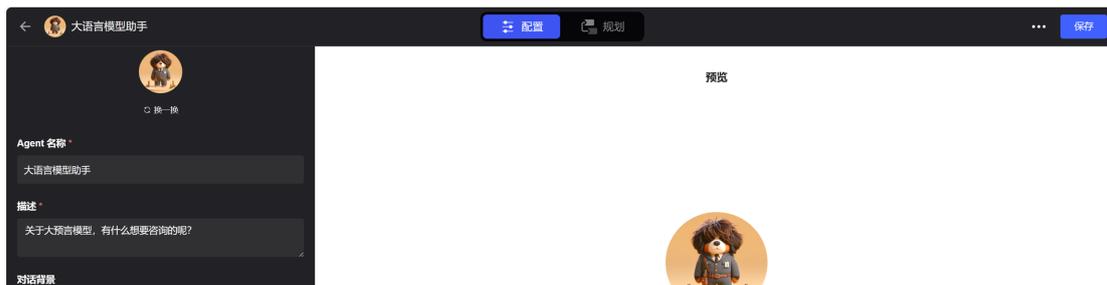
## 初级：知识库 Agent

知识库 Agent 是目前另外一类比较受欢迎的工具，它能将个人/企业专有数据与大模型相结合，进一步扩展大模型的知识库范围，让它与业务结合得更加紧密。

在 Agent 的配置过程中，如果在 Agent 设置页面中选择关联了“知识库”，我们就能够在规划页面中看到新增加的“知识库搜索”模块，本次案例将着重为大家介绍“知识库搜索”模块的功能设计。

## 基础配置

本次我们准备做一个“大预言模型助手”的 Agent，将整理好的大语言模型相关资料作为知识库，这个 Agent 将充当大语言模型方面的专家，回答我们关于大语言模型方面的提问。具体的配置如下：



## 画布规划

在 Agent 的画布规划中，此时会增加了一个“知识库搜索”模块，来实现借助“知识库”中上传的资料，辅助 AI 完成大模型相关问题的回复。它的具体逻辑如下：

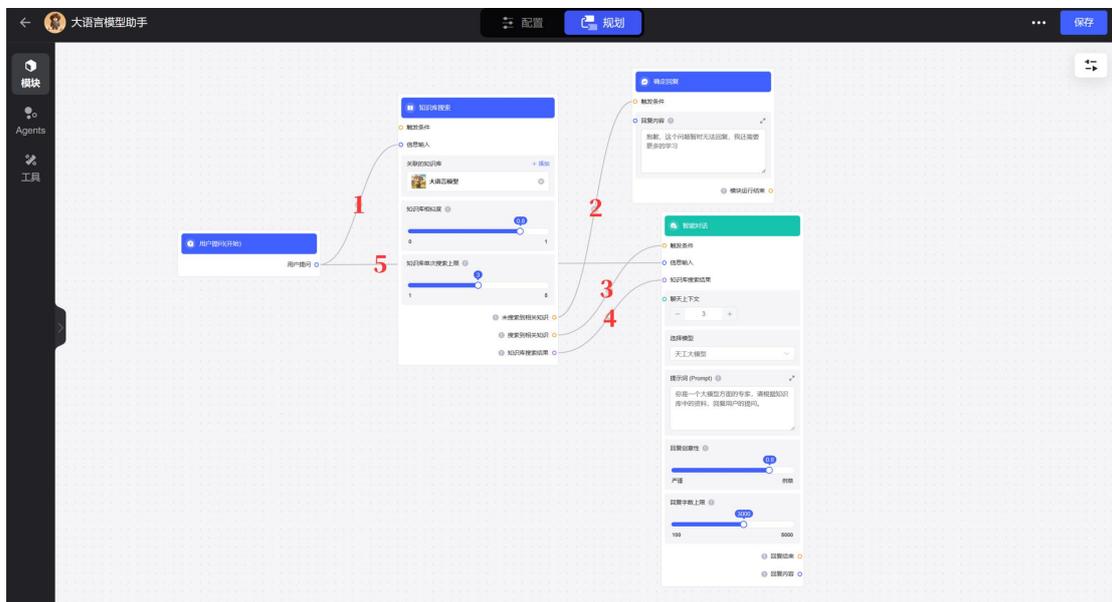
- **连线 1：**通过“用户提问”模块，获取用户输入信息，并通过“用户提问”输出
- “知识库搜索”模块通过“信息输入”输入用户的问题，后会执行如下几个步骤：

- 将“用户提问”进行向量化，并与“大语言模型”的知识库中，与已有的内容切片进行向量检索
- 将搜索结果进行相似度排序，将相似度排名靠前的切片进行返回

在搜索完成后，“知识库搜索”模块的 3 个输出连线分别走向了 2 个模块，逻辑如下：

- **连线 2：**如果“知识库搜索”结果，没有符合条件的切片内容，将触发“确定回复”模块，回复用户“抱歉，这个问题暂时无法回复，我还需要更多的学习”

- **连线 3:** 如果 “知识库搜索” 结果，搜索到有符合条件的切片内容，将触发 “智能对话” 模块，并将提示词、 “知识库搜索结果” (**连线 4**) 与 “用户提问” (**连线 5**) 一同传给 “智能对话” 模块进行处理。
- “智能对话” 模块接受到以上 3 个信息输入后，后会执行如下几个步骤：
  - 将提示词、用户提问及知识库搜索结果传给天工大模型
  - 大模型进行加工处理，并回复用户



在 “知识库搜索” 模块中，我们可以看到 “知识库相似度” 和 “知识库单次搜索上限” 这两个参数，也就是会控制知识库向量搜索的结果（最终匹配切片的数量）和返回结果（返回切片的数量），在使用过程中相关数值需要进行调整已确保知识库反馈更优的结果。

### 知识库相似度

控制用户提问与搜索结果之间的相似度，相似度数值越高，搜索出的匹配结果可能越少。因此对于用户问题可根据知识库内容，和用户提问的内容，思考场景。通常建议

相似度调整在 0.6-0.9 左右。

## 单次搜索上限

当需要在知识库进行搜索时，设置最多取 x 条作为搜索结果输出。请注意，所取结果并非越多越好，受模型最大 Tokens 和知识库内容切片大小等因素的影响，具体可用数量可通过调整参数并测试，选择更合适的上限数值。



## 中级：信息分类

除了知识库外，平台还提供多种文本处理模块，用于更加多元的场景中。

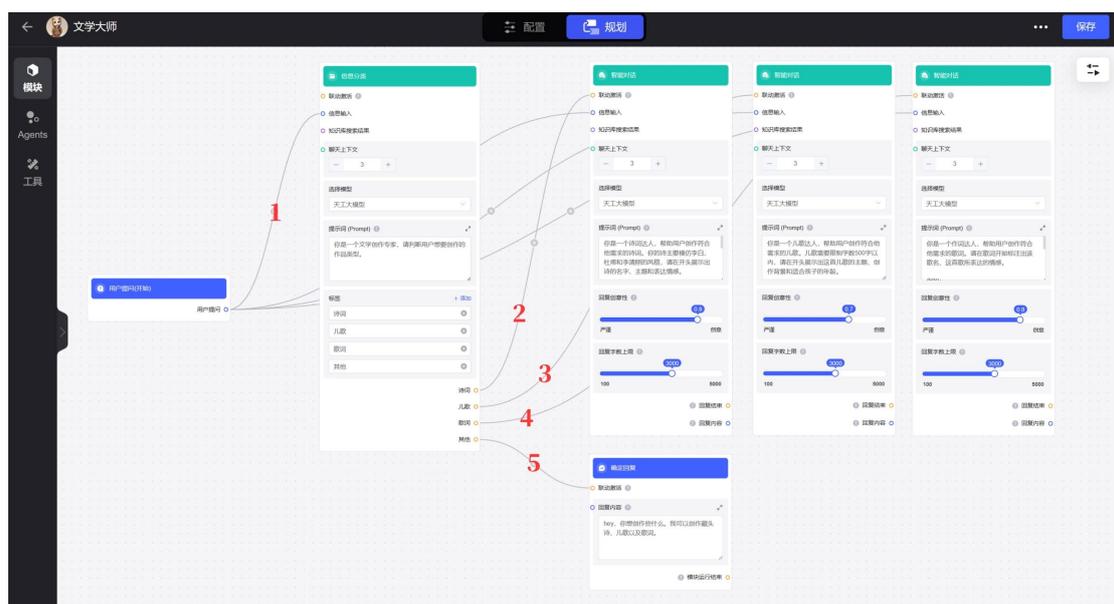
例如“信息分类”模块，能够通过 AI 对信息进行分类，从而针对分类进行不同流程。“字段提取”模块可以将信息中指定字段提取出来，便于在后续需要使用字段相关的场景中使用。“信息加工”模块能够将信息进行加工如组合处理。

## 画布规划

本次我们以“文学大师”这个 Agent 来演示 AI 如何处理不同的诗歌制作需求，并

以不同的格式实现输出。因此 Agent 的整体规划是首先就需要一个“信息分类”模块，来将用户意图进行分类，再根据不同的分类执行文案创作。

- **连线 1:** 通过“用户提问”模块，获取用户输入信息，并通过“用户提问”输出
- “信息分类”模块通过“信息输入”输入用户的问题，后会执行如下几个步骤：
  - 将模块的 prompt 与“用户提问”、聊天上下文进行传给天工大模型进行分类处理
  - 返回分类处理结果，并触发下一个模块的功能
- **连线 2-4:** 接下来，根据分类结果，将触发不同的智能对话功能。在本次 Agent 中，3 个“智能对话”将分别进行“诗词/儿歌/歌词”的创作，完成用户需求的输出。
- **连线 5:** 在“信息分类”的模块中，我们建议增加“其他”分类，来承接对于未能识别或没有定义的分类，以便引导用户更加清晰明确的表达自己需求意图。因此当用户的消息不属于以上分类时，触发“确定回复”模块，并回复指定内容。



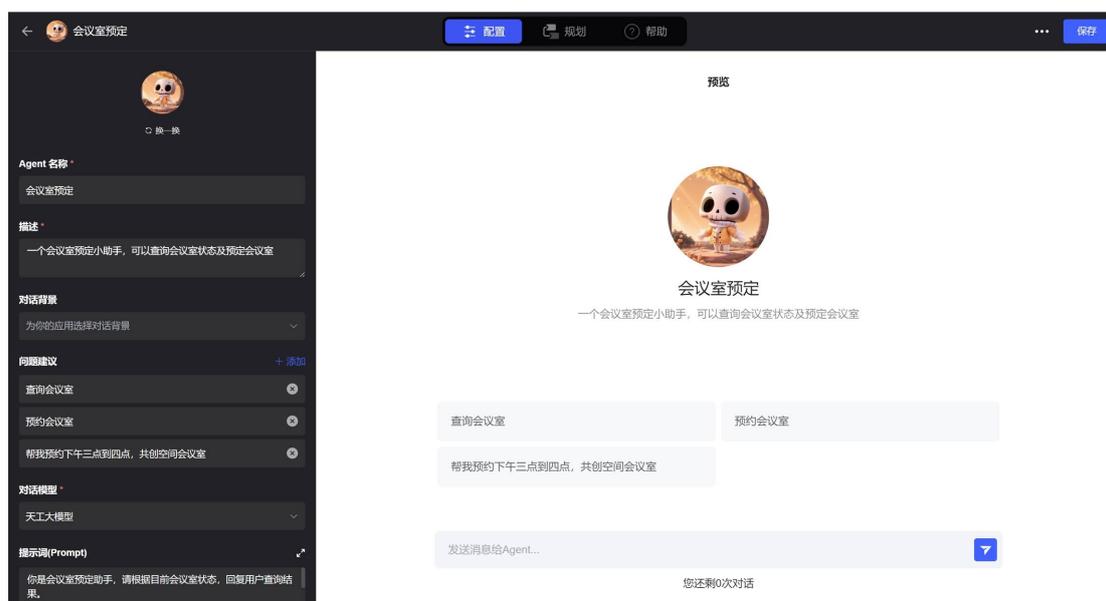
## 高级：字段提取+ HTTP 模块

“字段提取”模块能够将输入的内容进行分析并提取出所需字段，“HTTP 调用”工具，则可以连接到第三方，通过设置入参和出参，完成与第三方系统的通信，大大扩展 Agent 的使用场景。

在本次的案例中，我们将主要利用“字段提取”和“HTTP 调用”模块进行一个简单的会议室预定 Agent 的搭建，实现通过自然语言的方式，借助 Agent 完成会议室的预定。

## 基础配置

作为“会议室小预定”Agent，我们希望它能够实现查询会议室状态和预约会议室功能。其中查询和预约会议室的功能需要和现有的会议室预约系统通信，它的基本信息配置如下：

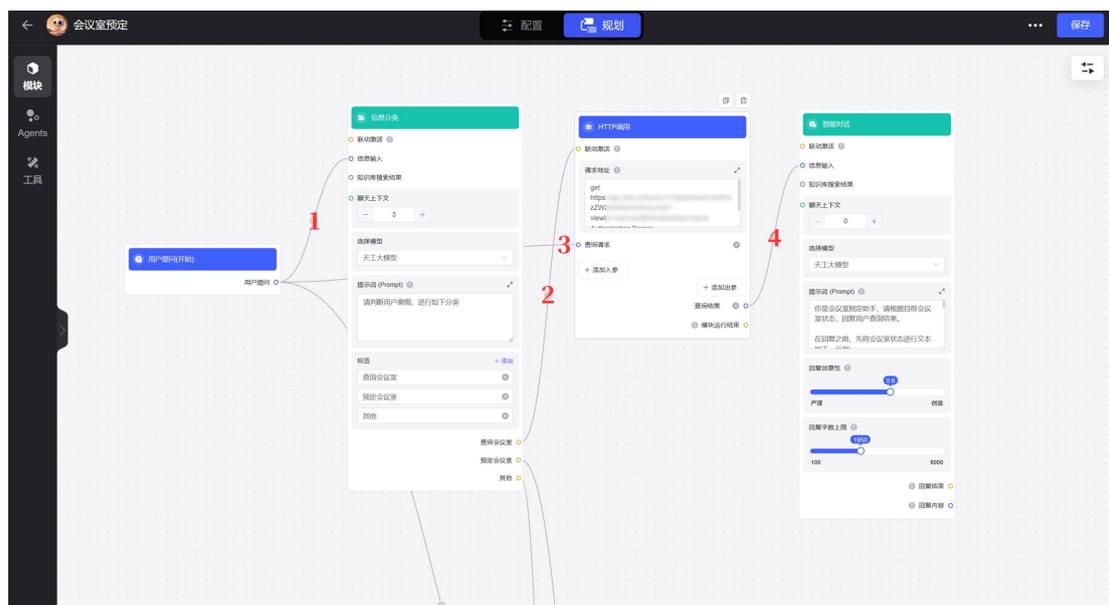


## 画布规划

Agent 的大概思路是：通过“信息分类”模块判断用户意图（查询会议室/预约会议室/其他），根据不同的用户意图，触发不同的数据通信，并执行后面的步骤。

## 查询会议室功能规划

下方为“查询会议室”功能的规划图，核心流程是通过“信息分类”模块判断用户的意图，如果是“查询会议室”相关的请求，将触发接下来的“HTTP 调用”，向会议室系统查询当前的会议室预约状态。

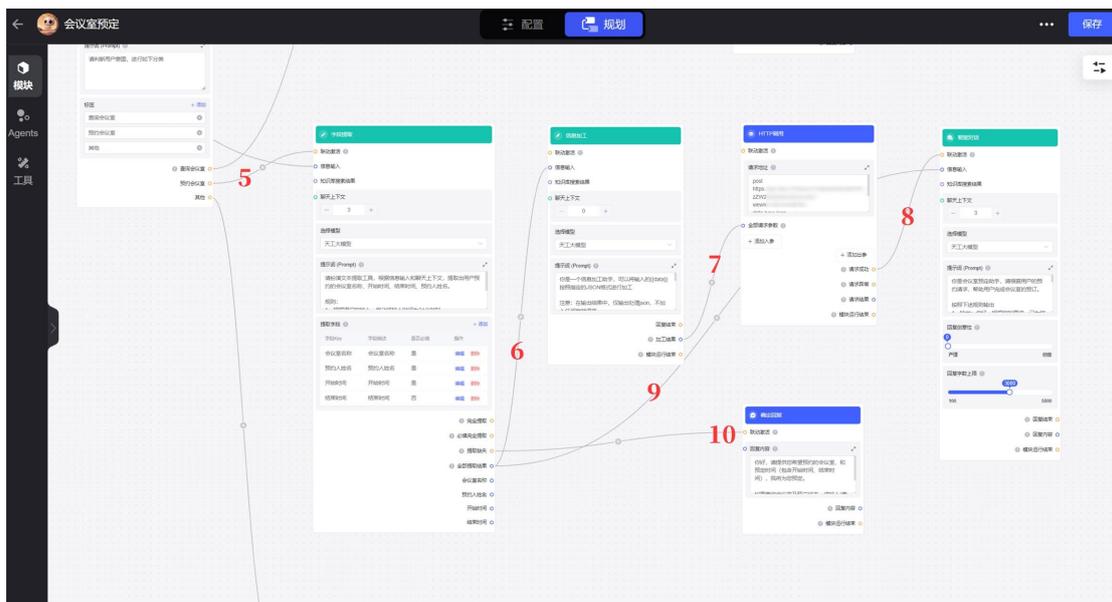


- **连线 1:** 通过“用户提问”模块，获取用户输入信息，并通过“用户提问”输出
- “信息分类”模块通过“信息输入”输入用户的问题，后会执行如下几个步骤：
  - 将模块的 prompt 与“用户提问”、聊天上下文进行传给天工大模型进行分类处理
  - 判断用户信息的类型，返回分类处理结果，并触发下一个模块的功能

- **连线 2:** 如果用户意图是“查询会议室”，将触发“HTTP 调用”模块
  - “HTTP 调用”模块中填写了“查询会议室”的请求链接，一方面接受查询（**连线 3**），一方面输出查询结果（**连线 4**），将目前会议室的状态信息传入下一个模块
- **连线 4:** “智能对话”模块接收会议室预约状态信息后，将通过
- “智能对话”模块接受到以上 3 个信息输入后，后会执行如下几个步骤：
  - 将提示词、用户提问及 HTTP 调用查询结果传给天工大模型
  - 大模型进行加工处理，并根据提示词回复信息给用户

## 预定会议室功能规划

而在“预定会议室”模块中，通过“信息分类”模块判断用户的意图，如果是“预约会议室”相关的请求，将触发接下来的流程，具体连线及功能说明如下：

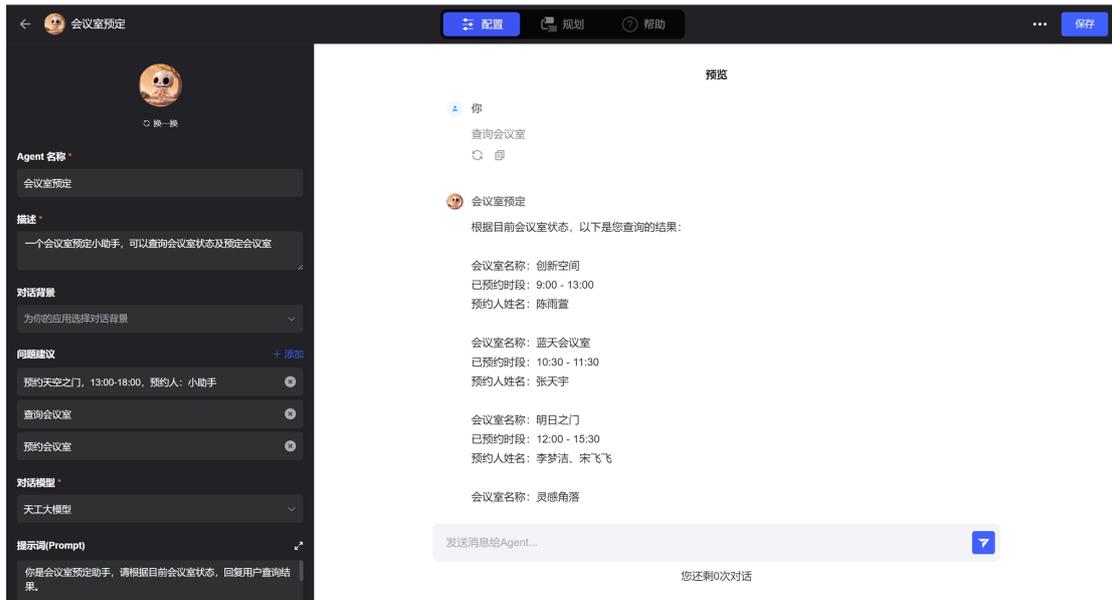


- **连线 5:** 如果用户意图是“预约会议室”，将触发“字段提取”模块

- “字段提取”模块的主要功能是将用户输入的信息进行结构化，并输出 JSON 格式的内容
- 例如输入“预约天空之门，13:00-18:00，预约人：小助手”，那么经过字段提取后的内容为“{"会议室名称":"天空之门","开始时间":"13:00","结束时间":"18:00","预约人姓名":"小助手"}”
- **连线 6**：为了更好的将预约信息传送给会议室预约系统，本次额外增加“信息加工”模块，希望将上方传入的 JSON 进一步处理成接口所需要的格式，通过 **连线 7** 输出给“HTTP 调用”模块，传入会议室预约系统。该步骤可根据实际需要选择性添加
- **连线 8**：一旦发起预约请求成功，会触发“智能对话模块”，提醒用户已预约完成，并同步用户预约信息（通过 **连线 9** 将用户的预约信息同步）
- **连线 10**：会议室预约需要用户提供必要的用户信息，这里为了防止用户提供的信息缺失，设置了“提取缺失”的提示，提醒用户提供必要的信息

## 对话测试

- 输入“查询会议室”后，Agent 可以返回当前会议室的相关预定情况



- 输入“预定会议室”后, Agent 会提示用户输入相关预定信息



- 输入“预约天空之门, 下午四点, 预约人小助手”后, Agent 会将信息发送给会议室预约系统, 并在请求成功后, 提示“预约成功”相关信息



以上就是一个简单的会议室预订 Agent 的搭建介绍, 灵活利用“字段提取”模块与“HTTP 调用”模块, 创建出更多符合个人/工作场景的 Agent。