请先仔细阅读这份文档：工作流介绍

工作流是一系列可执行指令的集合，用于实现业务逻辑或完成特定任务。它为应用/智能体的数据流动和任务处理提供了一个结构化框架。工作流的核心在于将大模型的强大能力与特定的业务逻辑相结合，通过系统化、流程化的方法来实现高效、可扩展的 AI 应用开发。​

扣子提供了一个可视化画布，你可以通过拖拽节点迅速搭建工作流。同时，支持在画布实时调试工作流。在工作流画布中，你可以清晰地看到数据的流转过程和任务的执行顺序。​

工作流与对话流​

扣子提供以下两种类型的工作流：​

工作流（Workflow）：用于处理功能类的请求，可通过顺序执行一系列节点实现某个功能。适合数据的自动化处理场景，例如生成行业调研报告、生成一张海报、制作绘本等。​

对话流（Chatflow）：是基于对话场景的特殊工作流，更适合处理对话类请求。对话流通过对话的方式和用户交互，并完成复杂的业务逻辑。对话流适用于 Chatbot 等需要在响应请求时进行复杂逻辑处理的对话式应用程序，例如个人助手、智能客服、虚拟伴侣等。​

智能体和应用均可以绑定对话流，但智能体不支持对话节点，无法绑定包含对话节点的对话流。​

​

节点​

工作流的核心在于节点，每个节点是一个具有特定功能的独立组件，代表一个独立的步骤或逻辑。这些节点负责处理数据、执行任务和运行算法，并且它们都具备输入和输出。每个工作流都默认包含一个开始节点和一个结束节点。​

开始节点是工作流的起始节点，定义启动工作流需要的输入参数。​

结束节点用于返回工作流的运行结果。​

通过引用节点输出，你可以将节点连接在一起，形成一个无缝的操作链。例如，你可以在代码节点的输入中引用大模型节点的输出，这样代码节点就可以使用大模型节点的输出。在工作流画布中，你可以看到这两个节点是连接在一起的。​

​

​​

​

​

在使用节点编排工作流时，灵活性和扩展性是实现高效编排的关键。工作流的开始节点、结束节点、输出节点、插件节点、子工作流节点、代码节点、SQL 自定义节点、新增数据节点、查询数据节点、更新数据节点、删除数据节点、问答节点、批处理节点、循环节点、变量聚合节点、变量节点、选择器节点均支持多种变量类型，包括 String、Integer、Number、Boolean、Object、File 和 Array等。你可以根据实际需求灵活选择合适的数据类型，而无需额外的数据转换，从而提升工作流编排的灵活性和扩展性。​

权限​

工作流的所有者享有工作流的全部权限，包括编辑、发布和删除自己创建的工作流。团队成员可以查看和使用团队内的所有工作流，而团队所有者和管理员有权删除工作流。​

默认情况下只有工作流的所有者才能编辑工作流，如需和其他团队成员一起编排工作流，所有者可以为工作流开启多人协作模式，并添加协作者，详情可参考​权限说明。​

​工作流与对话流

工作流用于处理功能类的请求，可通过顺序执行一系列节点实现某个功能。对话流是基于对话场景的特殊工作流，专门用于处理对话类请求。在应用中添加对话流，将对话中的用户指令拆分为一个个步骤节点，并为其设计用户界面，你可以搭建出适用于移动端或网页端的对话式 AI 应用，实现自动化、智能化的对话流程。​

对话流和工作流的区别​

相较于工作流而言，对话流更适合处理对话场景下的交互逻辑。每个对话流都绑定了一个会话，运行时可以从此会话中读取历史消息，同时将本次运行对话流产生的消息记录在这个会话中，相当于一个拥有了记忆的工作流。​

你如果需要搭建一个智能体，智能体本身支持上下文和会话能力，那么可以随意选择工作流或对话流，注意不要添加会话类节点即可。​

如果你需要搭建一个对话式的 AI 应用，例如 AI 助手、智能体客服等基于对话方式交互的 AI 应用，扣子推荐你使用对话流，对话流中的大模型可以读取会话上下文、管理会话，还可以搭建对话式的用户界面，发布到各种社交通讯软件中。​

如果你需要搭建一个工具类的 AI 应用，批量处理数据、实现任务流程的自动化，可以选择工作流实现。​

二者的详细具体区别如下：​

​

差异​

工作流​

对话流​

场景​

用于处理功能类的请求，可通过顺序执行一系列节点实现某个功能。适合数据的自动化处理场景，例如生成行业调研报告、生成一张海报、制作绘本等。​

基于对话场景的特殊工作流，更适合处理对话类请求。对话流通过对话的方式和用户交互，并完成复杂的业务逻辑。对话流适用于 Chatbot 等需要在响应对话请求时进行复杂逻辑处理的对话式应用程序，例如智能客服、虚拟伴侣等。​

节点​

大模型节点等涉及大模型处理的节点不支持查看对话历史。​

开始节点必须传入会话名称，以便记录用户和模型的对话历史。​

对话流中的模型类节点支持读取对话历史，会话中的上下文会和用户问题一起传递给大模型。​

角色配置​

不支持角色配置。​

支持角色配置。​

​

​​

​

​

创建对话流时，你可以为每个对话流单独配置角色名称、头像、开场白、音色、输入方式等角色属性。在以下场景中，对话服务将根据这些配置以指定的角色与用户进行交互：​

发布应用为 Chat SDK：当你将应用发布为 Chat SDK 后，智能对话服务将根据配置的角色属性与用户进行互动，提供个性化的用户体验。​

AI 对话组件绑定对话流：在应用中绑定 AI 对话组件时，选择相应的对话流，智能对话服务将以配置的角色属性与用户进行交互。​

如果 AI 对话组件配置了角色属性，则优先采用 AI 对话组件的配置；若智能体处于对话流模式并绑定了对话流，则以智能体中的角色配置为准。​

​

用户界面​

支持布局组件、展示组件等丰富的组件类型。​

目前仅支持 AI 对话组件。​

AI 对话组件可将对话流包装为一个聊天面板，展示在移动端或网页端页面中。​

发布渠道​

在 AI 应用中的工作流，支持发布到 API，工作流绑定用户界面之后，还可以发布到模板、商店。​

暂不支持发布到社交渠道、Web SDK、小程序。​

在 AI 应用中的对话流，支持发布到 API&SDK 、小程序、社交渠道、商店、模板等扣子提供的所有发布渠道。​

API 调用​

通过 API ​执行工作流或​执行工作流（流式响应）调用工作流。​

通过 API ​执行对话流调用对话流。​

​

常见问题​

对话流和工作流可以互转吗​

对话流和工作流可以互转。​

对话流转为工作流之后，大模型、意图识别等涉及模型处理的节点不支持读取对话历史，也无法绑定会话；开始节点的预置参数也会转为普通参数，可以删除或修改。​

工作流转为对话流之后，开始节点会添加两个预置参数 USER\_INPUT 和 CONVERSATION\_NAME，不可删除或修改。​

对话流和工作流的节点有什么区别​

对话流和工作流均支持扣子提供的所有节点，但节点配置略有差异。对话流的开始节点中需要指定一个会话，对话流中添加的大模型、意图识别等涉及模型处理的节点支持读取对话历史。​

详细差异如下：​

​

差异​

工作流​

对话流​

开始节点​

预置一个非必填参数，格式为 String，默认命名为 input。​

预置了以下必选参数：​

USER\_INPUT：用户在对话流中输入的原始内容。​

CONVERSATION\_NAME：对话流绑定的会话。​

大模型节点​

不支持对话历史。​

支持读取对话历史，会话中的上下文会和用户问题一起传递给大模型。​

意图识别节点​

不支持对话历史。​

支持读取对话历史，会话中的上下文会和用户问题一起传递给大模型。​

工作流使用限制

使用工作流时，应注意以下限制。​

​

限制​

说明​

超时时间​

如果工作流或节点运行超时，智能体可能无法提供符合预期的回复。各场景的超时时间说明如下：​

未开启工作流异步运行时，工作流整体超时时间为 3 分钟，模型节点 3 分钟，IDE 插件节点 1 分钟，其他类型的插件节点 3 分钟，其他节点 1 分钟。​

开启工作流异步运行后，工作流整体超时时间为 24 小时，模型节点 5 分钟，IDE 插件节点 1 分钟，插件节点 3 分钟，其他节点 1 分钟。​

一次对话中，智能体连续两次回复的时间间隔限制为 2 分钟。​

运行次数​

专业版无免费额度，且不限制运行次数，按照消耗的模型 token 计费。基础版每个工作流每天最多运行 500 次，否则会报错。试运行指通过扣子平台的工作流编排页面试运行工作流。​

​

​​

​

​

节点总数​

每个工作流中最多添加 1000 个节点。每次运行工作流时，包含循环节点的循环次数在内，所有节点共计最多执行 1000 次。​

图像节点并发​

每次执行工作流时，叠图和添加文字插件节点的并发限制为单节点 10 次/秒。其他官方图像插件和图像节点的并发限制为单节点 4 次/秒。​

​使用工作流

工作流是一系列可执行指令的集合，用于实现业务逻辑或完成特定任务。你可以在智能体和应用搭建中通过工作流实现特定的任务或指令。​

搭建工作流​

无论是在智能体还是应用中使用工作流，都需要先创建一个可运行的工作流。​

步骤一：创建工作流​

登录扣子平台。​

在左侧导航栏中选择工作空间，并在页面顶部空间列表中选择个人空间或团队空间。​

系统默认创建了一个个人空间，该空间内创建的资源例如智能体、插件、知识库是你的私有资源，其他用户不可见。你也可以创建团队或加入其他团队，团队内的资源可以和其他团队成员共享。更多信息，请参考​工作空间概述。​

在资源库页面右上角单击 +资源，并选择工作流。​

设置工作流的名称与描述，并单击确认。​

清晰明确的工作流名称和描述，有助于大语言模型更好的理解工作流的功能。​

​

创建后页面会自动跳转至工作流的编辑页面，初始状态下工作流包含开始节点和结束节点。​

开始节点用于启动工作流。​

结束节点用于返回工作流的运行结果。​

步骤二：编排工作流​

创建工作流后，你可以通过拖拽的方式将节点添加到画布内，并按照任务执行顺序连接节点。​

工作流提供了基础节点供你使用，除此之外，你还可以添加插件节点来执行特定任务。​

在左侧面板中选择要使用的节点。​

将节点拖拽到画布中，并与其他节点相连接。​

配置节点的输入和输出参数。​

步骤三：测试并发布工作流​

要想在智能体内使用该工作流，则需要发布工作流。​

单击试运行。​

运行成功的节点边框会显示绿色，在各节点的右上角可查看节点的输入和输出。​

单击发布。​

在智能体中添加工作流​

添加工作流​

前往当前团队的智能体页面，选择进入指定智能体。​

在智能体编排页面的工作流区域，单击右侧的加号图标。​

在添加工作流对话框，在我创建的页面选择自建的工作流。​

​

​​

​

​

在智能体的人设与回复逻辑区域，引用工作流的名称来调用工作流。​

​

​​

​

​

设置工作流异步运行​

工作流默认为同步运行，即智能体必须在工作流运行完毕后才会将工作流的输出传递给智能体用户。如果工作流复杂，或包含一些运行耗时长的节点，可能会导致工作流整体运行耗时长，智能体判断为工作流运行超时，在其运行完毕前就结束对话。例如包含图像流节点、多个大模型节点，或编排逻辑复杂的工作流节点。​

如果工作流或节点运行超时，智能体可能无法提供符合预期的回复。各场景的超时时间说明如下：​

未开启工作流异步运行：​

工作流整体超时时间为 3 分钟，模型节点 3 分钟，其他节点 1 分钟。部分插件节点超时时间可能 3分钟。​

用户问题和智能体回复的时间间隔最长为 2 分钟，如果智能体回复耗时 2 分钟以上，智能体可能会判断工作流超时。如果工作流运行耗时大于 2 分钟，建议添加消息节点用于输出中间消息，或者结束节点开启流式输出，保持对话状态。​

开启工作流异步运行后：工作流整体超时时间为 24 小时，模型节点 5 分钟，插件节点 3 分钟，其他节点 1 分钟。​

工作流异步运行，仅在调试智能体或与商店中的智能体对话时生效，飞书、豆包等渠道暂不支持工作流异步运行。 ​

工作流开启异步运行后，模型节点无法查看 Bot 对话历史。​

​

在这种场景下，你可以设置工作流为异步运行，设置后，智能体对话不依赖工作流的运行结果。工作流异步运行时会默认返回一条预设的回复内容，用户可以继续与智能体对话，工作流运行完毕后智能体会针对触发工作流的指令做出最终回复。​

开启异步运行：​

在指定工作流右侧单击设置。​

​

​​

​

​

开启异步运行，并设置回复内容。​

回复内容是工作流在异步运行时，智能体回复用户的默认文案。​

​

​​

​

​

异步运行效果：​

​

​​

​

​

在应用中添加工作流​

扣子支持在项目中创建一个新的工作流或复制一个已有的工作流使用。​

在资源列表中，找到工作流，然后选择一种添加方式。​

新建工作流：在该项目中创建一个新的工作流。​

新创建的工作流只能在项目中使用，无法共享给其他项目使用。​

引入资源库文件：复制一个项目所属的团队空间内已发布的工作流到该项目中使用。​

复制后，你可以对这个工作流进行修改。在项目中对工作流的修改不影响资源库中的工作流。​

​

​​

​

​

​

添加工作流后，你可以根据实际需求修改节点配置，或新增新的节点。​

管理工作流​

复制工作流​

在某一工作流的编辑页面，单击右上角的创建副本图标，可以将该工作流复制到你的工作流列表中。​

支持跨画布复制节点。​

在画布中选择并复制已配置好的节点，然后切换到目标画布，直接粘贴即可。​

删除工作流​

对于不再需要使用的工作流，你可以在工作流列表内找到该工作流，并在操作列单击删除图标。​

如果工作流已添加至智能体，在删除时会同步删除智能体中的工作流。​

​

​

​​

​

​

​

开始和结束节点

开始节点用于开启触发一个工作流，而结束节点用于输出工作流的结果。​

开始节点​

开始节点是工作流的起始节点，用于设定启动工作流需要的输入信息。开始节点只有输入参数，没有输出等其他参数。开始节点中默认有一个输入参数 BOT\_USER\_INPUT，表示用户在本轮对话中输入的原始内容。你也可以按需添加其他参数。​

开始节点配置说明如下：​

数据类型：开始节点支持配置 String、Number 等多种类型的输入参数。其中 Object 类型的参数最多支持 3 层嵌套。​

参数设置方式：支持直接添加参数并设置参数名称，也支持导入 JSON 数据，批量添加输入参数。​

如下图所示，点击导入图标后，在展开的面板中输入 JSON 数据，然后单击同步JSON到节点就可以自动导入输入参数。​

​

​​

​

​

参数描述：参数的描述信息，帮助模型理解传入的参数含义。将工作流绑定到智能体中使用时，模型会自动分析用户的 Query，将 Query 中表达的信息填入对应的参数中。​

是否必选：参数是否必选。如果未指定必选参数，无法开始执行工作流。将工作流绑定到智能体中使用时，用户 Query 中如果缺少必选参数，则不会触发工作流。​

结束节点​

结束节点是工作流的最终节点，用于返回工作流运行后的结果。结束节点支持两种返回方式，即返回变量和返回文本。​

返回变量​

返回变量模式下，工作流运行结束后会以 JSON 格式输出所有返回参数，适用于工作流绑定卡片或作为子工作流的场景。如果工作流直接绑定了智能体，对话中触发了工作流时，大模型会自动总结 JSON 格式的内容，并以自然语言回复用户。​

返回文本​

返回文本模式下，工作流运行结束后，智能体中的模型将直接使用指定的内容回复对话。回答内容中支持引用输出参数，也可以设置流式输出。具体说明如下：​

​

设置​

说明​

输出变量​

输出节点中输出的参数。为工作流绑定卡片时可以使用这些参数。​

回答内容​

工作流的最终输出内容，不可设置为空。​

支持引用输出参数，引用方式为{{变量名}}。​

​

​大模型节点

大模型节点是扣子提供的基础节点之一，你可以在该节点使用大语言模型处理任务。​

节点说明​

大模型节点可以调用大型语言模型，根据输入参数和提示词生成回复，通常用于执行文本生成任务，例如文案制作、文本总结、文章扩写等。​

大模型节点依赖大语言模型的语言理解和生成能力，可以处理复杂的自然语言处理任务，你可以根据业务场景的需求选择不同的模型，并配置提示词来定义模型的人设和回复风格。为了更精准地控制模型生成的结果，你还可以在大模型节点中设置模型的参数，从而影响模型回复的文本长度、内容的多样性等。​

配置大模型节点​

技能​

支持为大模型节点配置技能，添加插件、工作流或知识库，扩展模型能力的边界。大模型节点运行时，会根据用户提示词自动调用插件、工作流或知识库，综合各类信息输入后输出回复。​

配置技能后，大模型节点的能力更接近一个独立运行的智能体，可以自动进行意图识别，并判断调用技能的时机和方式，大幅度提高此节点的文本处理能力和文本生成效果，简化工作流的节点编排。例如用户需求是某地区的穿搭推荐，通常需要先通过插件节点查询某地天气，再由模型节点根据天气情况生成穿搭推荐，现在你可以直接在大模型节点添加查询天气的插件，大模型会自动调用插件，查询天气并推荐穿搭。​

模型​

选择要使用的模型。此节点的输出内容质量很大程度上受模型能力的影响，建议根据实际业务场景选择模型。可选的模型范围取决于当前的账号类型，基础版账号可以使用默认的几类模型，且存在对话数量限制，专业版账号可以使用火山引擎方舟平台的模型。​

你还可以单击配置图标，调整模型配置。模型配置的详细说明可参考​设置模型。​

输入​

需要添加到提示词中的动态内容。系统提示词和用户提示词中支持引用输入参数，实现动态调整的效果。添加输入参数时需要设置参数名和变量值，其中变量值支持设置为固定值或引用上游节点的输出参数。​

在多轮对话场景中，你还可以开启智能体对话历史。执行此节点时，扣子会将智能体与当前用户的最近多条对话记录和提示词一起传递给大模型，以供大模型参考上下文语境，生成符合当前对话场景的回复。一问一答场景下通常无需开启此功能。​

系统提示词​

模型的系统提示词，用于指定人设和回复风格。支持直接插入提示词库中的提示词模版、插入团队资源库下已创建的提示词，也可以自行编写提示词。编写方式可参考​编写提示词。​

编写系统提示词时，可以引用输入参数中的变量、已经添加到大模型节点的技能，例如插件工具、工作流、知识库，实现提示词的高效编写。例如{{variable}}表示直接引用变量，{{变量名.子变量名}}表示引用 JSON 的子变量，{{变量名[数组索引]}}表示引用数组中的某个元素。​

用户提示词​

模型的用户提示词是用户在本轮对话中的输入，用于给模型下达最新的指令或问题。用户提示词同样可以引用输入参数中的变量。​

输出​

指定此节点输出的内容格式与输出的参数。输出格式支持设置为：​

文本：纯文本格式。此时大模型节点只有一个输出参数，参数值为模型回复的文本内容。​

Markdown：Markdown 格式。此时大模型节点只有一个输出参数，参数值为模型回复的文本内容。​

JSON：标准 JSON 格式。你可以直接导入一段 JSON 样例，系统会根据样例格式自动设置输出参数的结构，也可以直接添加多个参数并设置参数类型。​

参数的名称和描述有助于模型在参数中正确返回匹配的内容。当存在多个输出参数时，建议为输出参数指定有意义的名称，并设置描述信息。例如用于改写 Query 的模型节点，可以设置输出参数为 new\_query，描述是改写后的 Query，另一个参数为 reason，描述是改写原因。​

异常设置​

大模型节点支持异常忽略功能。开启此功能后，如果试运行工作流时此节点运行失败，工作流不会中断，而是继续运行后续下游节点。如果下游节点引用了此节点的输出内容，则使用此节点预先配置的默认输出内容。​

​

​​

​

​

常见问题​

大模型节点配置技能，和直接使用插件等技能节点有什么区别？​

区别如下：​

大模型节点配置技能时，模型会根据用户的 Query 自动判断调用技能的时机与方式、判断执行技能时传递给技能的输入，整体流程更加灵活，是模型 Function call 能力的直接体现。​

直接使用插件等技能节点时，工作流是开发者人工设计并配置的执行流程，调用技能的时机和方式是固定的，技能的输入也是开发者的指定输入内容，如果编排合理，相对于模型配置技能的场景整体效果更加稳定。​

为什么模型输出和节点输出不一致，应如何处理​

参数的名称和描述有助于模型在参数中正确返回匹配的内容。当存在多个输出参数时，建议为输出参数指定有意义的名称，并设置描述信息。例如用于改写 Query 的模型节点，可以设置输出参数为 new\_query，描述是改写后的 Query，另一个参数为 reason，描述是改写原因。​

​

​​

​插件节点

插件节点用于在工作流中调用插件运行指定工具。​

插件是一系列工具的集合，每个工具都是一个可调用的 API。商店中的上架插件或已创建的个人或团队插件支持以节点形式被集成到工作流中，拓展智能体的能力边界。​

输入与输出​

插件节点的输入和输出结构取决于插件工具定义的输入输出结构，不支持自定义设置。在插件节点中你需要为必选的输入参数指定数据来源，支持设置为固定值或引用上游节点的输出参数。​

插件节点运行时，会调用工具处理输入参数，并根据工具定义输出处理后的数据。你可以在输出区域右上角单击查看示例，查看输出参数的详细说明、完整的输出示例。​

模拟集​

试运行工作流时，插件节点的输出默认使用真实的输出数据，你也可以选择使用模拟集的数据。模拟集是插件的模拟输出结果，每次试运行工作流时无需调用插件，直接使用数据集的数据作为后续节点的输入。支持自定义设置或 AI 自动生成模拟集。​

异常设置​

插件节点支持忽略异常功能。开启此功能后，如果试运行工作流时此节点运行失败，工作流不会中断，而是继续运行后续下游节点。如果下游节点引用了此节点的输出内容，则使用此节点预先配置的默认输出内容。​工作流节点

扣子提供工作流节点，实现工作流嵌套工作流的效果。​

节点说明​

在一个工作流中，你可以将另一个工作流作为其中的一个步骤或节点，实现复杂任务的自动化。例如将常用的、标准化的任务处理流程封装为不同的子工作流，并在主工作流的不同分支内调用这些子工作流执行对应的操作。工作流嵌套可实现复杂任务的模块化拆分和处理，使工作流编排逻辑更加灵活、清晰、更易于管理。​

输入与输出​

工作流节点的输入和输出结构取决于子工作流定义的输入输出结构，不支持自定义设置。在工作流节点中你需要为必选的输入参数指定数据来源，支持设置为固定值或引用上游节点的输出参数。​

批处理​

工作流节点默认单次运行，对于输入信息只做一次处理。你也可以设置此节点使用 Batch processing 模式，按照配置多次运行。每次运行都会分配参数值，直到达到次数限制或者列表的最大长度。​

工作流节点批处理模式通常用于批量执行一系列任务，支持同时处理多个输入，可显著提高海量数据的运行效率。例如子工作流的功能是在文生图，单次执行工作流节点只能生成一张图片，批处理模式下可以批量生成多张图片。​

批处理模式下，还可以设置以下参数：​

批处理次数上限：批处理运行的次数上限，默认为 100 次。​

并行运行数量：批处理的并发限制，即同时处理的任务数量，设置为 1 表示串行执行所有任务。​

异常设置​

工作流节点支持忽略异常功能。开启此功能后，如果试运行工作流时此节点运行失败，工作流不会中断，而是继续运行后续下游节点。如果下游节点引用了此节点的输出内容，则使用此节点预先配置的默认输出内容。​

HTTP 请求节点

HTTP 请求节点允许用户通过 HTTP 协议发送请求到外部服务，实现数据的获取、提交和交互。支持多种 HTTP 请求方法，并允许用户配置请求参数、请求头、鉴权信息、请求体等，以满足不同的数据交互需求。此外，HTTP 请求节点还提供了超时设置、重试机制，确保请求的可靠性和数据的正确处理。​

HTTP 请求节点的作用如下：​

数据获取：从外部服务获取数据，例如从 API 获取用户信息、天气数据等。​

数据提交：向外部服务提交数据，例如提交表单数据等。​

数据更新：更新外部服务中的数据，例如更新用户信息、修改订单状态等。​

数据删除：删除外部服务中的数据，例如删除用户账户、删除订单等。​

节点配置​

在使用 HTTP 请求节点实现数据获取、提交等功能时，通常需要配置 API、请求参数、请求头、鉴权、请求体等参数。为了简化这一配置过程，提高开发效率，HTTP 请求节点支持导入 cURL 功能，开发人员可以快速将 cURL 命令中的参数导入到 HTTP 请求节点。​

​

配置项​

说明​

API​

配置 API 请求地址和方法，支持以下请求方法： ​

GET：用于请求从外部服务获取数据，例如调用 API 获取用户信息、天气数据等。 ​

POST：用于向服务器提交数据，例如提交表单。 ​

HEAD：类似于 GET 请求，但服务器不返回请求的资源主体，只返回响应头。 ​

PATCH：用于在请求-响应链上的每个节点获取传输路径。 ​

PUT：用于向服务器上传资源，通常用于更新已存在的资源或创建新的资源。 ​

DELETE：用于请求服务器删除指定的资源。 ​

配置API请求地址时，可以使用变量来动态指定参数，通过输入{{即可唤出并使用该变量。​

请求参数​

请求参数是附加在 URL 后面的键值对，用于向服务器传递额外的信息。例如，在搜索请求中，可以通过请求参数传递搜索关键词。​

请求头​

请求头包含客户端的信息，如 User-Agent、Accept 等。通过配置请求头，可以指定客户端的类型、接受的数据格式等信息。​

鉴权​

通过配置鉴权信息，可以确保请求的安全性，防止未授权的访问，支持 Bearer Token 和自定义鉴权：​

Bearer Token：输入 Token 值，通常用于 OAuth2 等认证方式。​

自定义鉴权：配置自定义的认证信息，包括：​

Key：认证键。​

Value：认证值。​

Add To：选择将认证信息添加到请求头（Header）或查询参数（Query）。​

请求体​

请求体是 POST 请求中包含的数据，可以是表单数据、JSON 数据等。根据不同的请求类型和数据格式，可以选择相应的请求体格式，例如 form-data、x-www-form-urlencoded、raw text、JSON 等。​

超时设置​

通过配置超时设置，可以避免长时间等待，提高系统的响应速度。​

重试次数​

重试次数设置允许在请求失败时自动重试，提高请求的成功率。​

输出​

输出变量包括响应体、状态码和响应头。​

异常忽略​

支持异常忽略功能。开启此功能后，如果试运行工作流时此节点运行失败，工作流不会中断，而是继续运行后续下游节点。如果下游节点引用了此节点的输出内容，则使用此节点预先配置的默认输出内容。​

​

示例​

例如通过 HTTP 请求节点调用扣子的查看智能体列表接口，可以获取指定空间发布到 Agent as API 渠道的智能体列表。​

​

​​

​

​

节点配置如下：​

​

节点类型​

说明​

示例​

开始节点​

工作流的起始节点，本示例需要配置输入参数space\_id，用于指定要查看的空间。​

​

​​

​

​

HTTP 请求节点​

用于调用查看智能体列表接口，获取指定空间发布到 Agent as API 渠道的智能体列表，需要根据查看智能体列表的接口要求配置：​

API：请求地址为https://api.coze.cn/v1/space/published\_bots\_list，方法为 GET。​

请求参数：定义参数space\_id，并引用开始节点的输入参数space\_id。​

请求头：需要设置以下参数：​

Content-Type：固定值 application/json。​

Authorization：用于验证客户端身份的访问令牌，本示例以个人访问令牌为例，取值格式为 Bearer $Access\_Token。个人访问令牌可参考​添加个人访问令牌。​

鉴权：本示例无需鉴权。​

请求体：本示例设置为 none。​

超时设置、重试次数、输出：本示例保持默认配置。​

​

​​

​

​

结束节点​

选择返回变量模式，定义 output 参数，引用 HTTP 请求节点的输出参数 body。​

​

​​

​

​

​

常见问题​

如何解决 HTTP 请求超时问题？​

可以通过配置超时设置来解决 HTTP 请求超时问题。合理设置超时时间，可以避免请求长时间等待。​

如何处理 HTTP 请求失败的情况？​

可以通过配置重试次数设置来处理 HTTP 请求失败的情况。设置重试次数，可以提高请求的成功率。​

如何确保 HTTP 请求的安全性？​

可以通过配置鉴权信息来确保 HTTP 请求的安全性。

​代码节点

代码节点支持通过编写代码来生成返回值。扣子支持在代码节点内使用 IDE 工具，通过 AI 自动生成代码或编写自定义代码逻辑，来处理输入参数并返回响应结果。​

配置代码节点​

代码节点的配置参数说明如下：​

​

|  |  |
| --- | --- |
| **配置**​ | **说明**​ |
| **输入**​ | 声明代码中需要使用的变量。添加输入参数时需要设置参数名和变量值，其中变量值支持设置为固定值或引用上游节点的输出参数。​  **说明**  在代码中引用输入参数时，直接通过 params['input'] 取值即可。​  ​ |
| **代码**：​ | 代码节点中需要执行的代码片段。你可以直接编写代码，也可以通过 AI 自动生成代码。​   * 引用变量：直接使用输入参数中的变量，通过**return**一个对象来输出处理结果。​ * 函数限制：不支持编写多个函数。即使仅有一个输出值，也务必保持以对象的形式返回。​   支持 JavaScript 和 Python 两种语言。详细说明可参考​开发语言。​ |
| **输出**​ | 代码运行成功后，输出的参数。你可以根据实际需求，在输出结构中只保留必要的参数。​  **说明**  确保此处定义的参数名、类型与代码的 **return** 对象完全一致。以代码节点默认提供的代码为例，输出的参数与代码中定义的 **return** 对象完全一致。​  ​ |

​

**注意**

节点执行超时：单请求限制 60s。​

​

开发语言​

代码节点支持 JavaScript 和 Python 两种语言。​

JavaScript​

支持 V8 引擎的 11.3.244.8 版本（对应 Node.js 20.3.1 版本），并兼容 ECMAScript 2022 语法。​

代码 IDE 支持 TypeScript，提供静态语言的编码体验。​

在 JavaScript 中，仅内置了两个三方依赖库：​

* dayjs（版本 1.8.36）​
* lodash（版本 4.17.20） 。​

JavaScript 运行时遵循 [WinterCG](https://wintercg.org/) 规范，并支持一系列 Web API，包括：​

* atob()、btoa()、console、setTimeout()、clearTimeout()、structuredClone()、URL、URLSearchParams、AbortController、AbortSignal、TextEncoder、TextDecoder​
* WebStreams​
* WebCrypto（算法仅支持 AES、HMAC、SHA）​

JavaScript 的三方依赖库示例代码如下：​

​

*//only dayjs and lodash are allowed*​

import dayjs from 'dayjs';​

import \_ from 'lodash';​

​

async function main({ params }: Args): Promise<Output> {​

*// get input params by this way*​

return {​

content: params.name​

};​

}​

​

Python​

基于 Python 3.11.3 的标准库，大多数模块都能正常运行。不支持的模块包括 curses、dbm、ensurepip、fcntl、grp、idlelib、lib2to3、msvcrt、pwd、resource、syslog、termios、tkinter、turtle.py、turtledemo、venv、winreg、winsound、multiprocessing*、*threading*、*sockets*、*pty 和 tty*。*​

在 Python 环境中，仅内置了两个第三方依赖库：requests\_async 和 numpy。其中，requests\_async 与 requests 类似，但在使用时需要搭配 await。​

**说明**

* Python 运行时暂不支持 Http.client 方式的请求。​
* 不支持使用除 requests\_async 、numpy 以外的第三方依赖库。​
* time.sleep() 方法由于是阻塞调用，会对代码执行性能产生影响，因此推荐使用异步版本的 asyncio.sleep() 来替代。​

​

示例如下：​

​

import requests\_async as requests​

​

async def main(args: Args) -> Output:​

*# you can get url by this way*​

url = args.params['url']​

response = await requests.get(url)​

ret = {​

'code': response.status\_code,​

'res': response.text,​

}​

return ret​

​

使用代码 IDE 编写代码​

扣子提供了网页版代码 IDE 环境供你使用，无需考虑代码部署等问题，只要关注代码逻辑即可。代码 IDE 还提供了 AI 插件，辅助你生成代码。​

在节点内的**代码**区域单击在IDE中编辑可通过代码 IDE 编辑和调试代码。​

​

​​图片包含 文本

AI 生成的内容可能不正确。​

​

常用操作说明如下：​

​

|  |  |
| --- | --- |
| **操作**​ | **说明**​ |
| 使用 AI 生成代码​ | 你可以在 IDE 底部单击**尝试 AI**，并输入自然语言设定代码逻辑，AI 将自动生成代码。你也可以选中代码片段，通过快捷键（macOS 为 Command + I、Windows 为 Ctrl + I）唤起 AI，并输入自然语言让 AI 帮助你修改代码。​ |
| 补全代码​ | 如果你已经为代码节点配置好了输入参数，编辑时支持自动补全参数。​ |

​

试运行代码节点​

试运行代码节点时，填写输入参数，并单击**运行**。平台会自动执行代码片段，并展示运行结果。运行结果中包括输入参数和输出内容，其中输出内容由原始输出和最终输出组成。​

* **原始输出**：代码执行的真实输出结果，包含代码中定义的所有输出参数。你也可以单击 Sync output to Node，将原始输出的参数结构同步到代码节点。​
* **最终输出**：按照代码节点中指定的**输出**参数结构生成的输出结果。​

​

​​图片包含 矩形

AI 生成的内容可能不正确。​

​

常见问题​

可以在代码节点中引入依赖包吗？​

代码节点仅支持引用以下第三方依赖，引入其他依赖的语句不生效。​

* JavaScript：dayjs、lodash​
* Python：requests\_async 、numpy​

为什么代码节点输出了 tako\_bot\_userhistory？​

如果在代码节点中通过函数遍历所有变量，可能会识别到部分系统变量，例如系统变量 tako\_bot\_userhistory，导致代码节点运行报错。如果需要在代码中使用代码节点定义的输入变量，直接通过 params['input'] 取值即可，无需通过函数遍历。​

tako\_bot\_userhistory 为 String 格式，表示会话中的上下文消息，例如 TAKO\_BOT\_HISTORY :[{"content":"早上好","name":"小张","role":"user"}]。此系统变量即将下线，如需获取会话中的上下文消息，推荐使用​查询消息列表节点节点。​

代码节点中如何使用 sleep 函数？​

代码节点中，推荐使用 Python 标准库的 asyncio.sleep 函数，time.sleep 函数会阻塞工作流执行、导致触发限流。​

asyncio.sleep 函数在异步编程中挂起当前任务、实现非阻塞的延迟，最长可延迟 1 分钟。此函数常用于需要在异步操作之间添加延迟的场景，例如在异步的数据处理中控制处理速率。目前工作流暂不支持异步插件节点，如果下游节点需要使用异步插件的执行结果，可以通过循环节点实现异步等待和结果轮询，其中异步等待由代码节点的 asyncio.sleep 函数实现。​

例如以下示例代码表示等待 30 秒：​

​

​​​

​

如何用 AI 生成代码？​

在 IDE 底部单击**尝试 AI**，并输入自然语言设定代码逻辑，AI 将自动生成代码。你也可以选中代码片段，通过快捷键（macOS 为 Command + I、Windows 为 Ctrl + I）唤起 AI，并输入自然语言让 AI 帮助你修改代码。​

​

​​​

​

常见的 AI 生成代码的场景包括：​

通过正则匹配清洗数据

拆分字符串

代码优化

* 用户提示词：通过正则匹配的方式，删除 input 中所有中文句号之后的空格。​
* AI 生成的代码：​

​

import re​

async def main(args: Args) -> Output:​

input\_str = args.params['input']​

*# 使用正则表达式匹配*​

result = re.sub(r'[\u4e00-\u9fff。]+(\s+)', '', input\_str) ​

return {'key0': result}​

选择器节点

该节点是一个 if-else 节点，用于设计工作流内的分支流程。​

当向该节点输入参数时，节点会判断是否符合**如果**区域的条件，符合则执行**如果**对应的工作流分支，否则执行**否则**对应的工作流分支。​

每个分支条件支持添加多个判断条件（且/或），同时支持添加多个条件分支，可通过拖拽分支条件配置面板来设定分支条件的优先级。​

​意图识别节点

意图识别节点能够让智能体识别用户输入的意图，并将不同的意图流转至工作流不同的分支处理，提高用户体验，增强智能体的落地效果。​

节点说明​

意图识别（Intent Recognition）指的是让智能体理解用户通过自然语言表达的意图或目的。意图识别是智能助手的典型能力，例如用户在对话中输入“我想查看今天的 AI 新闻”，其中“查看新闻”为用户意图，也就是用户希望智能体执行的操作。扣子工作流支持意图识别节点对用户意图进行归类，无需再通过大模型节点配合选择器节点实现意图识别，使工作流运行更加高效。​

意图识别节点可用于以下场景：​

* 客户服务：识别用户问题的类型，并转交各类知识库处理，对于知识库中未匹配的问题，转交人工客服处理。​
* 医疗咨询：对用户咨询的医学问题进行归类，非医学问题的咨询则拒绝回复。​
* 综合类智能体：对于功能多样的智能体，可以先由意图识别节点对用户咨询进行初步分类，转交各个 Agent 分支处理。​

意图识别节点支持极速模式或完整模式运行。​

* 极速模式：以较快的速度对用户意图进行分类，适用于关注工作流运行效率的场景。极速模式下使用扣子预设的极速模型进行意图识别，不支持修改模型，且不支持通过**高级设置**来指定模型分类的提示词。​
* 完整模式：意图识别效果优先，响应速度可能较长，适用于需要复杂逻辑判断的场景，例如剧情类游戏中根据情节发展切换剧情。完整模式往往需要选择意图识别能力较强的模型，并配合细致、严谨的提示词。​

极速模式​

极速模式下，意图识别节点配置如下：​

​

|  |  |
| --- | --- |
| **配置项**​ | **说明**​ |
| 模型​ | 默认使用扣子预设的极速模型，不支持修改模型。​ |
| 输入​ | 指定需要做意图识别判断的内容。输入参数默认为 query，可引用前置节点的输出参数，或输入指定内容。该参数通常引用开始节点中的用户输入。​  在对话流中，意图识别节点支持开启智能体对话历史，并设置会话轮数：​   * **对话历史**：开启后模型在识别意图时会参考上下文信息，进行综合判断。​ * **会话轮数**：设置带入模型上下文的对话历史轮数。轮数越多，多轮对话的相关性越高，但消耗的 Token 也越多。​ |
| 意图匹配​ | 用户意图的分类选项，支持设置多个分类，极速模式下最多可设置 10 个意图。对于匹配到这些分类的意图，处理流程会流转到对应的后续节点，若意图未匹配到此处定义的任何分类，则流转到兜底策略。​  极速模式下不支持通过**高级设置**来指定模型分类的提示词，请确保你配置的各个意图名称之间应有明确的区分度，减少交叉语义，以便模型正确分类。​ |
| 输出​ | 节点的输出参数，可作为变量被后续节点引用。​  输出参数固定为 classificationId，即每个意图的 ID。根据意图匹配中配置的意图，从上到下依次排序，第一个意图的 ID 为 1。若未命中已配置的任何意图，则 ID 为 0，执行**其他**分支。​ |

​

对话流中的意图识别节点极速模式配置示例如下：​

​

​​图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。​

​

完整模式​

完整模式下，意图识别节点配置如下：​

​

|  |  |
| --- | --- |
| **配置项**​ | **说明**​ |
| 模型​ | 选择执行意图识别的大模型，支持设置模型在此节点中的生成多样性等参数配置，使模型效果更符合你的预期。​ |
| 输入​ | 指定需要做意图识别判断的内容。输入参数默认为 query，可引用前置节点的输出参数，或输入指定内容。该参数通常引用开始节点中的用户输入。​  在对话流中，意图识别节点支持开启智能体对话历史，并设置会话轮数：​   * **对话历史**：开启后模型在识别意图时会参考上下文信息，进行综合判断。​ * **会话轮数**：设置带入模型上下文的对话历史轮数。轮数越多，多轮对话的相关性越高，但消耗的 Token 也越多。​ |
| 意图匹配​ | 用户意图的分类选项，支持设置多个分类，完整模式下最多可设置 50 个意图。对于匹配到这些分类的意图，处理流程会流转到对应的后续节点，若意图未匹配到此处定义的任何分类，则流转到兜底策略。​ |
| 高级设置​ | 追加的系统提示词。扣子平台已指定一系列提示词，用于指导大模型合理识别用户意图并分类，你也可以追加提示词、提供各个分类的用户输入示例，使模型分类更精准。​ |
| 输出​ | 节点的输出参数，可作为变量被后续节点引用。​  输出参数固定为：​   * classificationId：每个意图的 ID。根据意图匹配中配置的意图，从上到下依次排序，第一个意图的 ID 为 1。若未命中已配置的任何意图，则 ID 为 0，执行**其他**分支。​ * reason：分类的原因和依据，由模型自动生成。​ |

​

对话流中的意图识别节点完整模式配置示例如下：​

​

​​图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。​

​

配置完成后，将意图识别节点与其他节点连接，形成完整的调用链路。​

* 意图识别节点的每个意图分类，都需要与后续的处理节点相连接，否则意图命中次分类时无法触发后续的处理流程。例如在客服智能体中，产品咨询的分类可以流转至产品咨询知识库节点处理。​
* 推荐为意图识别节点设置兜底策略，若意图未匹配到此处定义的任何分类，则流转到兜底策略处理。​

示例​

以客户服务类型的工作流为例，通过意图识别节点对用户问题进行归类，并流转至不同知识库处理，对于非售前或售后的咨询，统一转交人工处理。​

核心节点如下：​

* 意图识别节点：将用户问题分为售前咨询或售后问题，每个分类提供典型示例作为分类的依据，帮助模型更精准地识别用户意图。​
* 知识库节点：售前和售后问题分别流转到售前知识库和售后知识库处理，并直接输入给结束节点。你也可以按需在知识库节点之后增加一个大模型节点，用于处理未命中知识库的问题。​
* 文本处理节点：意图识别的兜底策略，对于未命中任何分类的意图，统一输出一段文案，指导用户提交工单或联系人工客服。​

​

​​图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。​

​

常见问题​

意图分类不准确应如何处理？​

意图识别的准确性受多个因素影响，例如模型性能、分类配置、提示词等。如果发现意图分类不符合预期，你可以参考以下建议提升意图识别节点分类的准确性：​

* 调整分类：确保意图分类简洁明了，避免模糊不清或容易混淆的语义。分类之间应有明确的区分度，减少交叉语义，以降低分类混淆的可能性。例如，应避免“动物”和“动植物”这样的分类重叠。​
* 选择**完整模式**：完整模式下可以选择效果更好的模型、设置提示词，提升意图分类效果。​
* 填写**高级设置**：提供具体的用户输入示例，帮助模型更准确地理解并执行分类任务。​
* 切换模型：即使在分类和高级设置都已优化的情况下，意图识别的准确性仍然不理想，建议尝试切换不同的模型，以找到最适合您需求的模型。​

​

循环节点

循环节点用于重复执行一系列任务。​

节点说明​

循环是一种常见的控制机制，用于重复执行一系列任务，直到满足某个条件为止。扣子工作流提供循环节点，当需要重复执行一些操作，或循环处理一组数据时，可以使用循环节点实现。​

​

​​​

​

例如，以下长文总结场景中，通过循环节点依次生成各个段落，并流式输出每个段落的内容。​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

循环类型​

循环节点的配置方式取决于循环类型。循环类型是循环节点的运行模式，支持设置为使用数组循环、指定循环次数和无限循环。​

使用数组循环​

**使用数组循环**类似编程语言中的 for 语法结构。遍历循环用于遍历一个已知的序列，对序列中的每个元素执行一系列相同的步骤。遍历循环的典型场景如下：​

* 长文生成或长文总结：输入每个段落的主题，通过循环节点依次生成各个段落，将完整的文章打包输出。也可以通过消息节点实现每轮循环之后流式输出段落内容。长文总结的原理类似。​
* 问卷调查：针对多个产品进行满意度、NPS 等问卷调查，每个产品的问卷题目相同，根据每一轮问题的用户评分算出各个产品的 NPS 得分。​

在编程中，数组是一种数据结构，用于存储一系列元素。数组中的基础概念如下：​

* item：元素，即数组中的单个数据。数组由多个元素组成，每个元素可以是数字、字符、字符串等数据类型。​
* index：索引，指数组中元素的位置。索引从 0 开始计数，表示第一次循环，1 表示第二次循环，2 表示第三次循环，以此类推。最后一次循环的索引为 n-1，其中 n 为循环次数。​
* 数组长度：数组中元素的数量。​

例如，在数组 arrayOfNumbers = [10, 20, 30, 40, 50] 中：​

* arrayOfNumbers[0] 的 item 是 10，index 是 0。​
* arrayOfNumbers[1] 的 item 是 20，index 是 1。​
* 数组长度为 5。​

使用数组循环时，需要指定**循环数组**，此参数仅支持引用上游节点的输出，且必须为**数组**格式。**使用数组循环**模式下执行循环节点时，循环的次数取决于**循环数组**引用的数组长度。例如数组 [1,2,3,4,5] 的长度为 5，循环节点会循环 5 次完成遍历。如果**循环数组**参数引用了多个数组，则将最短的数组作为循环次数。​

使用数组循环时，循环节点会遍历数组中的每个元素，每次循环都会将当前循环到的元素赋值给内置变量。内置变量仅限循环节点内部使用。目前支持的内置变量如下：​

* item<String>：数组元素，即当前循环到的数组元素。​
* index<Integer>：数组索引，index+1 为当前循环的轮次。​

例如在长文总结场景中，引用循环节点数组中的 item，指示模型节点总结 item 中传入的文章段落。​

​

​​​

​

指定循环次数​

**指定循环次数**模式通常用于批量、顺序处理数据的场景，需要同时设置**循环次数。**循环次数默认为 10 次，支持设置为 1~1000 次，你也可以引用上游节点数值类型的输出参数。​

**说明**

**循环次数**参数的引用值如果大于 1000，则设置为 1000，如果小于 1，则设置为 1。​

​

无限循环​

**无限循环**类似编程语言中的 while等语法结构，需要通过**终止循环**节点停止循环。循环第一次运行之后，工作流对指定条件进行判断，满足一定条件时则结束循环，否则继续下一次循环。通常是基于循环节点的执行结果进行判断，例如循环调用插件获取数据，当插件执行失败时停止循环，否则持续执行循环只会得到同样的错误结果。​

**无限循环**适用于以下场景：​

* 批量处理数据：例如通过自定义插件调用 API，查看多个用户 ID 的数据，如果插件执行失败，则停止循环。​
* 增强搜索：基于第一次检索内容询问用户，收到用户回答后，结合用户问题进行第二次检索，循环执行多次检索，直至符合用户要求。增强检索可提高检索结果的精准度、用户满意度。​
* 回合制游戏：游戏通常包含多轮，直到满足胜利条件或游戏结束条件，否则游戏回合反复进行。​

**无限循环**需要通过**终止循环**节点停止循环。**终止循环**节点通常和条件判断节点关联使用，条件判断节点判断某个条件成立时，流转到终止循环节点，自动跳出循环。​

例如批处理场景中，可以持续调用某个插件，如果插件返回报错，则停止循环。通过选择器节点判断插件的返回数据中是否存在 error\_code，如果 error\_code 不为空，说明插件执行报错，则流转到终止循环节点，主动跳出循环。​

​

​​图片包含 图形用户界面

AI 生成的内容可能不正确。​

​

配置循环节点​

**说明**

* 多轮循环之间是严格串行执行的，不支持并发执行循环。每个循环迭代完成后，才能启动下一个循环迭代。如需并行执行某些节点，可以使用批处理。​
* 不支持嵌套循环，循环节点中不允许添加另一个循环节点，但你可以在循环节点中嵌套一个包含循环节点的工作流。​
* 循环节点中不允许添加批处理节点。​

​

**设置中间变量**​

循环节点支持设置**中间变量**，此变量可作用于每一次循环。**中间变量**通常和循环体中的**设置变量**节点搭配使用，在每次循环结束后为**中间变量**设置一个新的值，并在下次循环中使用新值。​

例如在长文生成场景中，通过**中间变量**将每一轮的段落总结作为变量传递到下次循环中，让大模型参考之前的段落内容和下一段的主题，生成一个上下文衔接更流畅的段落。​

​

​​​

​

各个节点配置如下：​

* **循环节点**：将**中间变量**设置为 last\_paragraph，参数值设置为一个空格。首次循环中生成第一个段落时，不需要参考大纲以外的任何内容，所以将循环变量的值指定为一个空格，你也可以按需设置为其他内容。​
* **循环体**中的**设置变量**节点：中间变量选择循环节点中设置的中间变量 last\_paragraph；设置值选择大模型的输出参数 output，表示开始下次循环前，将本次循环中大模型生成的段落赋值给循环变量。​
* **循环体**中的**大模型节点**节点：添加 2 个输入参数，分别引用循环节点的内置变量 item 和循环变量，并在提示词中指定生成文章段落时参考上个段落的内容。​

**说明**

在**设置变量**节点中，中间变量的数据类型必须和设置值的数据类型一致，否则无法赋值。​

​

设置循环体​

创建循环节点后，会生成一个循环节点和对应的循环体画布。循环体画布是循环节点的内部运行机制，用于编排循环的主逻辑，每个循环迭代中，工作流会依次执行画布内的各个节点。你需要在循环节点和上下游节点之间添加连线，但无需调整循环体和循环节点之间的连线。​

选中循环体时，才能向循环体中添加新节点，或拖入新节点至循环体画布。循环体中无需设置开始节点或结束节点，默认按照连接线的箭头方向依次执行各个节点。​

**说明**

* 设置变量节点、继续循环节点和停止循环节点只能在循环体中使用。​
* 不支持将循环体外部的节点拖动至循环体内，循环体中的节点也不可移动到循环体之外。​

​

设置输出​

循环节点的输出参数可设置为循环体的执行结果集合，表示当数组中所有元素运行完毕之后，将所有循环的运行结果打包输出给下游。也支持设置为循环变量的取值。​

​

​​​

​

调试循环节点​

配置循环节点之后，你还需要试运行这个节点，查看其输入输出是否符合预期。调试结束后，循环节点的运行结果中会显示循环节点在多轮循环之后汇总的输入输出内容。循环体中的每个节点也会展示每次循环中的输入输出、变量赋值内容。​

​

​​​

​

常见问题​

如何将文本内容转为数组？​

循环节点中的**循环数组**参数必须引用上游节点的输出参数，且参数类型为数组类型。大模型、代码等节点均支持数组格式的输出参数。如果你只能拿到文本格式的内容，可以通过代码节点将文本内容转为数组格式。​

例如长文总结场景下，需要通过 LinkReaderPlugin 插件提取 PDF 内容，并将其作为循环节点的循环参数。LinkReaderPlugin 插件的输出参数 pdf\_content 为 String 类型，可以通过代码节点转为数组类型。​

代码示例：​

把段落按换行符（/n）拆分成数组。​

​

async def main(args: Args) -> Output:​

params = args.params​

ret: Output = {​

"key0": params['input'].split("\n") ,​

}​

return ret​

​

​

配置示例：​

​

​​​

​

​

如何结束循环？​

循环节点模式不同，结束循环的方式不同。​

* 使用数组循环：循环节点依次对数组中的元素执行处理后，自动结束并跳出循环。例如长文生成场景中，上游节点生成一个数组格式的文档大纲，你可以将**循环数组**参数设置为这个数组，循环节点中的大模型节点对数组中每个段落进行扩写，直到所有段落扩写完毕。​
* 指定循环次数：循环节点执行指定的次数后会自动终止循环。​
* 无限循环：通过**终止循环**节点停止循环。条件判断节点判断某个条件成立时，流转到**终止循环**节点，自动跳出循环。​

如何在循环体中插入节点​

选中循环体时，才能向循环体中添加新节点，或拖入新节点至循环体画布。不支持将循环体外部的节点拖动至循环体内，循环体中的节点也不可移动到循环体之外。​

​变量赋值节点

变量赋值节点是工作流中用于修改和存储变量值的节点。​

​

​​图形用户界面, 文本, 应用程序, 电子邮件

AI 生成的内容可能不正确。​

​

节点说明​

通过变量赋值节点，将特定的值赋给变量，可以实现数据的动态更新和传递，使工作流能够根据实时数据做出相应的处理和决策。变量赋值节点应用广泛，例如：​

* **存储中间结果**：在工作流中，将中间计算或处理的结果通过变量赋值节点存储到变量中，以便后续节点使用。例如，在开发一个智能医疗诊断应用时，需要对患者的病历数据进行预处理，如提取关键症状、检查结果等信息。此时，可以设置一个变量patient\_info，将患者的病例数据通过变量赋值节点存储起来，后续在进行疾病诊断时，可以直接从patient\_info变量中获取患者的详细信息。​
* **记录用户输入**：在与用户交互的工作流中，用户的输入信息是后续处理的重要依据。通过变量赋值节点，可以将用户的输入存储到变量中。例如，在开发一个智能客服机器人时，当用户向客服机器人咨询产品问题时，机器人提取用户关心的问题的关键词，然后将这些关键词通过变量赋值节点存储到一个名为user\_query的变量中。在后续的处理节点中，机器人会根据user\_query变量中的内容，从知识库中检索相关的答案或解决方案，并生成回复文本。​
* **控制流程分支**：在工作流中，通常需要根据不同的条件来决定执行不同的分支流程。变量赋值节点可以用来设置控制流程分支的条件变量，通过赋予变量不同的值，来引导工作流走向相应的分支。例如，在开发一个个性化推荐系统时，需要收集用户的基本信息和行为数据，如年龄、性别、浏览历史等，然后分析用户的兴趣偏好，并将结果通过变量赋值节点存储到一个名为user\_preference的变量中。在生成推荐内容时，根据user\_preference变量值来决定推荐策略。​

配置变量赋值节点​

为变量赋值时，在变量赋值节点的**输入**中添加需要赋值的参数。请注意，字段名设置为需要赋值的变量名称，变量值可以设置为固定值，也可以引用上游节点的输出参数。​

**说明**

变量赋值节点可以为用户变量、应用变量赋值，但不能为系统变量赋值。​

​

配置示例​

例如，搭建一个猜谜游戏工作流，定义一个应用变量score，默认值设置为 0 。在工作流中添加变量赋值节点，每猜对一个谜语，变量score数值加 1，重新开始新的一局游戏时，score都会重置为游戏设定的初始值 0。​

工作流流程如下：​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

节点配置如下：​

​

|  |  |  |
| --- | --- | --- |
| **节点类型**​ | **说明**​ | **示例**​ |
| 开始节点​ | 工作流的起始节点，本示例保持默认参数即可。​ | ​  ​​图形用户界面, 文本, 应用程序  AI 生成的内容可能不正确。​  ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 循环节点​ | 用于指定猜谜游戏的次数，需要设置：​   * 循环类型：选择**指定循环次数**。​ * 循环次数：本示例设置为 **5**。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 大模型节点（出迷人）​ | 用于生成谜语，需要设置：​   * 系统提示词：设置出谜专家的人设，可以通过 AI 自动生成。​ * 输出：定义两个变量。​ * riddle：模型生成的谜语。​ * answer：谜语的谜底。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 问答节点​ | 用于向用户提出谜语问题，需要设置：​   * 输入：定义 riddle 参数，引用大模型节点（出迷人）生成的谜语问题。​ * 提问内容：本示例设置为{{riddle}}。​ * 回答类型：本示例选择直接回答。​ * 输出：定义 user\_answer 参数，用于接收用户的回答。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 大模型节点（谜语裁判）​ | 用于判断用户回答的正确性，需要设置：​   * 输入：定义四个变量。​ * riddle：引用大模型节点（出迷人）生成的谜语问题。​ * answer：引用大模型节点（出迷人）生成的谜底。​ * user\_answer：引用问答节点的输出参数。​ * score：引用应用变量score。​ * 系统提示词：设置谜语裁判的人设，可以通过 AI 自动生成。​ * 用户提示词：设置为谜语的谜面是{{riddle}}，正确答案是{{answer}}，用户的回答是{{user\_answer}}。​ * 输出：定义两个参数。​ * total\_score：用户猜谜游戏的分数。​ * conclusion：用户猜谜的结果。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 变量赋值节点​ | 用于存储用户在猜谜游戏中的分数，将大模型节点（谜语裁判）中 total\_score的值赋给变量 score。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 输出节点​ | 用于输出用户猜谜的结论，并告知游戏的得分，需要设置：​   * 输出变量：定义两个变量。​ * output：引用大模型节点（谜语裁判）的输出参数 conclusion。​ * score：引用应用变量score。​ * 输出内容：设置为{{output}}，你当前得分为：{{score}}。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 文本处理节点​ | 用户游戏结束，展示最终的游戏得分，需要设置：​   * 输入：将应用变量 score 的值赋给 String1。​ * 字符串拼接：设置为游戏结束！你本次总得分为{{String1}}，欢迎再次挑战哦~。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 结束节点​ | 选择返回文本模式，并设置输出变量和回答内容。​   * 输出变量：定义 output 参数，引用文本处理节点的输出参数 output。​ * 回答内容：设置为{{output}}。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |

​

常见问题​

**变量赋值节点能为系统变量赋值吗？**​

系统变量仅可读不可写，所以不能通过变量赋值节点为系统变量赋值。​

**变量赋值节点能读取变量值吗？**​

目前，除了开始节点、输入节点、知识库写入

变量聚合节点

工作流变量聚合节点能够将多路分支的输出变量整合为一个，方便下游节点统一配置。​

节点说明​

如果工作流中设计了多个分支，那么往往需要一个节点来汇总所有分支的输出结果，作为工作流的最终输出。如果有任意一个分支未运行，汇总输出的节点会从该分支读取到一个空值，从而导致工作流运行报错。在这种场景下，你可以使用变量聚合节点聚合多路分支的输出变量，变量聚合节点会读取多路分支中第一个不为空的值，供流程下游的节点使用和操作，不用额外处理未运行分支的输出结果，简化了数据流的管理。​

例如选择器或意图识别节点会将工作流拆分为多路分支，每次执行时，工作流会根据选择器条件或用户意图决定运行其中某一个分支，此时未执行的其他分支输出变量为空。你可以将多路分支都连接到变量聚合节点，并设置一个输出变量。无论哪条分支被执行，其结果都能通过这个变量被引用与访问，避免了下游节点对相同语义输出变量的重复定义。​

变量聚合节点支持分组聚合，每组变量会聚合为一个输出变量，分组组内的数据类型必须相同。​

配置变量聚合节点​

配置变量聚合节点时，需要选择聚合策略和聚合变量，并按需设置聚合分组。​

聚合策略​

通过指定策略对每个分组中的所有变量进行聚合处理，同一组内的变量实施相应聚合策略。​

目前聚合策略仅支持设置为“返回每个分组中第一个非空的值”，你可以拖动变量、调整变量位置。例如组内按顺序设置三个变量 output1、output2 和 output3，将其聚合为一个变量 Group1，如果 ouput1 不为空，则用 output1 的值为 Group1 赋值；如果 ouput1 为空，则取 ouput2 的值，依次类推。​

​

​​​

​

聚合变量​

在 Group 1 中选择需要聚合的变量，分组内变量的数据类型必须相同。​

* 变量聚合节点支持聚合多种数据类型，包括字符串（String）、数字（Integer、Number）、文件（File）对象（Object）以及数组（Array）等。​
* 每个分组只能聚合一种数据类型的变量。例如将多个 String 类型的变量聚合为一个 String 变量、将多个 Integer 类型的变量聚合为一个 Integer 变量。​

聚合分组​

默认只有一个分组 Group1，对应一个输出变量 Group1。Group1 分组中所有变量类型和输出的变量类型相同。如果需要输出多个变量，可以添加多个分组。例如每个分支都有两个输出变量 String 和 Integer，可以设置两个分组，分别用于聚合 String 和 Integer。​

​

​​​

​

输出​

变量聚合节点的输出变量固定为 Group1，是分组 Group1 聚合的返回结果，具体的值取决于聚合策略。如果有多个分组则根据分组数量递增为 Group2、Group3 等变量。输出变量的数据类型取决于对应分组聚合的变量数据类型。​

示例​

在家教类 AI 应用中，通过 IF 选择器判断用户输入的问题属于哪个学科分类，并将对应的问题流转给对应的分支处理，例如数学问题流转给数学分支的大模型回答问题。三个分支均汇总到变量聚合节点，汇总为两个参数，即学科类型（type）、答案（answer），并传递到结束节点回答用户问题。​

整体流程如下：​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

节点配置如下：​

​

|  |  |  |
| --- | --- | --- |
| 节点类型​ | 说明​ | 示例​ |
| 开始节点​ | 定义两个变量：​   * BOT\_USER\_INPUT：用户输入的问题，也就是学生的问题。​ * type：问题对应的学科。​ | ​  ​​图形用户界面, 文本, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 选择器节点​ | 判断用户输入的 type 属于哪个学科，将问题流转到对应分支处理。这里我们定义语文和数学两个分支，其他学科统一流转到另外一个分支。​  ​ | ​  ​​图形用户界面, 文本, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 大模型节点​ | 用于回答对应学科的问题，需要设置：​   * 输入：定义 query 变量，引用开始节点收集的用户问题。​ * 系统提示词：设置学科教师的人设，可以通过 AI 自动生成。​ * 用户提示词：设置为 {{query}}，直接引用 query 变量即可。​ * 输出：定义两个变量：​ * answer：模型生成的答案。​ * type：学科类型，例如语文节点固定值为“语文”。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 变量聚合节点​ | 聚合大模型节点的两个输出变量 answer 和 type。每个分组只能聚合一个变量，所以我们需要设置两个分组。​ | ​  ​​图形用户界面  AI 生成的内容可能不正确。​  ​ |
| 结束节点​ | 选择返回文本模式，并设置输出变量和回答内容，透传大模型节点生成的答案即可。​   * 输出变量：定义两个变量 answer 和 type，分别引用变量聚合节点的输出参数 Group1 和 Group2。​ * 回答内容：透传大模型节点生成的答案即可。​ | ​  ​​图形用户界面, 文本, 应用程序  AI 生成的内容可能不正确。​  ​ |

​输入节点

输入节点用于在工作流运行期间收集用户输入。​

节点说明​

在比较复杂的工作流场景中，某些节点的执行往往需要额外的用户输入。如果上游节点中没有获取到这些信息，你可以添加一个输入节点来主动收集信息。工作流执行到输入节点时会暂时中断，直到此节点收集到必要的用户输入。​

**说明**

* 仅扣子商店、豆包渠道或 OpenAPI 支持使用输入节点。​
* 如果需要通过输入节点上传文件，可以先将文件上传到第三方存储工具，获取一个公开可访问的 URL 地址，将此 URL 传入输入节点即可。目前仅试运行输入节点时可以直接上传本地文件。​

​

配置输入节点​

输入节点中，只需设置输入参数即可。​

​

|  |  |
| --- | --- |
| **配置**​ | **说明**​ |
| 变量名​ | 输入参数的名称。​ |
| 变量类型​ | 输入参数的数据类型，支持 String 等多种数据类型。​ |
| 描述​ | 参数的描述信息。​ |
| 是否必选​ | 参数是否必选。工作流在执行此节点时，收集到所有必选参数之后才会继续执行后续节点。​ |

​

你也可以直接导入 JSON 格式的数据结构，示例如下：​

​

​​文本

AI 生成的内容可能不正确。​

​

示例​

通过输入节点获取城市名称，并调用模型节点查询该地区的热门景点。​

​

​​图形用户界面

AI 生成的内容可能不正确。​

​

核心节点说明如下：​

​

|  |  |  |
| --- | --- | --- |
| **节点类型**​ | **配置说明**​ | **示例**​ |
| 输入节点​ | 添加一个必选的输入参数，名为 city，即用户想去的城市。​ | ​  ​​​  ​ |
| 大模型节点​ | 设置以下参数：​   * 输入：添加一个输入参数，引用输入节点的参数 city。​ * 系统提示词：智能体的人设，按需设置。​ * 用户提示词：需要模型回答的问题，此处引用输入节点的参数 city。​ * 输出：维持默认设置即可。​ | ​  ​​​  ​ |

​

在智能体中添加工作流之后，实际运行效果如下：

输出节点

输出节点用于在工作流执行过程中输出指定的消息内容。​

**注意**

消息节点已更名为输出节点。​

​

节点说明​

通常情况下，工作流会在执行完毕后通过结束节点输出最终的执行结果。当工作流处理流程较长、运行时间较久时，开发者可以在工作流中添加输出节点，临时输出一段消息，避免用户等待时间过长、放弃对话。例如提示用户任务正在执行中，建议用户耐心等待。​

输出节点支持流式和非流式两种模式，同时支持绑定卡片，展示丰富的交互效果，提升对话过程的用户体验。​

​

​​​

​

配置输出节点​

在输出节点中，工作流会在执行的同时输出一条中间消息，消息内容为指定的输出内容，其中可引用已定义的输出变量。输出节点支持流式输出，即打印机效果。​

**输出变量**​

输出变量用于定义输出内容中可引用的变量，支持添加多个变量。每个变量均需要设置参数名和参数值，其中参数值可指定为某个固定值，或引用上游节点的输出变量。​

输出变量也可以在输出节点绑定的卡片中使用。​

**输出内容**​

在工作流运行过程中，智能体将直接使用这里指定的内容回复对话。你可以使用{{变量名}}的方式引用**输出变量**中定义的变量。​

**流式输出**​

流式输出表示输出节点配置的输出内容会逐字地显示在对话中，类似于打字机的效果。流式输出适用于输出文本较长或需要工作流即时反馈的场景，呈现实时对话的交互效果，用户无需等待一大段文字一次性加载，可显著提高对话过程中的用户体验。​

输出节点默认采用非流式输出，待接收到全部消息内容后，再一次性输出全部消息内容。​

**注意**

* 输出节点只有在大模型节点之后才能开启流式响应。​
* 如果输出节点绑定了消息卡片，即使开启流式输出，输出的卡片也是非流式的，卡片会等待所有回复内容加载完毕后一次性展示在对话中。​
* 当在一个工作流中配置了多个输出节点并开启了流式返回时，输出节点的执行遵循工作流的执行顺序，先执行的输出节点优先输出消息。​

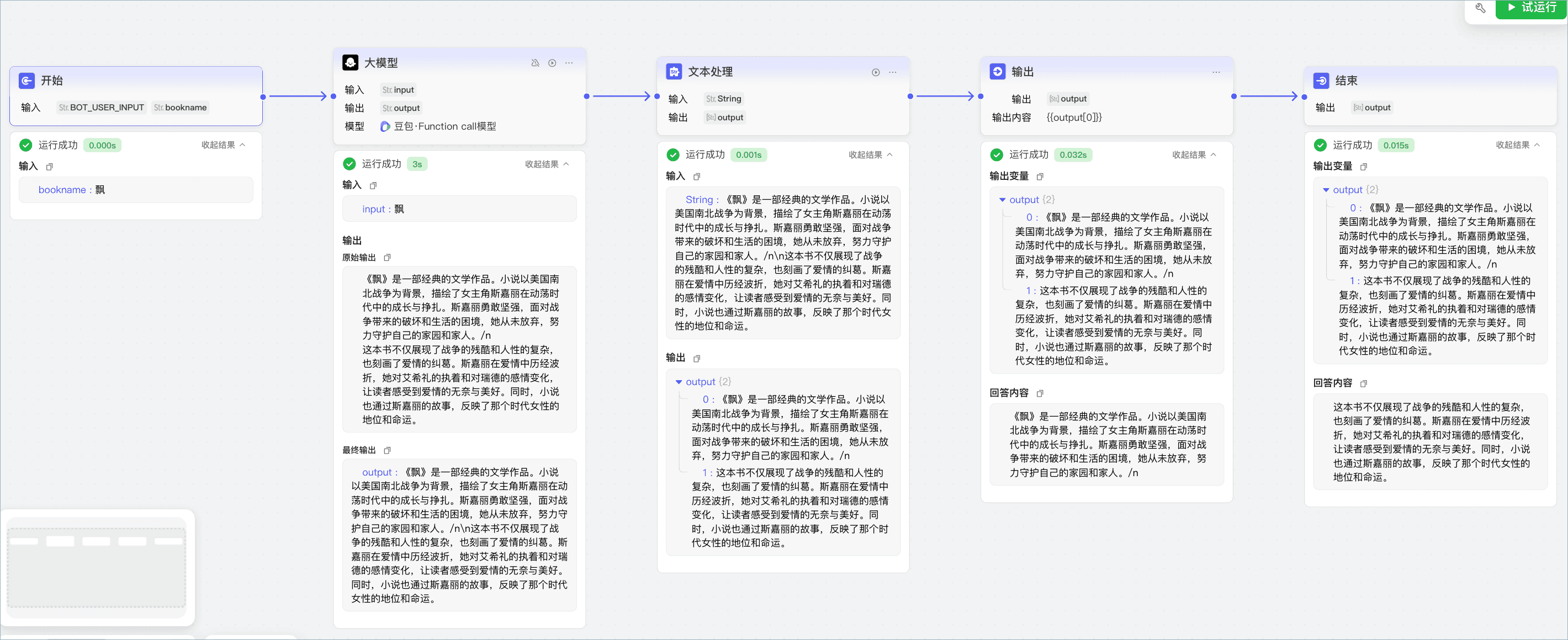
​

示例​

如果工作流的输出内容较长，可以使用输出节点拆分消息内容，一部分通过输出节点输出，一部分通过结束节点输出，同时开启流式输出。​

以好书解析智能体为例，通过大模型节点生成书籍的介绍，指定输出为两个段落，通过文本处理节点将两个段落拆开，一段通过输出节点输出，一段通过结束节点输出。​

​

​​​

​

核心节点配置说明如下：​

​

|  |  |  |
| --- | --- | --- |
| **节点类型**​ | **配置说明**​ | **示例**​ |
| 大模型节点​ | 引用开始节点的变量，获取用户需要查询的书籍名称。​  用户提示词中需要指定输出格式，例如指定输出为两个段落。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​  ​ |
| 文本处理节点​ | 使用字符串分隔模式，通过换行符将大模型节点的输出内容转为数组格式，每个段落是数组的一个元素。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 输出节点​ | 引用文本处理节点的输出，注意这里只引用数组的第一个元素，也就是书籍简介的第一个段落。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |
| 结束节点​ | 引用文本处理节点的输出，这里只引用第二个段落即可。​ | ​  ​​图形用户界面, 文本, 应用程序, 电子邮件  AI 生成的内容可能不正确。​  ​ |

​

在智能体中的体验效果如下：​

​知识库写入节点

知识库写入节点用于向指定的知识库中添加文档。​

节点说明​

在工作流执行过程中，用户不仅可以从已上传的知识库里检索信息，还可以通过知识库写入节点来主动更新知识库，上传新的文档内容。知识库写入节点是智能体或应用的用户上传知识库的唯一途径，它的本质是向一个已创建的知识库中上传文件，文件内容的切片策略沿用知识库的分段策略。​

**说明**

* 知识库写入节点仅支持向一个文本类型的知识库中上传文本类型的文档。​
* 每次运行知识库写入节点只能上传一个文件到知识库，但是你可以通过批处理或循环节点多次执行写入操作。​
* 知识库写入节点为异步节点，工作流执行时无需等待文档上传完成。​

​

输入与输出​

知识库写入节点的输入参数固定为 knowledge，参数格式为 File，表示用户需要上传到知识库中的文件。此文件可以固定为某个文件，由知识库的搭建者在搭建时上传；也可以引用上游节点的输出参数，在上游节点获取文件后上传。​

输出参数固定为以下参数：​

* documentId：知识库文档的 ID。格式为 String。​
* fileName：知识库文档的名称。格式为 String。​
* FileUrl：知识库文档的访问地址。格式为 String。​

数据写入设置​

文档解析策略​

文档解析策略用于定义系统如何解析文档中的文本、图片或表格内容。支持设置为：​

* **快速解析**：不提取文档中的图像、表格元素，适用于纯文本格式的内容，该模式下解析速度更快。​
* **精准解析**：提取文档中的图片、表格元素，支持选择提取的元素类型，默认包含图片元素、图片元素（OCR）、表格元素。该模式需要耗费更长的时间来解析并处理数据，如果工作流下游还有知识库检索节点，精准解析模式下上传的文档可能会未完成解析而无法被立即检索。​

分段策略​

每次上传文档时都可以设置新文档的分段策略，合理的切片策略可以提高检索效率、提升检索结果的准确性。支持选择自动分段或自定义分段，详细说明可参考​知识库概述。​

常见问题​

用户上传的文档和开发者上传的文档有什么区别？​

用户上传的文档和开发者上传的文档本身没有任何区别，都是知识库中的公开内容，可以被其他用户检索。但上传方式有区别，用户只能通过工作流的知识库写入节点上传文档，开发者则是在知识库管理页面或通过知识库 API 上传文档。​

知识库写入节点是异步节点吗？​

知识库写入节点为异步节点，工作流执行时无需等待文档上传完成。​

​知识库检索节点

知识库是智能体的私有知识合集，知识库检索节点可以基于用户输入查询指定的知识库，召回最匹配的信息，并将匹配结果以列表形式返回。​

输入与输出​

知识库节点的输入参数固定为 Query，表示用户希望在知识库中检索的关键信息，需要引用上游节点的输出参数。输入参数格式为 String，可以引用任何格式的数据​

输出参数固定为一个名为 outputList 的数组，其中包含多条召回结果，默认根据匹配度和相关性由高到低排序。​

添加知识库​

在知识库区域右上角单击 + 即可添加知识库到节点中，支持添加多个文档、表格和照片类型的知识库。​

知识库检索配置​

知识库的检索配置在很大程度上会影响召回的结果的准确性、相关性。​

搜索策略​

从知识库中获取知识的检索方式，不同的检索策略可以更有效地找到正确的信息，提高其生成的答案的准确性和可用性。支持的检索策略包括：​

* **混合**：结合全文检索和语义检索的优势，并对结果进行综合排序召回相关的内容片段。​
* **语义**：像人类一样去理解词与词、句与句之间的关系。推荐在需要理解语义关联度和跨语言查询的场景使用。例如下面两组句子，第一组的语义关系更强。​
* ​

"狼追小羊"和"豺狼追山羊"​

"狼追小羊"和"我爱吃炸猪排"​

​

* **全文**：基于关键词进行全文检索。推荐当查询内容包含以下场景时使用：​
* 特定名称或专有名词、术语等，例如比尔盖茨、 特斯拉 Model Y。​
* 缩写词，例如 SFT。​
* ID，例如 12s1w1s2 系列。​

表格 SQL 查询​

添加了表格知识库之后，默认开启表格 SQL 查询。开启后，查询表格知识库时，系统会自动将用户输入的自然语言转为 SQL 语句，根据 SQL 对表格知识库进行查询计算，SQL 执行结果与 RAG 召回的切片会一同传递给模型。​

**说明**

* 暂不支持跨表查询操作（JOIN）。​
* 知识库检索节点中添加了表格知识库时，才能使用表格 SQL 查询功能。​

​

最大召回数量​

从知识库中返回的最大段落数量。数值越大，返回的条目越多。默认召回 5 条检索结果。​

最小匹配度​

匹配度指的是系统检索到的上下文与被视为 ground-truth 的标注答案对齐的程度。设置最小匹配度参数后，系统会根据设置的匹配度选取段落，低于指定匹配度的内容不会被召回。默认的最小匹配度为 0.5。​

查询改写​

在多轮对话中，用户的 Query 和对话的上下文息息相关，仅凭借用户最新一条提问可能无法正确识别用户的真实检索意图。查询改写是指根据对话历史对用户输入的 Query 进行优化或重构，从而更准确地捕捉真实的用户意图，提升信息检索的效率。知识库检索节点默认开启查询改写。​

例如用户对话的上下文为：​

* 问题1：知识库检索节点可以用来做什么？​
* 回复1：知识库检索节点可以基于用户输入查询指定的知识库，召回最匹配的信息，并将匹配结果以列表形式返回。​
* 问题2：怎么用？​

对于问题2，不参考上下文的情况下无法判断用户的真实意图。开启查询改写后，问题2会被改写为“知识库检索节点怎么用？”​

**说明**

开启查询改写后，知识库节点的处理流程会加一轮模型处理，由大模型来改写用户输入。所以查看消息链路时，可能会看到一个名为 WorkflowLLMCall 的处理环节。​

​

​​​

​

​

结果重排​

结果重排是指根据相关性或质量对检索到的文档切片进行重新排序，以提高生成答案的准确性和相关性，适用于追求回答高精度和高相关性的场景，例如智能客服、专业技术答疑等场景。未开启结果重排时，节点输出的是向量检索的结果，根据匹配度从大到小排序；开启结果重排后，系统会将 Embedding 的召回结果交由 Rerank 模型进行质量和相关性判断，对结果重新排序，将与输入问题最相关的文档排在前面。​

假设用户查询“如何制作意大利面？” ，首先会从知识库中检索得到以下几个文档片段，其中先按照ABCD先后顺序排列：​

* 切片 A：介绍意大利面的历史。​
* 切片 B：讨论了不同种类的意大利面和它们的搭配。​
* 切片 C：详细描述了制作意大利面的步骤，包括所需材料和烹饪技巧。​
* 切片 D：提供了一些意大利面食谱。​

在结果重排的过程中，知识库检索节点会分析用户的真实意图，对切片重新排序，使得最相关的内容排在前面。最终的排序可能变为：​

* 切片 C：详细描述了制作意大利面的步骤，包括所需材料和烹饪技巧。​
* 切片 D：提供了一些意大利面食谱。​
* 切片 B：讨论了不同种类的意大利面和它们的搭配。​
* 切片 A：介绍意大利面的历史。​

仅查看个人文档​

指定查询范围是否仅限于用户的个人文档。默认为关闭状态。​

* 开启：在知识库检索节点添加的全部知识库中，用户只能搜索自己通过知识库写入节点写入的文档，不能在开发者和其他人上传的文档中检索内容。​
* 关闭：检索范围为知识库节点中添加的全部知识库。​

常见问题​

知识库检索节点和大模型节点添加知识库有什么区别？​

前者是人工指定输入的检索内容，百分百调用​

后者是大模型根据query决定是否调用知识库、以及使用什么内容去检索知识库

​批处理节点

批处理节点用于批量执行部分操作。​

节点说明​

工作流执行时，每个节点按顺序运行一次，如果需要一次性运行多次，例如一次性查询多个城市的天气，可以使用批处理节点提高工作流效率。批处理节点会批量运行除批处理和循环节点以外的任何节点，直到达到次数限制或者列表的最大长度。​

批处理节点适用于大量数据并行处理的场景。相对于添加多个相同的节点执行任务，批处理节点的效率更高。例如批量生成根据故事生成绘本的场景，需要根据每个故事片段生成对应的图片，你可以使用批处理节点执行文生图的工作流，一次性生成多张图片。​

**说明**

* 每一批任务默认并行运行，如需串行执行，可以将并行数量设置为 1，或者使用循环节点。​
* 批处理节点中不支持添加另一个批处理节点或循环节点。​
* 在工作流中，同一时刻只能运行一个流式插件或消息节点。请勿通过批处理节点批量并行运行流式插件或消息节点，否则可能导致这些节点无法按预期正常输出消息。​

​

配置批处理节点​

批处理节点的处理对象为输入参数引用的数组结构，例如上游某个节点输出的多条数据结果。批处理节点会遍历数组中的每个元素，对所有元素同时执行一次性处理，直到所有元素处理完成，或达到指定的次数上限。​

批处理设置​

为避免并行运行数量过大导致性能风险，批处理节点是分批运行的，默认每批执行 10 次，最多运行 200 次。通过批处理设置，你可以调整每一批运行的最大次数，和多批处理的总次数上限。具体参数包括：​

* **并行运行数量上限**：每一批运行的最大次数。默认并行运行 10 次，最大可指定为 200 次。此参数可指定为某个固定值，例如 5；也支持引用上游节点数值类型的的输出参数。​
* **说明**

并行运行数量参数的引用值如果大于 200，则设置为 200，如果小于 1，则设置为 1。​

​

* **批处理次数上限**：批处理执行总次数达到此上限时，此节点终止运行。此节点的执行逻辑是处理数组中的元素，当批处理次数达到上限时，即使节点未遍历数组中的每个元素，也会停止运行。​

例如批处理的处理对象是一个长度为 50 的数组，并行运行数量上限设置为 10，批处理次数上限为 30。那么批处理节点会执行 3 批，每一批同时运行 10 次，共计执行 30 次后结束运行。​

输入​

输入参数指批处理体中要使用的参数。批处理节点通常需要批量处理数组中的所有元素，那么可以在批处理节点定义输入参数，引用前置节点 Array 类型的输出，批处理体中就可以引用批处理节点输入参数中的某个元素或索引，否则只能引用上游节点 Array 类型参数的整个数组结构。​

批处理节点应最少设置一个输入参数，且只能引用上游节点的 Array 类型。​

输出​

输出参数是批处理完成后输出的内容，仅支持引用批处理体中节点的输出变量，格式为数组。​

批处理体​

批处理体和循环体类似，每个批处理节点都会生成一个批处理体画布。开发者需要选中批处理体，通过拖拉拽的方式将待批处理的节点添加到批处理体中，并使用连接线依次将批处理体和各个节点顺序连接起来。你还需要在批处理节点和上下游节点之间添加连线，但无需调整批处理体和批处理节点之间的连线。​

**说明**

不支持将批处理体外部的节点拖动至批处理体内，批处理体中的节点也不可移动到批处理体之外。​

​

示例​

例如通过视觉理解模型批量分析图片内容。添加一个批处理节点，其中只有一个大模型节点，用于批量分析多张图片。开始节点接收图片数组后传递给批处理节点，视觉模型会处理图片数组中每个图片元素。工作流的编排方式如下：​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

工作流核心节点如下：​

​

|  |  |  |
| --- | --- | --- |
| 节点类型​ | 说明​ | 示例​ |
| 开始节点​ | 定义两个输入参数：​   * image：图片数组格式，支持上传多张图片。​ * query：String 格式，后续节点将作为模型的提示词。​ | ​  ​​​  ​ |
| 批处理节点​ | 设置输入参数 image，引用开始节点的 image 参数，便于批处理体引用。​ | ​  ​​​  ​ |
| 批处理体​ | 添加一个大模型节点，其设置如下：​   * 模型：选择豆包视觉理解模型，它支持读取并分析图片内容。​ * 输入：​ * image 参数：引用批处理节点的 image 参数元素（item in input），表示每次执行一个数组元素，即图片。​ * query 参数：开始节点的 query 参数。​ * 用户提示词：直接引用两个输入参数即可，即 {{query}}{{image}}。​ | ​  ​​​  ​ |

​

常见问题​

如何设置批处理次数​

批处理节点会按照设定的并行数量分批运行批处理体，直到处理的数组中所有元素处理完成，或达到指定的次数上限。你可以调整待处理的数组长度，或者通过**批处理次数上限**来限制批处理节点的最大运行次数，后者最大值为 200。​

批处理节点和循环节点有什么区别​

* **应用场景**：批处理节点通常用于批量执行一组数据或任务，目的是对整个数据集进行一次性并行处理，而不是逐个处理，运行效率更高。循环节点用于循环执行某个操作或任务，直到满足指定条件为止。循环节点的核心是“次序性”，每次循环处理一个数据子项或执行一次子任务，总耗时可能较长。​
* **运行方式**：批处理节点仅支持处理数组参数，运行次数默认为数组长度，处理完毕后自动结束运行；循环节点支持多种方式结束循环，例如设置固定的循环次数、数组长度、指定条件等等。

SQL 自定义节点

SQL 自定义节点用于对指定数据库进行常见的 SQL 操作。​

**注意**

数据库节点已更名为 SQL 自定义节点。​

​

节点说明​

SQL 自定义节点可以连接智能体或应用中指定的数据库，对数据库进行新增、查询、编辑、删除等常见操作，实现动态的数据管理。SQL 自定义节点需要指定待操作的数据库表和对应的 SQL 语句，支持通过自然语言智能生成 SQL 语句。​

**注意**

* 开发调试阶段不会改动数据库原表，在调试区查看到的是测试数据，和数据库中的真实数据是隔离的。​
* 在工作流中调试 SQL 自定义节点时，不能使用库数据表中的真实数据，需要先插入数据后再进行查、删、改等操作的测试。​

​

扣子工作流还支持通过图形化方式进行数据的增、删、改、查操作，当你不熟悉 SQL 语句时，可使用以下节点操作数据库。​

* ​新增数据节点​
* ​查询数据节点​
* ​更新数据节点​
* ​删除数据节点​

配置 SQL 自定义节点​

输入​

节点的输入参数，即 SQL 语句中需要使用的参数，可以设置为固定值，也可以引用上游节点的输出参数。​

数据表​

在数据表区域，你需要根据页面提示添加需要操作的数据表，每个 SQL 自定义节点仅支持操作一张数据表。​

在调试期间，SQL 自定义节点显示和使用的是数据表的**测试数据**，而非数据库中的真实线上数据。单击数据表或单击查看数据，弹出数据表的详情页，可查看此数据表的测试数据。你可以手动添加或修改测试数据，也可以试运行 SQL 自定义节点，通过 SQL 语句插入或修改数据。​

​

​​​

​

SQL​

在 SQL 区域输入需要对数据表执行的 SQL 操作，兼容 SQL92 的常用语法。SQL 语句中可以引用SQL 自定义节点输入参数中定义的变量，引用格式为 {{变量名}}。​

**注意**

* 每个 SQL 自定义节点中仅支持添加一条 SQL 语句。​
* 不支持 Select\* 语法、多表 Join 操作。​
* 每次执行SQL 自定义节点，最多返回 100 行数据。​

​

你可以自行编写 SQL 语句，也可以根据页面提示由 AI 帮你生成一段 SQL 语句。单击**自动生成**，并在弹出的页面中使用自然语言描述要执行的操作，单击**自动生成**生成 SQL 语句，确认无误后单击**使用Use**。​

​

​​​

​

输出​

SQL 自定义节点的输出参数是 SQL 执行后的输出内容，固定为以下两项：​

* outputList：SQL 执行后数据表中的字段和数据。你可以按需新增子项，注意变量名需与 SQL 中定义的字段名一致、数据类型需要和数据表中定义的数据类型一致。​
* rowNum：返回的行数或者受影响的行数。​

常见问题​

为什么调试 SQL 自定义节点的结果和线上效果不一致？​

调试工作流时，SQL 自定义节点使用的是临时的测试数据，线上执行工作流时使用的是数据库中的真实数据，所以统计查询的结果会有差异。​

新增数据节点

新增数据节点用于向指定的数据库插入数据，对应 SQL 中的 INSERT 操作。​

节点说明​

在工作流中，你可以添加**新增数据节点**，以图形化方式配置数据库内数据的插入操作。此节点中需要指定待操作的数据库表和待插入的字段名称及字段值，每次执行此节点时自动插入一行数据。整个过程无需编写 SQL 语句，操作简单直观。​

相比 ​SQL 自定义节点，新增数据节点中的操作极大降低了操作门槛，对于不熟悉 SQL 的用户，也能高效、轻松地操作数据库。​

**注意**

* 开发调试阶段不会改动数据库原表，在调试区查看到的是测试数据，和数据库中的真实数据是隔离的。​
* 在工作流中调试新增数据节点时，不能使用库数据表中的真实数据，需要先插入测试数据后再进行查、删、改等操作的测试。​

​

配置数据节点​

数据表​

在数据表区域，你需要根据页面提示添加需要操作的数据表。每个新增数据节点仅支持操作一张数据表。​

在调试期间，新增数据节点中显示和使用的是数据表的**测试数据**，而非数据库中的真实线上数据。单击数据表或单击查看数据，弹出数据表的详情页，可查看此数据表的测试数据。​

​

​​​

​

选择并设置字段​

在**选择并设置字段**区域，系统会根据你所选数据表的结构，罗列出该表所包含的所有字段。你可以根据业务需求选择目标字段，为其配置字段值，支持设置为固定值，或引用上游节点的输出参数。​

配置完成后，执行运行，数据表中将新增对应的测试数据。​

输出​

新增数据节点中的输出参数是执行数据新增操作后的输出内容，固定为以下两项：​

* outputList：执行数据新增操作后数据表中的字段和数据。你可以按需新增子项，注意变量名需与定义的字段名一致、数据类型需要和数据表中定义的数据类型一致。​
* rowNum：成功新增的数据行数。​

示例​

例如使用**新增数据节点**，向用户信息表 userinfo 中插入一条数据，companyname 为 co，username 为 xiaoming，tel 为 123。

查询数据节点

查询数据节点用于在指定的数据库中查询数据，对应 SQL 中的 SELECT 操作。​

节点说明​

在工作流中，你可以添加**查询数据节点**，以图形化方式配置数据库内数据的查询操作。此节点中需要指定待操作的数据库表，如果你有更精细化的查询需求，可以添加查询字段、查询条件、排序方式和查询上限等配置。整个过程无需编写 SQL 语句，操作简单直观。​

相比 ​SQL 自定义节点，查询数据节点中的操作极大降低了操作门槛，对于不熟悉 SQL 的用户，也能高效、轻松地操作数据库。​

**注意**

* 开发调试阶段不会改动数据库原表，在调试区查看到的是测试数据，和数据库中的真实数据是隔离的。​
* 在工作流中调试查询数据节点时，不能使用库数据表中的真实数据，需要先插入测试数据后再进行查询操作的测试。​

​

配置数据节点​

数据表​

在数据表区域，你需要根据页面提示添加需要操作的数据表。每个查询数据节点仅支持操作一张数据表。​

在调试期间，查询数据节点显示和使用的是数据表的**测试数据**，而非数据库中的真实线上数据。单击数据表或单击查看数据，弹出数据表的详情页，可查看此数据表的测试数据。​

​

​​​

​

查询字段​

在**查询字段**区域，系统会根据你所选数据表的结构，罗列出该表所包含的所有字段。你可以根据业务需求，从中选择想要查询的目标字段，支持添加多个查询字段。如果不指定查询字段，系统将返回表中的所有字段。​

查询条件​

在**查询条件**区域，你可以设置对应的筛选条件，以此来指定待查询数据的范围。​

* 支持等于（=）、不等于（!= 或 <>）、模糊匹配（LIKE）、模糊不匹配（NOT LIKE）、包含（IN）、不包含（NOT IN）、为空（IS NULL）、不为空（IS NOT NULL）等运算符。​
* 模糊查询时，后端会自动在字符串前后添加通配符 %，例如 %值%，无需手动配置通配符。​
* 支持添加多个条件，多个条件之间的逻辑关系可设置为且、或关系。​

排序方式​

在**排序方式**区域，你可以设置查询结果的排序方式。系统会根据查询字段的数据类型提供对应的排序选项，详细说明如下：​

* 如果查询字段是字符串类型，支持按照 A→Z 或 Z→A 进行排序。​
* 如果查询字段是数值类型，支持按照 0→9 或 9→0 进行排序。​
* 如果查询字段是时间类型，支持按照 0→9 或 9→0 进行排序。​
* 如果查询字段是布尔类型，支持按照 0→1 或者 1→0 进行排序。​
* 其中，0 代表 fasle，1 代表 true。 ​

如果添加了多个查询字段，你还可以拖动字段来调整字段排序的优先级。​

查询上限​

查询上限用于限制返回结果行数，相当于 SQL 语句中的 LIMIT 限制，默认值为 100，最大可设置为 1000。​

输出​

查询数据节点中的输出参数是执行数据查询操作后的输出内容，固定为以下两项：​

* outputList：执行数据查询操作后数据表中的字段和数据。你可以按需新增子项，注意变量名需与定义的字段名一致、数据类型需要和数据表中定义的数据类型一致。​
* rowNum：返回的行数或者受影响的行数。​

示例​

例如使用**查询数据节点**，在用户信息表 userinfo 中查询 companyname 字段值为 co 的数据，并返回对应的 username 和 tel 字段值。查询结果按照 username 字段值进行 A 到 Z 字母序生序排序。​

​

​​​

​

​

更新数据节点

更新数据节点用于更新指定的数据库中数据，对应 SQL 中的 UPDATE 操作。​

节点说明​

在工作流中，你可以添加**更新数据节点**，以图形化方式配置数据库内数据的更新操作。此节点中需要指定待操作的数据库表、更新条件、待更新的字段名称及字段值，每次执行此节点时将更新所有符合条件的数据。整个过程无需编写 SQL 语句，操作简单直观。​

相比 ​SQL 自定义节点，更新数据节点中的操作极大降低了操作门槛，对于不熟悉 SQL 的用户，也能高效、轻松地操作数据库。​

**注意**

* 开发调试阶段不会改动数据库原表，在调试区查看到的是测试数据，和数据库中的真实数据是隔离的。​
* 在工作流中调试查询数据节点时，不能使用库数据表中的真实数据，需要先插入测试数据后再进行更新操作的测试。​

​

配置更新数据节点​

数据表​

在**数据表**区域，你需要根据页面提示添加待更新数据的数据表。每个更新数据节点仅支持操作一张数据表。​

在调试期间，更新数据节点中显示和使用的是数据表的**测试数据**，而非数据库中的真实线上数据。单击数据表或单击查看数据，弹出数据表的详情页，可查看此数据表的测试数据。​

​

​​​

​

更新条件​

在**更新条件**区域，你可以设置对应的筛选条件，以此来指定待更新数据的范围。​

* 支持等于（=）、不等于（!= 或 <>）、模糊匹配（LIKE）、模糊不匹配（NOT LIKE）、包含（IN）、不包含（NOT IN）、为空（IS NULL）、不为空（IS NOT NULL）等运算符。​
* 模糊查询时，后端会自动在字符串前后添加通配符 %，例如 %值%，无需手动配置通配符。​
* 支持添加多个条件，多个条件之间的逻辑关系可设置为且、或关系。​
* 至少添加一个更新条件。​

选择并设置字段​

在**选择并设置字段**区域，系统会根据你所选数据表的结构，罗列出该表所包含的所有字段。你可以根据业务需求，选择待更新的字段，为其更新字段值。支持设置为固定值，或引用上游节点的输出参数。​

输出​

更新数据节点中的输出参数是执行数据更新操作后的输出内容，固定为以下两项：​

* outputList：执行数据更新操作后数据表中的字段和数据。你可以按需新增子项，注意变量名需与定义的字段名一致、数据类型需要和数据表中定义的数据类型一致。​
* rowNum：返回的行数或者受影响的行数。​

示例​

例如使用**更新数据节点**，在用户信息表 userinfo 中更新用户电话信息，即查询 username 字段等于 xiaoming 的数据，并将其对应的 tel 字段值更新为新值。​

​删除数据节点

删除数据节点用于删除指定的数据库中数据，对应 SQL 中的 DELETE 操作。​

节点说明​

在工作流中，你可以添加**删除数据节点**，以图形化方式配置数据库内数据的删除操作。此节点中需要指定待操作的数据库表和删除条件，每次执行此节点时将删除所有符合条件的数据。整个过程无需编写 SQL 语句，操作简单直观。​

相比 ​SQL 自定义节点，删除数据节点中的操作极大降低了操作门槛，对于不熟悉 SQL 的用户，也能高效、轻松地操作数据库。​

**注意**

* 开发调试阶段不会改动数据库原表，在调试区查看到的是测试数据，和数据库中的真实数据是隔离的。​
* 在工作流中调试查询数据节点时，不能使用库数据表中的真实数据，需要先插入测试数据后再进行删除操作的测试。​

​

配置删除数据节点​

数据表​

在数据表区域，你需要根据页面提示添加需要操作的数据表。每个删除数据节点仅支持操作一张数据表。​

在调试期间，删除数据节点中显示和使用的是数据表的**测试数据**，而非数据库中的真实线上数据。单击数据表或单击**查看数据**，可查看此数据表的测试数据。​

​

​​​

​

删除条件​

在**删除条件**区域，你可以设置对应的筛选条件，以此来指定待删除的数据，只有符合删除条件的数据才会被删除。​

* 支持等于（=）、不等于（!= 或 <>）、模糊匹配（LIKE）、模糊不匹配（NOT LIKE）、包含（IN）、不包含（NOT IN）、为空（IS NULL）、不为空（IS NOT NULL）等运算符。​
* 模糊查询时，后端会自动在字符串前后添加通配符 %，例如 %值%，无需手动配置通配符。​
* 支持添加多个条件，多个条件之间的逻辑关系可设置为且、或关系。​
* 至少添加一个删除条件。​

输出​

删除数据节点中的输出参数是执行数据删除操作后的输出内容，固定为以下两项：​

* outputList：执行数据删除操作后数据表中的字段和数据。你可以按需新增子项，注意变量名需与定义的字段名一致、数据类型需要和数据表中定义的数据类型一致。​
* rowNum：返回的行数或者受影响的行数。​

示例​

例如使用**删除数据节点**，删除用户信息表 userinfo 中所有 companyname 字段不等于 co 的数据。

问答节点

你可以在工作流中设计问答节点，主动收集用户的信息、获取用户意图。​

节点说明​

工作流中的某些节点依赖用户的信息输入或明确意图，问答节点会以自然语言问题或选项的方式收集指定的信息，让对话更加顺畅。如果对话中触发了包含问答节点的工作流，智能体会以指定问题向用户提问，并等待用户回答。​

**说明**

* 豆包渠道暂不支持问答节点。即使在豆包渠道中触发了问答节点，此节点也不会按照预期运行。​
* 包含问答节点的工作流建议使用同步模式。异步运行模式下，问答节点不会被触发。​
* Chat SDK 暂不支持问答节点的卡片效果，用户需要手动复制问答节点的选项，并发送给智能体。​

​

问答节点支持以两种方式收集用户的信息或意图：​

​

|  |  |
| --- | --- |
| **收集方式**​ | **说明**​ |
| 直接回答​ | 节点中指定一个开放式问题，用户直接以自然语言回复问题，智能体会提取用户的整段回复，或提取回复中的关键字段。如果用户的响应和智能体预期提取的信息不匹配，例如缺少必选的字段，或字段数据类型不一致，智能体会主动再次询问，直到获取到关键字段。​  例如在查询天气场景下，询问日期和城市，并从用户回复中提取位置字段。如果用户提供的信息不足，智能体将继续询问。​  ​  ​​​  ​ |
| 选项回答​ | 问答节点预置固定选项，用户以固定选项回复问题，通常用于聊天式的智能体中，推进对话进度、增强互动性。你可以将用户可以执行的操作设置为选项，帮助用户在指定范围内快速回复，也可以将常见的意图作为选项，作为用户输入的提示信息。每个选项通常对应不同的工作流分支处理，用户在选项之外的回复也需要有分支处理，例如可以引导用户再次选择。​  例如在互动类游戏场景下，收集用户在每个环节的路径选择，推进剧情进展。​  **说明**  在飞书中，选项会展示为普通文字，用户应复制选项文字进行回答。​  ​  ​  ​​​  ​ |

​

配置问答节点​

1.

在工作流编辑页面左侧工具栏中，选择问答节点。​

2.

为问答节点添加以下配置。​

* ​

|  |  |
| --- | --- |
| **配置项**​ | **说明**​ |
| 模型​ | 选择执行此节点的模型，支持设置模型在此节点中的生成多样性等参数配置，使模型效果更符合你的预期。​ |
| 输入​ | 设置需要添加到问题中的参数，参数值可以引用前置节点的输出参数，或指定内容。​ |
| 回答内容​ | 设置智能体在对话中向用户发起的问题内容。​ |
| 回答类型​ | 设置用户回答问题的方式，支持设置为：​   * + 直接回答：允许用户自定义回复内容。此时应设置**输出**参数。​   + 选项回答：设置预置选项，引导用户从指定范围内快速回复。此时还应设置**选项内容**，选项在对话中展示为按钮形式，用户可以直接点击按钮或回复对应的选项编号。​ |
| 输出​ | 指定节点的输出设置。节点默认将用户回复作为 USER\_RESPONSE 变量。​  你也可以开启从回复中提取字段，并设置待提取的关键字段。如果用户回复中未包含必选的字段信息，智能体将一直追问，直至成功采集信息。​  ​  ​​​  ​ |

* ​

3.

将问答节点与其他节点连接，形成完整的调用链路。​

* 问答类型为选项回答时，应为每个分类都设置后续的处理节点。例如在客服类智能体中，为用户提供“售前咨询”、“售后问题”两个选项，分别流转至对应的知识库。​
* 推荐为问答节点设置兜底策略，若意图未匹配到此处定义的任何分类，则流转到兜底策略处理。例如在客服类智能体中，兜底策略为转人工处理，如果用户回复与任一选项都不相关的内容时，流转到消息节点，指引用户如何联系人工客服。​
* ​

​​​

​

示例​

直接问答​

以一个需求咨询智能体的留资工作流为例，该节点用于收集用户的个人信息，包括企业名称、联系人、联系方式，并将其记录在数据库中，用于后续销售人员联系客户沟通具体需求。工作流的关键节点如下：​

* 问答节点：通过问答节点收集用户信息，回答方式指定为**直接回答**。开启**从回复中提取字段**，智能体会自动提取关键字段，各个信息均为必选项，如果用户回复信息不完整，智能体会继续追问。​
* 数据库节点：用于保存用户回复中的个人信息，输入参数的值引用问答节点的输出参数，通过 SQL 写入一行新数据，记录用户的个人信息。​

​

​​​

​

问答节点配置：​

​

​​​

​

数据库节点配置：​

​

​​​

​

结束节点配置：​

​

​​​

​

​

选项回答​

以个人助手智能体为例，助手提供查看热门新闻、查看热门视频的技能，这两项技能均以工作流官方插件节点方式实现。​

工作流的关键节点为问答节点，此节点通过问答方式引导选择意图，即用户期望执行的操作。此处提供两个操作，即“查看热门新闻”和“查看热门视频”。这两个技能均流转至对应的扣子官方插件处理。如果用户在问答节点回复了无关内容，则流转至文本节点，并结束工作流。​

​

​​​

​

问答节点配置：​

​

​​图形用户界面, 文本, 应用程序

AI 生成的内容可能不正确。​

​

插件节点配置：​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

​

文本处理节点配置：​

​

​

结束节点配置：​

文本处理节点

文本处理节点用于将多种类型的输入数据进行字符串处理，适用于内容二次总结、文本拼接、文本转义等场景，例如对多轮对话中的关键字拼接为 prompt 后进行文生图。​

文本处理节点的处理方式支持字符串拼接和字符串分隔。​

​

|  |  |
| --- | --- |
| **处理方式**​ | **说明**​ |
| 字符串拼接​ | 将输入中指定的内容根据一定顺序拼接为一个字符串，用于组合前置节点的关键信息，作为后置节点的输入。例如在分段总结场景中，把前置节点的总结结果拼接为完整段落，再输入给后置节点处理。​  支持引用输入参数中的变量，引用格式包括 {{变量名}}、{{变量名.子变量名}}、{{变量名[数组索引]}}。直接引用数组类型的参数时，默认通过逗号连接数组中的每个元素，你也可以引用数组中指定位置的元素。​ |
| 字符串分隔​ | 将输入中的内容用指定分隔符拆分为字符串数组，便于后续节点对不同内容进行差异化处理。​  你需要指定分隔符来拆分内容，支持自定义分隔符，例如设置多字符的分隔符，例如...。​ |

​

字符串拼接示例：​

* 配置示例：​

​

​​​

​

* 输入示例：白色​
* 输出示例："白色在英语中被称为White，在日语中被称为白い。"​

字符串分隔示例：​

* 配置示例：​
* ​

​​​

​

* 输入示例：小明，18岁，A市人，喜欢看书、看电影，偶尔健身。​
* 输出示例：[ "小明", "18岁", "A市人", "喜欢看书、看电影", "偶尔健身" ] ​

长期记忆节点

长期记忆节点用于在工作流中召回长期记忆中储存的用户的个性化信息。​

节点说明​

在用户喜好推荐等个性化的场景中，通常需要基于用户画像、关键记忆点等个人数据进行推荐、筛选，让智能体效果更加贴合用户需求、提高用户体验。通常情况下，我们可以通过多轮会话的上下文来收集这些信息，但是基于上下文轮数限制，个性化信息无法长期记忆和保存，此时可以开启长期记忆功能，记录并调用用户的个性化信息。在工作流中，也可以通过长期记忆节点调用智能体的长期记忆，查询智能体已记录的用户喜好、用户画像等信息，让工作流的效果更加个性化。​

**说明**

* 长期记忆节点需要召回智能体存储的长期记忆数据，所以试运行长期记忆节点或包含长期记忆节点的工作流时，需要指定一个已开启 **Long-term memory** 功能的智能体。​
* 如果智能体绑定了包含长期记忆节点的工作流，则智能体需要开启长期记忆功能，否则工作流执行会报错 This Bot does not have LTM enabled。同时建议关闭 **Supports calling in the Prompt**，否则在对话中容易同时触发长期记忆召回和工作流执行，影响对话效果。​

​

配置长期记忆节点​

长期记忆节点的配置说明如下：​

​

|  |  |
| --- | --- |
| **配置**​ | **说明**​ |
| 输入参数​ | 输入参数固定为 Query，表示需要从长期记忆中匹配的关键信息，例如查询用户的喜好、生日、男友名字等信息。以新闻搜索的工作流为例，Query 可以固定为“喜欢什么类型的新闻”，基于用户喜好检索并推荐新闻。​  Query 可指定为 **Reference** 或 **Input**：​   * 引用：引用上游节点的输出参数。​ * 输入：指定为某个字符串。​ |
| 输出参数​ | 输出参数固定为 outputList，格式为 Array<Object>，智能体会列举和 Query 相关的长期记忆。如果下游节点引用了这个参数，智能体会总结长期记忆中的内容，并将总结内容作为下游节点的输入。​ |

​

长期记忆节点配置示例：​

​

​​​

​

下游节点引用长期记忆节点的输出参数：​

​

​​​

​

示例​

例如对于查看热点新闻的工作流，可以召回用户的长期记忆，根据用户喜好来筛选出其可能感兴趣的内容。​

工作流主要设计如下：​

​

​​图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。​

​

各节点说明如下：​

​

|  |  |  |
| --- | --- | --- |
| **节点**​ | **说明**​ | **配置示例**​ |
| 开始节点​ | 维持默认参数设置即可，无需任何必选的输入参数。​ | ​  ​​图形用户界面, 文本, 应用程序, 聊天或短信  AI 生成的内容可能不正确。​  ​ |
| 长期记忆节点​ | 长期记忆节点的输入参数设置为 **Input**，Query 固定为“喜欢什么类型的新闻”。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 插件节点​ | 插件节点中使用新闻插件 getNews，输入参数中引用长期记忆节点的输出参数，基于用户喜好检索并推荐新闻。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 结束节点​ | 结束节点引用插件节点的输出参数，返回变量由智能体生成回答。​ | ​  ​​​  ​ |

​

​

智能体对话效果如下，你也可以将工作流绑定卡片，定制个性化的展示效果。​

​

​​

图像生成节点

图像生成节点用于生成图片。通过图像生成节点，你可以将一段文字转为图片，也可以根据参考图生成图片。​

节点说明​

在 AI 智能处理场景下，我们通常使用文生图的方式调用大模型节点来生成图像，如果模型生成效果不符合预期，则通过一系列关键词和参数调整细节。但对话方式处理图像通常存在以下问题：​

* 过程繁琐、保密性低，不适合批量的图像生成和处理。​
* 普通开发者也可能不熟悉图像处理的模型参数及其效果，反复调试非常耗时。​

你可以使用工作流的图像生成节点来生成图像，也可以使用循环节点嵌套图像生成节点来批量生成图像，提高文生图的效率和效果。​

**说明**

前往[图像生成器模板](https://www.coze.cn/template/project/7442540084944994344)可直接体验图像模型的生图效果。​

​

​​​

​

​

配置图像生成节点​

模型设置​

选择用于生成图片的模型，并设置生成的图像比例和质量。​

* **模型**：扣子提供多种多预训练的模型供你挑选，每个模型出图的风格不同，例如专用于动漫场景的动漫模型、面部处理更加细致自然的人像模型等，你可以选择不同的模型后，分别试运行体验模型效果。​
* **比例**：生成的图像比例，默认为 1024 × 1024，支持的宽高范围为 [512,1536]​
* **生成质量**：数值越大画面越精细，生成时间越久。默认为 25，范围为 [1,40]。​

参考图​

生成图像的参考图，支持设置多个参考图。​

* **参考模式**：指定模型如何参考指定图片，支持画面的空间关系、人物姿势等多个参考模式，例如你可以根据页面提示选择一个合适的模式。​
* **参考图**：参考图的模型参考图可以使用开发者上传的图片，也可以引用上游节点输出的图片。​
* **参考程度**：参考程度越高，图像越相似。​

输入​

提示词中可使用的输入参数，用于动态传入内容。输入参数可以指定为一个固定值，也可以引用上游节点的输出。​

提示词​

图像模型的提示词，用于下达图像生成相关的指令，也就是你对画面的描述。提示词中只需要输出正向和负向的关键词即可。支持引用输入中定义的变量，引用方式为{{变量名}}。​

* 正向提示词：即 positive prompt，必选。用于描述你想要出现在画面中的内容，例如 1girl,necklace, jewelry,best quality,。​
* 负向提示词：即 negative Prompt，可选。用于描述你的画面中不想生成的内容，例如 watermark, long neck, missing finggers, extra arms,worst quality, low quality, normal quality,。同时扣子也预设了一系列负向提示词，包括 nsfw,nude,blurry,watermark,identifying mark, low resolution,mutated,lack of hierarchy。​

输出​

节点的输出参数固定为：​

* data：Image 格式的图像，即模型生成的图像。通常是一个公开可访问的 URL 链接。​
* msg：节点执行状态，success 表示执行成功。​

示例​

文生图​

通过文字描述生成一张动漫风格的图片。模型选择动漫，引用开始节点的用户原始输入作为正向提示词，并添加一系列正向的关键词作为提示词。负向提示词为 low quality，表示避免低质量图像。节点配置及试运行结果如下：​

​

​​图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。​

​

图生图​

通过参考图片生成一张相同人物造型和姿势的动漫图片。模型选择动漫，参考图上传一张想要的人物造型与动作图片。引用开始节点的用户原始输入作为正向提示词。负向提示词为 low quality，表示避免低质量图像。节点配置及试运行结果如下：​

​

​​图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。​

​

​

​

画板节点

画板节点是一个支持自定义绘制的图形创作节点。画板节点通常用于图文排版和设计场景，例如电商海报、营销 banner、社交媒体博文配图等。​

你可以在画板节点中插入各种元素，例如上传指定图片、添加文本、添加矩形等各种线性图形、画笔自由绘制等，还可以添加变量元素，引用上游节点的输出。除此之外，你还可以设置元素图层、画板尺寸、画板颜色、画板透明度等。​

在画板编辑区域双击预览图即可编辑画板，你也可以在画板编辑右上角单击图标来进入编辑状态。​

**注意**

* 画板节点限时免费中。详细收费计划请关注后续官网通知。​
* 每次执行工作流时，叠图和添加文字插件节点的并发限制为单节点 10 次/秒。其他官方图像插件和图像节点的并发限制为 4 次/秒。​

​

​

变量元素​

画板节点中添加的变量可以作为一个画板元素展示在画面中，这种画板元素被称为变量元素。​

如果需要引用上游节点的输出参数，作为画板节点的元素之一，可以在画板节点的配置页面元素设置区域单击加号（+）来增加一个元素，元素值引用上游节点的输出参数。​

​

​​​

​

添加变量元素后，此元素不会默认展示在画布中，你需要在画布手动添加这个元素。目前支持通过以下方式在画布中添加变量元素：​

​

|  |  |  |
| --- | --- | --- |
| **添加方式**​ | **操作说明**​ | **示例**​ |
| 直接插入变量元素​ | 在画布中单击引用变量图标，添加一个变量元素。默认插入与变量类型相同的元素对象，即 String 类型对应文本对象，image类型对应图片对象。​  ​  ​ | 例如根据指定输入文案生成小红书笔记配图。我们可以在开始节点中添加输出参数 title，并在画板节点中引用这个参数 title，title 参数的值会自动添加到画板节点中。​  ​  ​​​  ​  ​ |
| 在固定元素中设置引用变量​ | 如需为固定元素设置引用，选中文本对象或图片对象，并在配置框中选择要引用的变量即可。​  固定元素中，仅文本对象和图片对象可以引用变量，且均可引用 image 和 String 类型的变量。​   * 如果文本对象引用了 image 变量，最终生成的图片中，本文框内会展示变量对应图片的 URL。​ * 如果 String 变量的值是一个公开可访问的图片 URL，那么图片对象可以引用这个 String 变量，最终生成的图片中，自动将 URL 渲染为图片。​ | ​  ​​​  ​ |

​

固定元素​

固定元素指添加到画板中的非变量元素，包括本地上传的图片、固定文本、画板中绘制的线性图形、画笔自由绘制等。对于固定元素的常见操作如下：​

​

|  |  |  |
| --- | --- | --- |
| **操作**​ | **说明**​ | **示例**​ |
| 添加图片​ | 在画板中单击对应图标，上传一张本地图片到画板节点中。​  添加图片后，可以设置**填充样式**和**描边样式**。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 添加文字​ | 在画板中单击对应图标，添加一段文本内容到画板节点中。支持添加单行文本或区块文本，其中区块文本指固定大小的文本框。​  添加文本框后，双击文本框可以设置文字内容，单击文本框可以调整文字的大小、位置等展示效果。​ | ​  ​​​  ​ |
| 添加图形​ | 在画板中单击对应图标，添加一个线性图形到画板节点中。​  添加图形后，可以设置**填充样式**和**描边样式**。​ | ​  ​​​  ​ |
| 画笔模式​ | 在画板中单击对应图标，进入画笔模式。​  在画笔模式下，你可以自由操控鼠标或触控板，在画板中自由绘制。​  如需删除画笔轨迹，可以单击画笔图标退出画笔模式，再按下键盘上的 **Delete** 键。​ | ​  ​​​  ​ |

​

元素样式​

添加变量元素和固定元素之后，可以随时调整元素的样式、展示效果、通过快捷键复制或删除元素。​

​

|  |  |  |
| --- | --- | --- |
| **操作**​ | **说明**​ | **示例**​ |
| 调整元素样式​ | 选中变量元素，可以快速调整变量元素的位置、元素大小等。选中变量元素后，界面会浮现出用于编辑图形的工具栏，你还可以在浮现的工具栏中点击相应图标，调整变量元素的外观样式，例如调整字体大小、颜色、字体类型等。​  此外，你还可以在画板中修改变量元素，直接预览显示效果，例如编辑文字预览效果、更换图片预览效果。​ | ​  ​​​  ​ |
| 调整元素位置​ | 在画布中直接鼠标拖拽元素即可移动元素位置。移动时页面会显示坐标辅助线，帮助你判断元素之间的相对位置。​  选中多个元素之后，还可以批量设置对齐方式。​ | * 移动：​   ​  ​​​  ​   * 对齐：​   ​  ​​​  ​ |
| 快捷键操作​ | 画布中支持通过以下快捷键复制操作：​   * Command/Ctrl+C：拷贝​ * Command/Ctrl+V：粘贴​ * Command/Ctrl+D：拷贝并粘贴​ * Alt+鼠标拖拽：自定义坐标快速复制​ * Backspace/Delete：删除​ | ​  ​​​  ​ |
| 删除元素​ | 支持快捷键（Backspace/Delete）删除变量元素在内的所有元素。​ | ​  ​​​  ​ |

​

画板设置​

画板节点的基础设置包括尺寸、颜色、透明度等画板设置、元素图层、预览比例等。​

​

|  |  |  |
| --- | --- | --- |
| **操作**​ | **说明**​ | **示例**​ |
| 调整画板设置​ | 在画板中单击对应图标，调整画板设置。支持调整画板尺寸、颜色和透明度。​ | ​  ​​​  ​ |
| 重置视图​ | 调整预览比例后，单击重置视图图标，恢复到 100% 预览比例。​ | ​  ​​​  ​  ​  ​​​  ​ |
| 调整预览比例​ | 在画板中单击加号或减号，调整预览比例。预览的最小比例为 100%。​ | ​  ​​​  ​ |
| 调整元素图层​ | 选中元素后，在画板中单击对应图标，将指定元素置底或置顶。适用于元素重叠展示的场景下调整元素所在图层，例如在某个图形上叠加文字。​ | ​  ​​​  ​ |

​

输出​

画板节点的输出参数固定为：​

* data：Image 格式的图像，即排版后的最终图像。通常是一个公开可访问的 URL 链接。​
* msg：节点执行状态，success 表示处理成功。​

示例​

例如通过画板节点搭建一个生成小红书配图的图像流。其中：​

* 添加变量元素 mbti 和 title，引用开始节点的输入参数，作为画板中动态调整的 MBTI 部分、主题部分。​
* 添加变量元素 image，引用开始节点的图片输入，作为画板中的配图。​
* 最后为图片设置一个底色、调整尺寸即可。​

画板节点配置：​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​

​

生成效果：​

​

​​​

​

​

​

​​

创建会话

会话指的是用户与模型之间基于同一主题的一系列连续的对话交互，会话中存储用户输入的消息和模型回复的消息。创建会话节点用于在工作流或对话流中创建一个空的会话。​

节点说明​

对话流通过会话来存储用户和模型的对话消息。每个对话流都需要绑定一个会话作为开始节点的默认会话，执行对话流时产生的消息都会自动写入到这个会话中。创建会话节点用于在工作流或对话流中生成一个新的会话，新会话中默认没有任何消息记录，它被绑定对话流并运行一次后才会有消息接入。​

在一些需要在运行过程中创建会话的场景，可以通过会话节点创建会话，再将新会话作为对话流的入参，运行对话流、写入消息。​

**说明**

* 会话类节点只能在应用中使用。智能体暂不支持包含**创建会话**、**清除上下文**等**会话**类节点的工作流或对话流。​
* 创建会话节点创建出的是动态会话，试运行时会展示在应用会话列表的动态会话区域，是应用的草稿态临时数据，与线上数据隔离。​

​

配置创建会话节点​

输入​

创建会话节点的输入参数固定为 conversationName，必选，String 类型，表示新会话的名称。同一个应用中会话名称必须唯一。会话名称参数可以指定为一个固定值，或引用上游节点的输出参数。​

输出​

创建会话节点的输出参数用于展示创建会话节点的执行结果。参数固定为：​

* isSuccess：Boolean 类型，表示创建会话节点是否执行成功。​
* isExisted：Boolean 类型，表示应用中是否已存在此名称的会话。​
* conversationId：新会话的 ID。​

输出示例如下：​

* 应用中没有同名的会话，创建新会话成功，返回会话 ID：​
* ​

conversationId : 7444890359211262037​

isSuccess : true​

isExisted : false​

​

* 应用中已有同名的节点，所以未创建会话，此时返回原会话的 ID：​
* ​

conversationId : 7444890359211262004​

isSuccess : true​

isExisted : true​

​

试运行创建会话节点​

对于资源库中的工作流或对话流，试运行创建会话节点时，需要关联应用，表示在指定的应用中创建会话。​

​

​​​

​

调试完成后，可以在试运行按钮左侧单击配置按钮，在已存会话页面中查看刚创建出的新会话。试运行创建出的会话是草稿态的测试数据，和线上数据隔离，你也可以按需删除这些会话。​

因为试运行时绑定了应用，**已存会话**页面会展示该应用中的所有会话。​

​

​​图形用户界面, 应用程序, Teams

AI 生成的内容可能不正确。​

​

​

清除上下文节点

清除上下文节点用于清除指定会话中存储的上下文消息。​

节点说明​

会话中的历史消息也被称为上下文，对大模型的性能和文本生成效果有重要影响。较长的上下文可以为模型提供更多的信息参考，生成与上下文语义匹配的文本片段，使模型生成更为连贯、逻辑性更强的内容。​

多轮对话中启动新话题时，过时的上下文也会同样影响模型的生成效果，例如模型可能召回一些过时的知识或记忆。此时可以使用清除上下文节点，清除会话中已存储的上下文，成功清除后，对话流中的大模型节点会读到的对话历史为空，之后的对话不会收到之前历史消息的影响。​

例如以下对话中，清除上下文后模型的回复会更加准确：​

* 用户：最近看了几个恐怖电影。​
* 模型：真是非常刺激和紧张的观影体验！你看了哪些恐怖电影？​
* 用户：你有什么别的好电影推荐吗？​
* 模型：当然！恐怖电影《闪灵》你感兴趣吗？​
* 【清除上下文】​
* 用户：你有什么别的好电影推荐吗？​
* 模型：根据最新的搜索结果，这里有一些 2024 年备受期待和推荐的电影。​

**说明**

* 会话类节点只能在应用中使用。智能体暂不支持包含**创建会话**、**清除上下文**等**会话**类节点的工作流或对话流。​
* 试运行清除上下文节点时，只能清除应用中测试数据的上下文，无法清除线上数据。​
* 清除上下文只会清除模型可见的上下文，不会真正删除会话中存储的历史消息。所以清除上下文之后，仍旧可以通过查询消息列表节点查看完整的消息内容。​

​

配置清除上下文节点​

输入​

清除上下文节点的输入参数固定为 conversationName，必选，String 类型，表示待清除上下文的会话名称。会话名称参数可以指定为一个固定值，或引用上游节点的输出参数。​

输出​

清除上下文节点的输出参数用于展示清除上下文节点的执行结果。参数固定为 isSuccess，Boolean 类型，表示此节点是否执行成功。​

​

​​​

​

试运行清除上下文节点​

对于资源库中的工作流或对话流，试运行清除上下文节点时，需要关联应用，表示清除指定应用中的指定会话上下文。​

​

​​​

​

调试完成后，可以在试运行按钮左侧单击配置按钮，在已存会话页面中查看已清除上下文的会话。对话框中会展示“上下文已清除”的字样。​

​

​​图形用户界面, 应用程序

AI 生成的内容可能不正确。​

​查询消息列表

查询消息列表节点用于查看指定会话中的所有历史消息。​

节点说明​

每个对话流都需要绑定一个会话作为开始节点的默认会话，执行对话流时产生的消息都会自动写入到这个会话中，包括用户输入的消息和对话流输出的消息。查询消息节点可以展示指定会话中已记录的所有历史消息，在工作流或对话流中添加这个节点并绑定一个列表组件，你可以在用户界面中实时展示对话消息列表。​

**说明**

* 会话类节点只能在应用中使用。智能体暂不支持包含**创建会话**、**清除上下文**等**会话**类节点的工作流或对话流。​
* 试运行查询消息列表节点时，只能查看应用中测试数据，无法查看线上数据。​
* 消息在服务端的保存时长为180天，期满后，消息将自动从会话的消息记录中删除。​

​

配置查询消息列表节点​

输入​

消息列表节点预置了多个输入参数，且不支持手动调整参数名称。​

默认按时间倒序查看最近的 50 条消息，支持通过 limit 参数翻页。如果会话中消息较多，你可以先查看部分消息，获取消息列表中返回的 first\_id、last\_id，再通过 beforeId 或 afterId 从指定位置开始查看消息。​

输入参数的详细说明如下：​

​

|  |  |
| --- | --- |
| **参数**​ | **说明**​ |
| conversationName​ | 待查看消息列表的会话名称。必选，String 类型，可以指定为一个固定值，或引用上游节点的输出参数。​ |
| limit​ | 返回的最大消息数量。必选，Integer 类型，默认为 50，取值范围为 1~50。​  可以指定为一个固定值，或引用上游节点的输出参数。​ |
| beforeId​ | 查看指定位置之前的消息。必选，String 类型，可以指定为一个固定值，或引用上游节点的输出参数。​   * 传入空字符串，表示不指定位置。​ * 如需向前翻页，则指定为输出参数中的 first\_id。​ |
| afterId​ | 查看指定位置之后的消息。必选，String 类型，可以指定为一个固定值，或引用上游节点的输出参数。​   * 传入空字符串，表示不指定位置。​ * 如需向后翻页，则指定为输出参数中的 last\_id。​ |

​

输出​

创建会话节点的输出参数用于返回执行结果和消息列表。参数固定如下：​

​

|  |  |  |
| --- | --- | --- |
| **参数**​ | **类型**​ | **说明**​ |
| messageList​ | Array<Object>​ | 指定会话中的消息列表。​ |
| messageList.messageId​ | String​ | Message ID，即消息的唯一标识。​ |
| messageList.role​ | String​ | 发送这条消息的实体。取值：​   * **user**：代表该条消息内容是用户发送的。​ * **assistant**：代表该条消息内容是智能体发送的。​ |
| messageList.content\_type​ | String​ | 消息内容的类型，取值包括：​   * 1：文本。​ * 2：多模态内容，文件、图片、语音等。​ |
| messageList.content​ | String​ | 消息的内容，支持纯文本、多模态等多种类型的内容。​ |
| first\_id​ | string​ | 返回的消息列表中，第一条消息的 Message ID。​ |
| last\_id​ | string​ | 返回的消息列表中，最后一条消息的 Message ID。​ |
| has\_more​ | bool​ | 是否已返回全部消息。​   * true：未返回全部消息，可再次调用此接口查看其他分页。​ * false：已返回全部消息。​ |

​

试运行查询消息列表节点​

对于资源库中的工作流或对话流，试运行查询消息列表节点时，需要关联应用，表示查看指定应用中的指定会话。​

​

​​​

​

试运行此节点，得到的消息列表如下：​

试运行结果：​

​

​​​

​

对应的消息列表：​

​

​​图形用户界面, 文本, 应用程序, 聊天或短信

AI 生成的内容可能不正确。​

​

​

​设置定时触发器

设置定时触发器节点用于在扣子应用中创建一个定时触发器，或修改已创建的定时触发器。​

节点说明​

和智能体一样，扣子应用也支持设置定时触发器，在指定的时间点触发定时器调用某个工作流。扣子应用的触发器分为预设触发器和用户触发器两种，前者是开发者编排应用时创建的触发器，对所有用户生效，后者是扣子应用的用户通过定时触发器节点创建的，仅对指定用户生效。关于扣子应用触发器的详细说明，可参考​为应用设置触发器。​

设置定时触发器用于用户主动设置触发器的场景，例如在一个英语学习的 AI 应用中，每个用户都可以创建自己的学习计划，A 用户可以要求每天 10 点学习 10 个单词，B 用户可以要求 12 点学习 5 个。​

**说明**

设置定时触发器等触发器节点仅支持在扣子应用中创建或使用，包含触发器节点的工作流无法添加到智能体中。​

​

设置节点参数​

输入​

设置定时触发器节点预设多个参数，配置说明如下:​

​

|  |  |  |
| --- | --- | --- |
| 变量​ | **说明**​ | **示例**​ |
| id​ | 触发器的 ID，是触发器在扣子平台中的唯一标识，由扣子平台自动创建，不支持手动创建或编辑。​  **说明**   * 创建触发器时，无需设置 id，系统会自动生成一个 id。​ * 更新触发器配置时，需要输入 id，表示更新指定的触发器。​   ​ | ​  ​  ​​​  ​  ​ |
| 用户id​ | 使用触发器的扣子用户 ID，此触发器仅对该用户生效。扣子应用的系统变量 sys\_uuid 可以用来标识用户，你可以使用变量节点获取系统变量 sys\_uuid 的值，并赋值给设置定时触发器节点的用户id 参数。​ |
| 名称​ | 触发器的名称，用于标识触发器的用途。​ |
| 时区​ | 触发器的运行时区，建议根据 AI 应用的用户所在时区设置，确保定时任务在正确的时间启动。​ |
| 触发时间​ | 触发器的触发时间，触发时自动执行这个工作流。可以选择预设的时间选项（如每天 18 点），也可以使用 Cron 表达式来自定义复杂的时间。其中，Cron 表达式支持 AI 自动生成，也可以引用上游节点的输出。​ |
| 绑定工作流​ | 选择一个工作流，当触发器被激活时，扣子平台会自动运行该工作流。​ |
| 参数​ | 工作流开始节点设置的参数，其中必选参数应设置运行时的参数值。​ |

​

输出​

设置定时触发器节点固定的输出参数为 triggerId，String 类型，表示已创建或更新的触发器 ID。​

示例​

为一个扣子应用添加每日个性化书单推荐的功能，可以根据不同用户的偏好推荐好书、为每个用户设置触发器，更新推荐书单，用户可以主动查看自己更新后的每日书单。​

主要功能及对应流程如下：​

​

|  |  |  |
| --- | --- | --- |
| **功能模块**​ | **说明**​ | **工作流示例**​ |
| 更新推荐书单​ | 通过变量节点获取用户 ID 和读书偏好，通过数据库节点查看过去已经推荐过的图书名称，大模型根据读书偏好和已推荐的书单，生成新的推荐书单，并写入数据库中。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 创建或更新用户触发器​ | 通过变量节点获取用户 ID，为用户设置个性化的触发器，绑定**更新推荐书单**的工作流，每天定时为每个用户更新书单。​  此工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，创建触发器，每天定时更新书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 查看每日书单​ | 通过变量节点获取用户 ID，根据用户 ID 在数据库中查询今日推荐书单。​  这个工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，展示今日推荐书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 查看触发器列表​ | 查看用户已创建的所有触发器列表。通过变量节点获取用户 ID，根据用户 ID 查询触发器列表。​ | ​  ​​图形用户界面, 应用程序, Teams  AI 生成的内容可能不正确。​  ​  ​ |
| 删除书单触发器​ | 不再需要获取每日书单时，可以删除书单触发器，每日书单不会定时更新。​  通过变量节点获取用户 ID，根据用户 ID 查询触发器列表、获取触发器 ID，根据用户 ID 和触发器 ID 删除指定用户的触发器。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |

​

​

查询定时触发器节点

查询定时触发器节点用于查看指定用户的触发器。​

节点说明​

和智能体一样，扣子应用也支持设置定时触发器，在指定的时间点触发定时器调用某个工作流。设置定时触发器用于用户主动设置触发器的场景，例如在一个英语学习的 AI 应用中，每个用户都可以创建自己的学习计划，A 用户可以要求每天 10 点学习 10 个单词，B 用户可以要求 12 点学习 5 个。关于扣子应用触发器的详细说明，可参考​为应用设置触发器。​

通过​设置定时触发器节点节点创建触发器之后，可以通过查询定时触发器节点查看指定用户的触发器，支持查看某个触发器的详情，也可以查看触发器列表。​

**说明**

设置定时触发器等触发器节点仅支持在扣子应用中创建或使用，包含触发器节点的工作流无法添加到智能体中。​

​

设置节点参数​

输入​

查询定时触发器节点预设多个参数，配置说明如下:​

​

|  |  |
| --- | --- |
| **参数**​ | **说明**​ |
| id​ | 触发器的 ID，是触发器在扣子平台中的唯一标识，由扣子平台自动创建。可以通过设置定时触发器节点的输出参数中获取，也可以先不指定 id，查看此用户的所有触发器详情，详情中包括每一个触发器的 ID。​  **说明**   * 未指定 id 时，此节点返回指定用户的所有触发器。​ * 指定 id 时，此节点仅返回指定触发器的详细信息。​   ​ |
| 用户id​ | 使用触发器的扣子用户 ID，此触发器仅对该用户生效。扣子应用的系统变量 sys\_uuid 可以用来标识用户，你可以使用变量节点获取系统变量 sys\_uuid 的值，并赋值给设置定时触发器节点的用户id 参数。​ |

​

输出​

查询定时触发器节点固定的输出参数为 ouputList，数组类型，表示触发器列表。其中包含每个触发器的 ID 等信息，结构体详细说明如下：​

​

|  |  |
| --- | --- |
| **参数**​ | **说明**​ |
| triggerId​ | 已创建的触发器 ID。​ |
| triggerName​ | 已创建的触发器名称。​ |
| createTime​ | 触发器的创建时间。​ |
| triggerTime​ | 触发器的触发时间，展示为 Cron 表达式格式。​ |
| userId​ | 触发器对应的用户 ID。​ |

​

示例​

为一个扣子应用添加每日个性化书单推荐的功能，可以根据不同用户的偏好推荐好书、为每个用户设置触发器，更新推荐书单，用户可以主动查看自己更新后的每日书单。​

主要功能及对应流程如下：​

​

|  |  |  |
| --- | --- | --- |
| **功能模块**​ | **说明**​ | **工作流示例**​ |
| 更新推荐书单​ | 通过变量节点获取用户 ID 和读书偏好，通过数据库节点查看过去已经推荐过的图书名称，大模型根据读书偏好和已推荐的书单，生成新的推荐书单，并写入数据库中。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 创建或更新用户触发器​ | 通过变量节点获取用户 ID，为用户设置个性化的触发器，绑定**更新推荐书单**的工作流，每天定时为每个用户更新书单。​  此工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，创建触发器，每天定时更新书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 查看每日书单​ | 通过变量节点获取用户 ID，根据用户 ID 在数据库中查询今日推荐书单。​  这个工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，展示今日推荐书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 查看触发器列表​ | 查看用户已创建的所有触发器列表。通过变量节点获取用户 ID，根据用户 ID 查询触发器列表。​ | ​  ​​图形用户界面, 应用程序, Teams  AI 生成的内容可能不正确。​  ​  ​ |
| 删除书单触发器​ | 不再需要获取每日书单时，可以删除书单触发器，每日书单不会定时更新。​  通过变量节点获取用户 ID，根据用户 ID 查询触发器列表、获取触发器 ID，根据用户 ID 和触发器 ID 删除指定用户的触发器。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |

删除定时触发器节点

删除定时触发器节点用于删除指定用户的触发器。​

节点说明​

和智能体一样，扣子应用也支持设置定时触发器，在指定的时间点触发定时器调用某个工作流。设置定时触发器用于用户主动设置触发器的场景，例如在一个英语学习的 AI 应用中，每个用户都可以创建自己的学习计划，A 用户可以要求每天 10 点学习 10 个单词，B 用户可以要求 12 点学习 5 个。关于扣子应用触发器的详细说明，可参考​为应用设置触发器。​

通过​设置定时触发器节点节点创建触发器之后，可以通过删除定时触发器节点删除指定用户的触发器，支持删除某个触发器，也可以删除指定用户的所有触发器。​

**说明**

设置定时触发器等触发器节点仅支持在扣子应用中创建或使用，包含触发器节点的工作流无法添加到智能体中。​

​

设置节点参数​

输入​

删除定时触发器节点预设多个参数，配置说明如下:​

​

|  |  |
| --- | --- |
| **参数**​ | **说明**​ |
| id​ | 触发器的 ID，是触发器在扣子平台中的唯一标识，由扣子平台自动创建。可以通过设置定时触发器节点的输出参数中获取 id，也可以先不指定 id，查看此用户的所有触发器详情，详情中包括每一个触发器的 ID。​  **说明**   * 未指定 id 时，表示删除指定用户的所有触发器。​ * 指定 id 时，表示删除指定触发器。​   ​ |
| 用户id​ | 使用触发器的扣子用户 ID。扣子应用的系统变量 sys\_uuid 可以用来标识用户，你可以使用变量节点获取系统变量 sys\_uuid 的值，并赋值给设置定时触发器节点的用户id 参数。​ |

​

输出​

删除定时触发器节点固定的输出参数如下：​

​

|  |  |
| --- | --- |
| **参数**​ | **说明**​ |
| success​ | 是否已成功删除触发器。​ |
| numberOfDeletions​ | 删除的定时触发器数量。​ |

​

示例​

为一个扣子应用添加每日个性化书单推荐的功能，可以根据不同用户的偏好推荐好书、为每个用户设置触发器，更新推荐书单，用户可以主动查看自己更新后的每日书单。​

主要功能及对应流程如下：​

​

|  |  |  |
| --- | --- | --- |
| **功能模块**​ | **说明**​ | **工作流示例**​ |
| 更新推荐书单​ | 通过变量节点获取用户 ID 和读书偏好，通过数据库节点查看过去已经推荐过的图书名称，大模型根据读书偏好和已推荐的书单，生成新的推荐书单，并写入数据库中。​ | ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 创建或更新用户触发器​ | 通过变量节点获取用户 ID，为用户设置个性化的触发器，绑定**更新推荐书单**的工作流，每天定时为每个用户更新书单。​  此工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，创建触发器，每天定时更新书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​  ​ |
| 查看每日书单​ | 通过变量节点获取用户 ID，根据用户 ID 在数据库中删除今日推荐书单。​  这个工作流可以绑定用户界面中的某个按钮，用户单击按钮时触发工作流，展示今日推荐书单。​ | ​  ​  ​​图形用户界面, 应用程序  AI 生成的内容可能不正确。​  ​ |
| 查看触发器列表​ | 查看用户已创建的所有触发器列表。通过变量节点获取用户 ID，根据用户 ID 删除触发器列表。​ | ​  ​​图形用户界面, 应用程序, Teams  AI 生成的内容可能不正确。​  ​  ​ |
| 删除书单触发器​ | 不再需要获取每日书单时，可以删除书单触发器，每日书单不会定时更新。​  通过变量节点获取用户 ID，根据用户 ID 删除触发器列表、获取触发器 ID，根据用户 ID 和触发器 ID 删除指定用户的触发器。 |  |

工作流常见问题

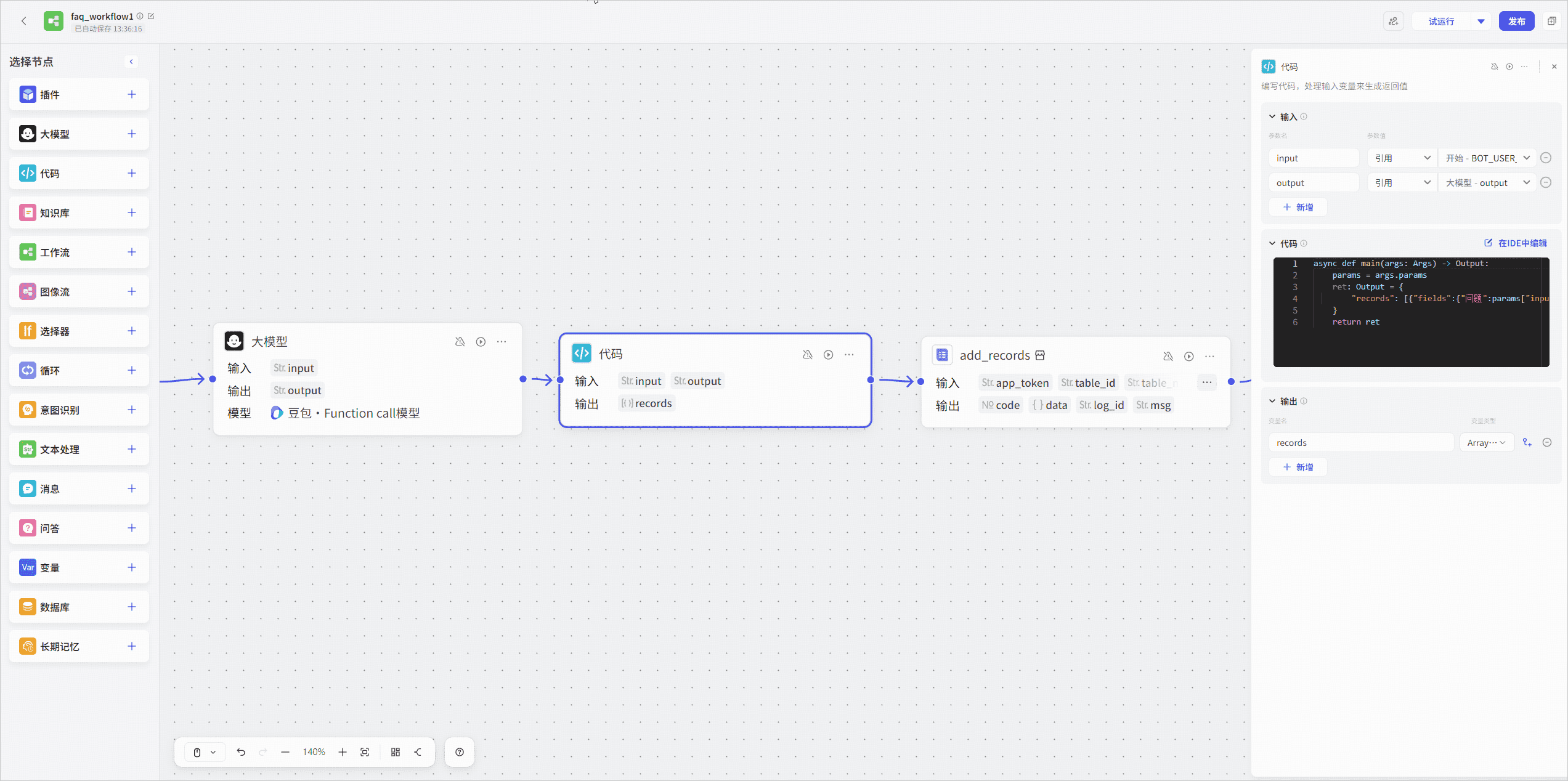
如何生成一个 Array 类型的参数？​

飞书多维表格 add\_records 等节点的必选输入参数中要求参数格式为 Array。对于 Array 类型的输入参数，建议引用前置节点中的输出参数，目前代码等类型的节点支持 Array 类型的输出参数。如果你需要将多个不同类型的参数拼接为一个 Array 类型的参数，可以考虑使用代码节点进行参数拼接。​

例如通过代码节点拼接用户提问字段 input 和模型回复字段 output，并将这两个字段分别写入多维表格的问题和Bot回答两列中。​

工作流示例：​

​

​​​

​

​

代码示例：​

​

async def main(args: Args) -> Output:​

params = args.params​

ret: Output = {​

"records": [{"fields":{"问题":params["input"],"Bot回答":params["output"]}}]​

}​

return ret​

​

​

Array 类型的输入参数可以指定一个固定值吗？​

所有类型的参数均支持指定为一个固定的值，但指定的固定值会被作为 String 类型写入。对于 Array 类型的输入参数，建议引用前置节点中的输出参数，如果一定要指定固定的值，可以在此节点之前增加一个代码节点，将固定值转为 Array 类型，并引用代码节点的输出。​

例如：​

**错误输入**：​

直接将 List 格式的 records 参数设置为固定值时，页面报错错误：wrong param type, param[records] should be array。​

​

​​​

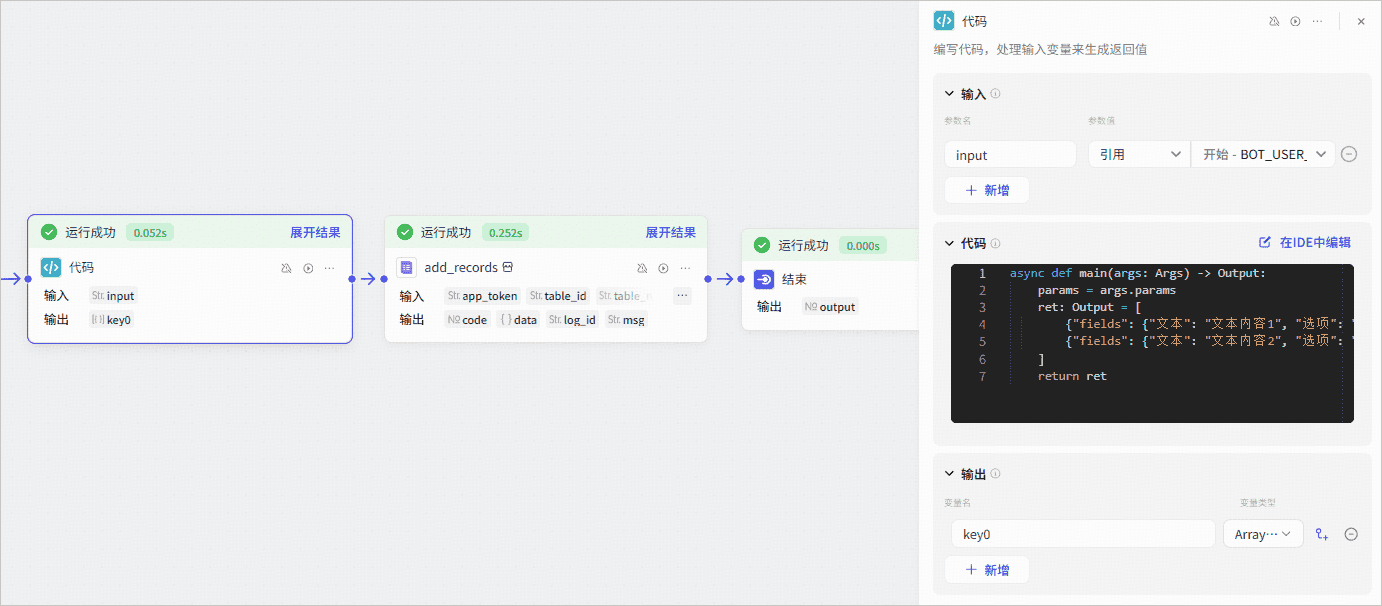
​

​

**正确输入**：​

增加一个代码节点，用于将固定的值转为 Array 类型的参数，records 引用代码节点的输出参数，工作流执行成功。​

​

​​​

​

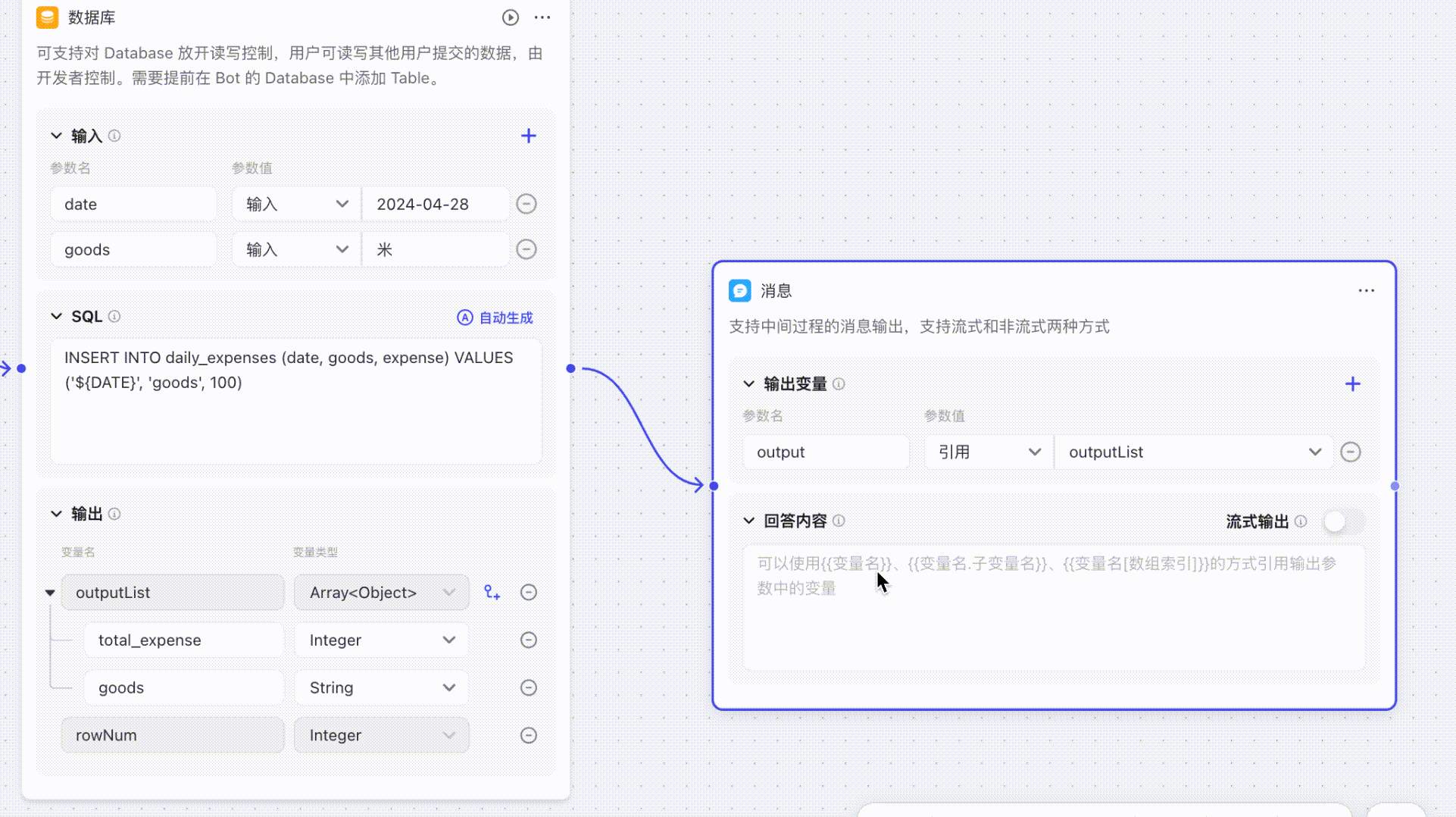
​

​

如何引用数组中的对象​

**结束**节点、**消息**节点和**大语言模型 LLM** 节点等节点支持直接引用当前节点导入的参数，且支持联想，当输入{{后，能够自动关联所引用的数组/对象数据。​

​

​​​

​

智能体没有按照预期调用插件或工作流​

如果智能体没有按照预期调用插件或者工作流，你可以尝试：​

* 在智能体的**人设与回复逻辑**区域，通过自然语言明确要求智能体在何种场景下调用哪个插件或者工作流。​
* 如果是你自己创建的插件或工作流，可以优化插件或工作流的名称和简介，便于智能体准确识别插件或工作流的功能，提高智能体自动使用插件或工作流的概率。​

工作流中，智能体连续两次回复的时间间隔限制**是多久？** ​

一次对话中，智能体连续两次回复的时间间隔限制为 2 分钟。 ​

工作流的消息节点、问答节点等节点会在工作流运行期间主动向用户发送消息。工作流运行时，如果智能体连续两次发送消息的时间间隔大于 2 分钟，后端服务会判断工作流运行超时，停止运行工作流并报错。例如节点数量多、工作流逻辑复杂时可能后端处理时间长，导致工作流运行超时。​

建议合理编排设置工作流，缩短两次智能体回复的时间间隔，避免后端服务判断超时。例如在后端处理耗时较久的环节适当增加消息节点、为消息节点或结束节点开启流式输出，尽量缩短智能体消息的时间间隔。​

工作流运行时间超过 3 分钟导致运行超时，应该怎么解决？​

工作流默认为同步运行，整体超时时间为 3 分钟。如果工作流复杂或包含一些运行耗时长的节点，可能会导致工作流整体运行耗时长，智能体判断为工作流运行超时。​

在这种场景下，你可以设置工作流为异步运行，整体超时时间为 24 小时。工作流异步运行时会默认返回一条预设的回复内容，用户可以继续与智能体对话，工作流运行完毕后智能体会针对触发工作流的指令做出最终回复。详细信息请参考​步骤四：（可选）设置异步运行。​

如何恢复工作流的历史版本​

仅团队空间中开启了**多人协作**的工作流可以查看并恢复历史版本，操作步骤可参考​管理历史版本。​

​

​​​

​

设置参数时显示“（x）暂无数据”​

为工作流节点的变量设置参数值时，如果希望引用上游节点的输出参数，但页面显示暂无数据，可能原因如下：​

* 此工作流节点未连接任何上游节点。建议从节点左侧添加一条连接线，连接到其他节点后重试。​
* ​

​​​

​

* 此节点上游未定义任何参数。建议在上游节点合理设置参数，或为此节点的参数设置一个固定值。​
* 此参数限制数据类型，且上游节点没有符合此数据类型的参数。例如某个插件节点的输入参数固定为 Array<Object>，那么它引用的参数必须是同类型的参数，参数值的下拉列表中也仅展示 Array<Object> 的参数。​

**工作流报错“工作流的版本号冲突: xxx” ，应如何处理？**​

问题现象​

工作流运行时报错**“工作流的版本号冲突，冲突工作流 ID: xxx，父工作流 ID: xxxx”** ，其中“xxx”部分为工作流的 ID。​

​

​​​

​

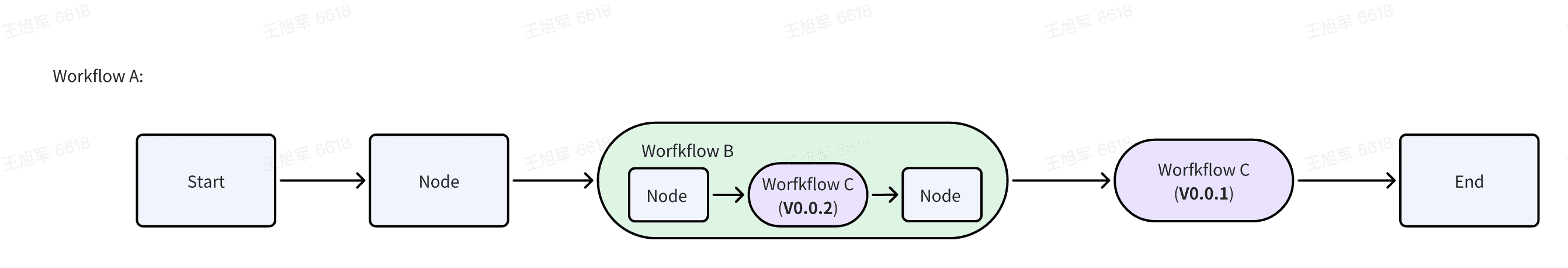
原因及方案​

**版本号冲突​**

如果某个工作流通过工作流节点嵌套的方式多次引用了另一个工作流，那么必须引用同样的工作流版本。如果子工作流发布了新版本，但是父工作流仅更新了部分节点版本，并未更新每一处引用的版本，会导致父工作流引用的同一子工作流版本不一致，引发此报错。​

例如，工作流 A 引用了工作流 C，同时也引用了包含工作流 C 的工作流 B。如果工作流 A 直接引用的工作流 C 版本号升级为 V0.0.2，但工作流 B 引用的工作流 C 仍为旧的版本 V0.0.1，那么执行工作流 A 时页面会报错“工作流的版本号冲突: {工作流C的ID}**，父工作流 ID： {工作流B的ID}， {工作流A的ID}**”。​

​

​​​

​

解决方案如下：​

报错中会显示版本冲突的父工作流、子工作流 ID，你可以凭借此 ID 定位到工作流名称，并在父工作流中升级引用的此子工作流版本，保证每一处均引用最新版本。具体操作步骤如下：​

1.

**确定版本冲突的子工作流名称**。​

在资源库中打开任意工作流，将 URL 中 workflow\_id= 之后的字符串替换为报错信息中获取的子工作流 ID“xxx”，重新访问此页面后可在页面左上角查看工作流名称。例如以下工作流 ID 对应的工作流名称为 wfC。​

​

​​​

​

2.

**找到对应的父工作流节点**。​

报错信息中包含父工作流 ID，你可以通过同样的方式定位到嵌套了这个子工作流的所有父工作流。​

打开运行报错的父工作流，确认父工作流的每一个工作流节点是否引用了步骤1中找到的子工作流。注意工作流节点可能存在多层嵌套关系，例如父工作流 A 嵌套工作流 B，工作流 B 中又嵌套了工作流 C，你需要找到工作流 A 中引用工作流 C 的每一个节点。这些节点引用的工作流 C 版本不是完全一致的。​

在工作流节点中，如果节点名称之后存在红色标签提示，表示这个子工作流已发布了最新版本，但当前引用的是历史版本。鼠标指向这个标签即可查看引用的子工作流版本号。​

​

​​​

​

3.

**为每个工作流节点升级子工作流的版本**。​

在工作流 A 中，为每一个引用子工作流 C 的节点升级工作流版本，确保这些节点引用的工作流 C 版本一致。目前仅支持升级到最新版本。升级子工作流版本的方式可参考【[引用工作流版本](https://www.coze.cn/open/docs/guides/workflow_version#c6aaf871)】。​

全部升级后，可以试运行这个工作流，如果试运行通过，表示此版本冲突的问题已解决。​

​

​​​

​

**历史数据问题​**

对于工作流版本管理功能上线前已经存在的工作流节点，节点中可能不显示工作流版本，表示此节点始终引用最新版本工作流。如果工作流中同时存在带版本号和不带版本号的工作流节点，即使这些节点引用的同一版本，也会导致此报错。​

对于不显示版本号的工作流节点，建议在当前工作流中新增一个相同的工作流节点，新增时根据页面提示更新工作流版本，更新后原工作流节点会自动显示版本号信息。然后删除这个新增节点即可。​

​

​​​

​

​

上一篇