第 1 章

智能体概述

在大模型技术爆发之前，智能体（Agent）的功能与体验受限于人工智能（Artificial Intelligence，AI）技术，简单地讲，就是受限于“大脑”不够聪明。而当前 AI 技术在大模型上的突破与爆发，使得 AI Agent 技术也变得成熟并迅速火爆起来。大模型在生成、计算以及逻辑推理能力上都实现了质的飞跃，从而让 Agent 能够为用户带来更多的功能和更好的体验。本章将介绍 Agent 的背景知识，帮助读者对智能体有个一般性的认识。

1.1 为什么需要一个智能体（Agent）

为了体现当代 Agent 依赖于人工智能大模型的能力，我们将其称作 AI Agent、AI 智能体或者人工智能体，还有一些文章将其直译为“AI 代理”。目前，在计算机、人工智能专业技术领域，一般将 Agent 或 AI Agent 统一翻译为“智能体”。

在信息技术飞速发展的当下，人工智能领域持续推陈出新，智能体与 DeepSeek 大模型成为近期科技圈的焦点。在此时代背景下，“智能体+DeepSeek”正崭露头角，有望开启下一个重大的 IT 发展浪潮，引领未来变革，成为科技领域的下一个风口。

1. AI 的发展历程

要向读者讲清楚智能体的概念，我们首先需要了解人工智能（AI）的基本概念。

AI 是指通过计算机程序模拟人类智能的技术。这些程序可以执行诸如学习、推理、规划、自然语言处理等任务。自 20 世纪 50 年代 AI 概念提出以来，AI 技术经历了多次重大的突破。

AI 的发展历程可以分为以下几个重要阶段：

（1）初期发展阶段（20 世纪 50 年代—20 世纪 70 年代）：这个阶段的 AI 研究主要集中在符号主义和逻辑推理上。艾伦·图灵提出了图灵测试，作为衡量机器是否具有智能的标准。1956 年的达特茅斯会议确定了人工智能这个概念，被认为是 AI 研究的开端。因此，1956 年也被称为人工智能元年。

（2）早期发展阶段（20 世纪 80 年代—20 世纪 90 年代）：这一时期，专家系统成为 AI 研究的主要方向。专家系统通过编码专家知识来解决特定领域的问题，取得了显著的成果，但也暴露出知识获取难题和系统僵化等问题。

（3）现代发展阶段（21 世纪—）：随着计算能力和数据量的爆炸式增长，机器学习特别是深度学习技术迅速发展。AI 系统从依赖预定义规则转向通过数据训练模型，实现了图像识别、自然语言处理、自动驾驶等多种复杂任务。

2. Agent 能解决什么问题

大语言模型（Large Language Model，LLM，简称大模型）是近年来人工智能领域的重大突破之一。大模型旨在理解和生成人类语言，它们在大量的文本数据基础上进行训练，可以执行广泛的任务，包括文本总结、翻译、情感分析等。大模型的特点是基于神经网络、自然语言处理（Natural Language Processing，NLP）技术，多轮对话和写作生成能力非常优秀。尤其是像 GPT-4 这样的大语言模型，简直就是 AI 界的“超级明星”。这些模型通过海量的数据训练，具备强大的自然语言处理能力，可以生成高质量的文本，进行复杂的对话。例如，GPT-4 在文本生成和理解任务中就像是“语言魔法师”。大模型能适应不同的应用场景，从生成文本到处理对话，再到复杂的决策任务，样样在行。大模型能够生成高质量的自然语言文本，就像一个写作天才，永远不会有创意枯竭的时候。

现在已经有了 AI 大模型，例如 DeepSeek、OpenAI 的 GPT、字节的豆包等，为什么又出现了 Agent，这是刻意为了显得厉害搞出来的概念吗？

我们列举个例子，当你想要让大模型帮忙整理一篇文章，假设你这样问大模型：请你帮我生成一篇 100 万字的武侠小说。这个时候，AI 大模型给你什么答案！是不是写不出这么多字，也给不出你想要的答案？

为什么会出现这种问题？是不是 AI 大模型不够厉害？假设这件事让我们人类来做，我们一般会按照如下流程来完成这件事：

第一步：使用搜索引擎搜索一些相关书籍和信息进行阅读，为我们打开思路。
第二步：形成本书的大纲，并且考虑清楚每一个章节要编写的内容。
第三步：针对每一个章节进行内容的编写，在编写过程中可能会调整文章的大纲。
第四步：在编写后面章节的时候，可能会忘记前面写的内容，需要翻阅前面已经完成的内容。
第五步：文章初步完成之后，我们可能会找相关专业人士帮忙修改和审阅。
第六步：也是最后一步，经过几番调整之后，书稿最终成型。

大模型不能直接完成这件事，是大模型的能力不行吗？不是的，这是因为明显缺少了几个步骤：没有办法使用搜索引擎获取最新的外部信息（大模型的训练数据是以往数据，有日期限制的）；没有对整个事情进行规划（比如先写大纲，再编写每个章节，然后和别人讨论，最后成文）；大模型没有记忆的能力，由于上下文（脑容量）的限制，无法一次性完成 100 万字的文章，会造成前言不搭后语的现象。

而智能体 Agent 就是为了解决这个问题。思考一下，为了完成这个任务，我们用到了这些操作：上网查询、分解任务、逐步规划、审核修改。这里面涉及规划、思考、步骤等操作，还用到大脑、手、计算机或者助手等“工具”。大模型在这个过程中只充当了大脑思考的角色，它没有额外的工

具、没有规划和额外步骤，因此这个任务交给它，它是无法单独完成的。为了让大模型能够真正满足我们的要求，我们需要给它配备上网查询的能力、使用工具的能力、分解任务的能力等。

这就是 Agent 的价值，它使得大模型不仅仅是一个大脑，而且还是一个能做规划、能使用工具的类人智能体。

再举个例子，假设你让大模型帮你写工作日报，可以不可以？可以，但操作会很复杂。AI Agent 写工作日报就不同了，设定好格式、语气、任务等关键信息，你只需要口语化告诉它做了什么，剩下的事它会帮你自动完成。

假设你让大模型帮你写一篇软文，可以不可以？可以，但操作同样复杂，来来回回折腾几遍可能还不能让你满意。用 Agent 来写软文就不同了，提前设定好标题、开头、内容和语气等关键信息，告诉 Agent 你要写什么主题内容，它就能按照设定一步一步帮你完成。

假设你去旅游，让 AI 帮你介绍景点信息，可以不可以？可以，但是每次你都要主动发问，AI 还不一定回答正确。用 Agent 就不同了，按照景点情况设定好 Agent，你走到哪，它就会告诉你景点的相关信息；你问他洗手间在哪，它还可以根据你的位置给你指定最近的洗手间。

通过上面介绍的这些例子，我们理解了 Agent 和大模型使用起来到底有什么不同。简单来说，大模型相当于可以咨询的大脑，Agent 相当于有智慧又能干活的机器人。

1.2　认识 Agent

大模型时代，Agent 将基于大模型构建，此时的 Agent 是一种能够感知环境、进行决策和执行动作的智能体。是否具备通过独立思考、调用工具逐步完成给定目标的能力，成为基于大模型的 Agent 与基于传统 AI 技术的 Agent 之间最大的不同。这个区别也是很多人在给当代 Agent 下定义时一直强调的要点。例如，告诉 Agent 帮忙下单一份外卖，它就可以直接调用 App 选择外卖，再调用支付程序下单支付，而无须人类指定每一步的操作。

1. Agent 的组成

OpenAI 研发出 ChatGPT 并持续引领大模型发展，它定义 AI Agent 就是由大模型驱动，由规划决策（Planning）组件、记忆（Memory）组件、工具（Tools）组件、行动（Action）组件等组件组成的可以自主执行任务的程序，如图 1-1 所示，它就像一个代替人类完成工作的代理人。Agent 各个组件的作用概括如下：

- 规划决策组件：依赖于大模型自身的能力和提示词的指引，让模型反思和自我批评，并把任务分解成多个步骤，然后逐个完成。
- 记忆组件：分为短期记忆和长期记忆两种类型，用于记住沟通上下文。
- 工具组件：调用各种 API，包括日历、代码解释器、计算器、搜索 API 等。
- 行动组件：说白了就是它动手干活的部分。它能根据任务选择不同的方法——要查资料就翻记忆库，要分析问题就分步推理，甚至还能自己写代码。

可以看到，Agent 类似人的大脑的思考能力和四肢的执行能力。有了这些能力，Agent 可以被认为是一种类人智能体。

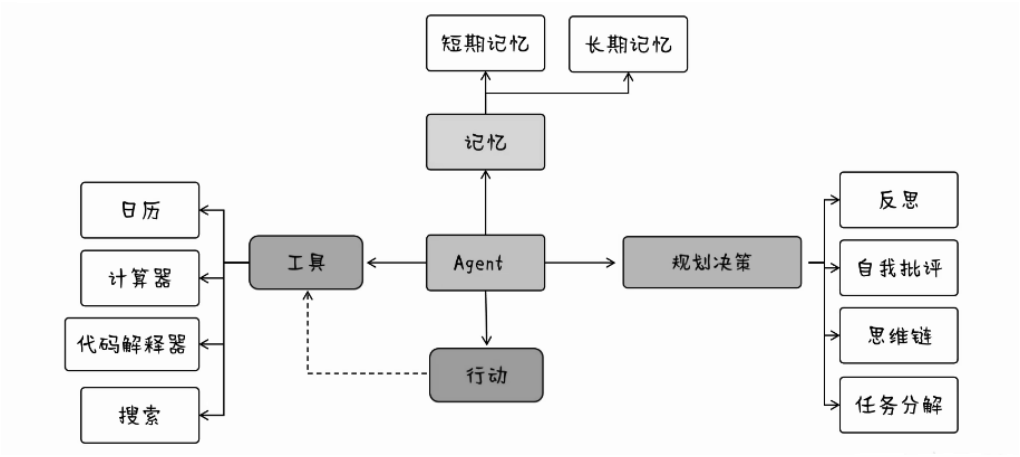

图 1-1　Agent 的组成部分

可以用一个不太恰当的比喻来说明：大模型（LLM）就像是人的大脑，而 Agent 则是人本身。大模型只有输入输出功能，而 Agent 则包括大模型、规划、记忆和工具。以前，智能机器人无法“理解”人类语言，但随着 AI 大模型的发展，它们开始“理解”人类语言，这使得 Agent 的能力得到了显著提升。未来，Agent 将在各个领域发挥重要作用，日益改变我们的生活和工作。

2. Agent 每个模块的作用

我们用一个管理花园的园丁的例子来说明组成 Agent 的每个模块的详细作用。

（1）LLM（大模型）：就像园丁的智慧和知识库，他阅读了海量的园艺书籍和资料，不仅知道各种植物的名字，还懂得如何照顾它们。在 AI Agent 中，LLM 提供了庞大的信息存储和处理能力，以理解和响应我们提出的各种问题。以 GPT 为代表的大模型的出现，将 Agent 的理解处理能力提高到了前所未有的高度。

（2）Planning（规划决策）：Agent 将大型任务分解为更小、可管理的子目标，从而能够有效处理复杂的任务，正如园丁需要规划整个花园的布局。AI Agent 的规划功能就像园丁制定种植计划，决定先种哪些花草，后种哪些蔬菜，或者如何分步骤修剪树冠。Agent 可以对过去的行为进行自我批评和自我反思，从错误中吸取教训，并针对未来的步骤进行完善，从而提高最终结果的质量。Agent 像人类一样一步一步思考，一步一步推理，以保证最后结果的正确性。

（3）Memory（记忆）：在与朋友沟通的过程中，我们需要记住沟通的上下文，但对于时间久远的对话，我们可能会记不住对话过程。短期记忆就类似于对话现场记下来的内容，而长期记忆则类似于把久远的聊天过程整理成一个记忆点，随时让大脑能够回忆当时说了什么重要的事。这类似于园丁的笔记本，记录了每个植物的种植时间、生长情况和前一次施肥的时间。记忆模块让 AI Agent 能记住以往的经验和已经完成的任务，确保不会重复错误。

（4）Tools（工具）：就像园丁的工具，比如铲子、水壶和剪刀。AI Agent 的工具模块，指的是它可以运用的各种软件和程序，帮助它执行复杂的任务。这些外部工具包括上网查询信息、代码执行、调用外部 App 等能力，就像园丁用工具进行园艺活动一样。

（5）Action（行动）：Agent 基于规划和记忆来执行具体的行动。这可能包括与外部世界互动，或者通过调用工具来完成一个动作（任务）。

业界对 Copilot（智能助手）和 Agent（智能体）是否有区别有一定的争论。Copilot 这个术语源自飞行术语，意思是副驾驶员（Co-pilot）。在飞机上，副驾驶员是协助主驾驶员操作飞机的人。Copilot 在帮助用户解决问题时起辅助作用，例如 GitHub Copilot 是帮助程序员编程的助手，它更多地依赖于人类的指导和提示来完成任务。Copilot 在处理任务时，通常是在人为设定的范围内操作，比如基于特定的提示生成答案。它的功能很大程度上局限于在给定框架内工作。

Agent 更像一个主驾驶，可以根据任务目标进行自主思考和行动，具有更强的独立性和执行复杂任务的能力。Copilot 主要用于处理一些简单、特定的任务，更多是作为一个工具或助手存在，需要人类的引导和监督。Agent 能够处理复杂、大型的任务，并在 LLM 薄弱的阶段使用工具或 API 等进行增强。

1.3　Agent 与大模型的关系及应用领域

众所周知，Agent 的大脑是大模型。大模型作为生成式人工智能的代表，其在推理分析、任务规划等方面显示出了一定价值，自然也为智能体的决策分析环节提供了新的动力。2024 年 5 月 15 日，火山引擎正式发布了豆包大模型家族。凭借更强的模型能力、更低的应用成本和更易落地的解决方案，豆包大模型在各行各业都得到了广泛的应用，其日均调用量也在高速增长。2024 年 5 月豆包大模型刚推出时，该模型的日均 tokens 调用量为 1200 亿，到 7 月份时涨到 5000 亿，到 9 月份涨到 1.3 万亿，截至 2024 年 12 月 15 日，已突破 4 万亿。豆包大模型调用量的高速增长，是大模型市场快速发展的一个缩影。

大模型为 Agent 的发展注入了强大动力，它使得 Agent 能够突破传统的性能瓶颈。从大模型到 Agent，是 AI 真正走向落地应用的关键一步。DeepSeek 的横空出世大大加速了 Agent 的落地速度，一场深刻的科技变革悄然展开。从智能体技术市场发展来看，Agent 吸引了海量资本涌入。各大科技巨头纷纷布局，初创企业也如雨后春笋般蓬勃生长，力求在这片新蓝海中抢占先机。市场咨询机构 Gartner 将 Agent 列为 2025 年十大战略技术趋势之首。业界认为，2025 年有望成为 Agent 的商业化应用元年。

随着“Agent+大模型”的潜力逐渐显现，国内外科技巨头纷纷敏锐地捕捉到这一机遇，开始悄然布局。在国内，字节、阿里、腾讯等科技巨头积极探索与大模型 DeepSeek 的合作，试图将其技术融入自身的业务体系中。例如，在电商领域，利用 Agent 和 DeepSeek 提升智能客服的服务质量；在云计算领域，借助 DeepSeek 的高效模型，为企业提供更强大的人工智能计算服务。

当技术发展到某一阶段时，往往会展现出迅猛发展的势头。毋庸置疑，“Agent+大模型”已经展现出巨大的潜力。可以说，2023 年到 2024 年的主流趋势是训练强大的大模型，因为 AI 应用的前提是得有个靠谱的“大脑”；而 2025 年开始的风口就是关注 Agent 方向，因为我们有了靠谱的大脑，如 DeepSeek、豆包大模型等，现在要做的就是完善 AI 应用落地场景。市场上开始完善 Agent 开发平台，例如火山引擎的扣子 AI 应用开发平台等。开发平台的完善加上基础 AI 能力的提升，才有可能实现 Agent 应用场景的落地。

1. Agent 和大模型的关系

1）Agent 和大模型的角色定位相互补充

大模型作为智能中枢，通过海量数据训练形成多模态处理能力，可解析文本、图像、语音等输入并生成上下文理解；Agent 则作为执行实体，基于大模型的输出进行决策和行动，两者形成“大脑”与“肢体”的协作体系——大模型提供认知能力，Agent 实现物理或数字世界的功能落地。

2）Agent 和大模型在不同层次上相互协作

- 感知与理解层：大模型处理原始输入（比如用户指令、环境数据），生成结构化任务目标及策略建议。
- 决策与执行层：Agent 根据大模型输出的上下文提示，结合预设规则（比如行业标准、安全限制）和实时反馈动态调整行动路径，确保任务不偏离目标。
- 动态优化层：Agent 在行动中积累的数据可反哺大模型，实现迭代升级（比如强化学习机制）。

3）Agent 和大模型在功能扩展上相互依存

- 大模型需要 Agent 实现场景化应用：大模型的通用能力需要通过 Agent 对接具体业务场景（比如半导体制造流程优化、代码生成），才能转换为实际生产力。
- Agent 依赖大模型提升智能水平：Agent 的决策质量直接受大模型理解能力的制约，例如复杂任务拆解、跨领域知识调用等均需大模型支撑。

4）Agent 和大模型在技术实现的关键要素上相互配合

- 规划：大模型提供任务拆解逻辑与优先级建议，Agent 结合资源约束生成可执行计划。
- 记忆：大模型提供存储通用知识库，Agent 管理短期会话数据与长期业务特征。
- 工具：大模型生成 API 调用代码或插件使用指令，Agent 调用外部接口/设备完成物理操作。
- 交互：大模型生成自然语言反馈，Agent 实现多模态人机交互界面。

2. Agent 的应用领域

Agent 技术未来将应用于多个领域，以下是一些典型的应用场景：

（1）教育辅导：Agent 可以作为个性化学习助手，根据学生的学习进度和兴趣提供定制化的辅导。通过分析学生的学习数据，Agent 可以识别学生学习的薄弱环节，并推荐相应的学习资源和练习题目。

（2）日常办公：在日常办公环境中，Agent 能够处理日常文档、安排会议、管理日程等任务，大大提高了办公效率。例如，Agent 可以帮助你处理邮件、安排日程，并提醒你重要事项的截止日期。

（3）推荐领域：在电子商务和内容推荐领域，Agent 能够分析用户的行为和偏好，提供个性化的推荐。例如，在购物网站上，Agent 可以根据用户的浏览历史和购买记录，推荐相关产品，从而提高销售额。

（4）医疗诊断：在医疗领域，Agent 可以辅助医生进行疾病诊断和治疗方案推荐。通过分析大量的医疗数据，Agent 可以帮助医生更准确地诊断疾病，并给出个性化的治疗方案。此外，Agent 还可以用于患者资料的处理、疾病趋势的预测以及个性化医疗建议的提供。

（5）客户服务：在客户服务领域，Agent 通过自动化处理大量的客户咨询，显著提升了服务效

率和顾客满意度。智能客服机器人能够 24 小时不间断提供服务，通过自然语言处理（NLP）技术理解用户需求并给出准确回答。此外，Agent 还能够基于用户历史数据提供个性化的服务推荐。

（6）股市交易：Agent 在股市交易领域也发挥着重要作用。它可以分析复杂的市场数据，为投资者提供基于数据的决策支持。通过学习大量的交易模式，Agent 能够识别出潜在的交易机会，并给出买卖建议。

（7）智能交通：在智能交通领域，Agent 被广泛应用于自动驾驶车辆和交通管理系统中。它能够实时感知道路情况并作出驾驶决策，包括车辆导航、避障、车道保持以及速度控制等功能。通过持续学习和优化，Agent 在提升驾驶安全性和舒适度方面展现出巨大的潜力。

（8）生产制造：Agent 可以自动化处理各种复杂的流程任务，如生产调度、库存管理和物流优化，从而提高整体效率和准确性。

在大模型时代，Agent 技术无疑是 AI 领域的一颗璀璨明珠。通过结合大模型技术，Agent 具备了更强大的语言理解与生成能力、决策能力和适应性，使其在各个领域中的应用更加广泛和深入。笔者相信，未来 Agent 会在各个方面影响我们的生活和工作。

另外，随着低代码、无代码开发理念的持续普及，低门槛易用 Agent 开发平台将吸引越来越多的非专业开发者和中小企业参与其中。更多的企业会利用其搭建适合自身业务的智能体工作流，降低开发成本和技术门槛，实现业务流程的智能化升级，从而推动整个智能体市场的进一步繁荣和发展。

1.4　Agent 开发者如何入局

Agent 的爆火表明了 AI 在当前 IT 领域的热度和潜在价值，而且还预示着未来可能会有更多资源投入这一领域。Agent 的核心思想就是给 DeepSeek、GPT、豆包等大模型配备工具和规划等能力，使它更像人。Agent 的需求量非常大，但真正懂行的人却不多，这意味着如果你拥有开发 Agent 的能力，你将拥有大量的工作机会。对于想要提升自己职业竞争力的开发人员来说，Agent 开发可能是一个不错的选择。

对于个人开发者和小微企业来说，面临的是前所未有的机遇。个人开发者和小微企业能够以更低的成本、更便捷的方式参与到 Agent 的开发中。借助 Agent 开发平台无须复杂编程基础即可搭建工作流的特点，他们可以根据自身的创意和业务需求，快速打造出个性化的智能体应用，并应用到诸如内容创作等领域，实现业务流程自动化和智能化转型，从而提升自身的竞争力。例如，自媒体创作者可以利用扣子开发智能体，高效地生成高质量的文章、视频脚本等内容，吸引更多的粉丝和流量；店主可以利用扣子开发智能客服，及时准确地回复客户咨询，提高客户满意度，推动客户购买行为。

对于企业开发者来说，可以在智能体开发平台提供的丰富功能基础上，开发出更复杂、更具创新性的 AI 应用。例如，通过整合不同的插件、优化大模型与代码块的交互逻辑，打造出适用于特定行业的智能体解决方案，甚至可以参与到智能体工作流相关的插件开发、生态建设中，拓展自身的职业发展路径。

第 2 章

扣子 AI 应用开发平台介绍

扣子是字节跳动公司面向用户提供的新一代 AI 应用开发平台。它基于火山引擎（也就是字节跳动提供互联网云服务的平台）和豆包大模型开发而成，可供用户定制开发各种 AI 应用，并为多种多样的 AI 应用场景提供解决方案。本章将介绍扣子 AI 应用开发平台的相关背景知识，并结合一个入门示例演示其简单用法。

2.1 扣子的背景与核心特征

扣子的英文名叫 Coze，这个名字大概同时匹配了其中文和英文的发音，也契合了扣子的目标功能定位（个人推测是从 Code 这个词变形而来，Code 代表代码，象征着开发，确实与扣子作为 AI 开发平台的功能定位非常贴合）。在扣子 AI 应用开发平台，可以通过零代码或低代码的方式快速搭建基于 AI 大模型的各类智能体应用。扣子平台上的智能体被称为 Bot，它可以是各种类型的聊天机器人。除了简单的对话外，通过扣子的插件和工作流等机制，还可以实现相对复杂的业务流程。

1. 扣子的背景

在 AI 大模型兴起的 2024 年，整个行业除了在讨论 AI、大模型这两个关键词外，另一个被广泛讨论的概念是 Agent。

很多业界人士将大模型的出现类比为当年的移动互联网。在移动互联网时代，应用的呈现形式是 App。而一个被广泛认同的观点是，AI 时代的应用呈现形式是智能体。无论是互联网大厂的高层还是行业专家，他们都同样认同这一观念。

AI 应用领域需要一个快速构建 AI 应用的开发平台。就好像移动互联网时代，行业提供了很多标准技术用于开发 App，也提供了应用商店用于 App 的分发。那么，AI 时代的应用开发平台是什么？应用分发的平台又是什么？

AI 时代需要有一个平台能够以更低的门槛帮助用户快速搭建 AI 应用，以及寻找各种各样的 AI

应用。因此，顺着这些思路，我们看到了像扣子这样的产品。可以说，扣子这类产品是 AI 时代背景推进下自然催生的产物。

2. 扣子的核心特征

扣子的核心特征主要有两点：首先，扣子是一个开发平台；其次，通过扣子开发的是 AI 应用，包括智能体和功能更全面的 AI 应用。例如，我们可以用扣子开发智能客服、个人助理、英语外教等智能体。我们在第 1 章中提到过，AI Agent 由 5 个关键部分组成，分别是大模型（LLM）、规划决策（Planning）、记忆（Memory）、工具（Tools）和行动（Action）组件。我们可以围绕这 5 个关键部分对扣子开发智能体的创建页面进行标注，如图 2-1 所示（此页面是第 13 章智能客服智能体编排页面，读者暂时不需要重现这个页面，只需理解智能体的 5 个关键部分即可）。

图 2-1　搭建智能体页面的 5 个关键要素

通过标注，读者会发现搭建一个智能体的过程，实际上也是在配置智能体 5 个关键要素的过程。例如，选择接入的 LLM 大模型；人设与回复逻辑，即 Prompt 提示词的填写，对应的是规划决策（Planning）；插件、工作流、图像流等技能，以及文本、表格等知识配置，对应的是工具（Tools）；变量、数据库、长期记忆、文件盒子等配置，对应的是记忆（Memory）；预览与调试对应的是行动（Action）。

可以看出，Agent 开发和传统软件开发不是一回事，无论你是否有编程基础，都可以在扣子上快速搭建基于大模型的各类 AI 应用。因为 Agent 更看重的不是软件编程能力，而是 Agent 的综合规划能力及流程理解能力。因此，对于非技术出身的伙伴，不要被开发平台吓怕了。我们日常的很多工作其实都涉及综合规划和流程理解能力，如果你在这方面有优势，开发 Agent 就会简单许多。

与其他智能体开发平台相比，扣子可以说是一个非常适合新手的平台。即使没有任何编程基础，用户也可以轻松创建自己的智能体，并且创建的智能体可以轻松对接并发布到多个平台，例如支持

扣子商店、豆包、飞书、抖音、微信等。同时，扣子还适配多个大模型，比如豆包、通义千问、DeepSeek、百川智能等。

作为一个全面集成的开发环境，扣子让每个人都能够轻松上手。无论用户的编程背景如何，借助扣子提供的可视化设计与编排工具，用户可以通过零代码或低代码的方式，快速搭建出基于大模型的各类 AI 项目，满足个性化应用开发的需求，从而实现商业价值。

2.2 选择扣子的理由

本节将介绍 AI 应用开发者的需求、AI 应用开发平台存在的挑战和难题，以及扣子的核心产品能力。从这 3 方面可以看出，扣子目前是最能满足 AI 应用开发用户需求的开发工具。

1. AI 应用开发者的痛点和需求

扣子产品的出现，主要用于解决如下的 AI 应用开发者的痛点和需求：

（1）用户需求多样且个性化难以满足：AI 的应用端发展目前才刚刚兴起，AI 应用的规模还不算大，远远无法满足用户丰富且个性化的需求，如果依靠企业端提供 AI 应用，短期内基本无法快速满足用户对大模型能力的需求。扣子的出现，让用户有办法根据自己的需求和问题，自己定义 AI 应用。

（2）AI 产品开发门槛高，成本投入大：对于很多企业或个人而言，自研开发一个 AI 应用是一件非常复杂的事情，包括产品设计、模型接入、应用开发等，需要投入比较多的资源并经历一定的时间周期。这对于大部分企业和个人而言，基本是无法实现的。扣子的出现，极大地降低了整个 AI 应用开发的难度和成本。

（3）AI 应用资源集成工作量大：开发一个 AI 应用需要多方面的技术能力，比如模型、搜索引擎、图像识别等。对于大部分企业而言，很多技术能力只能通过接入外部 API 来实现，无法全部自研，集成外部技术能力本身工作量巨大。

可以说，扣子类的应用的出现，主要就是为了解决以上所说的用户痛点和需求。

2. AI 应用开发平台存在的挑战和难题

虽然目前类似 AI 应用开发平台的产品越来越多，但是依然存在不少挑战和难题。

对于开发者而言，这些挑战包括：

（1）产品的使用难度比较高：对于完全没有编程和产品设计经验的业务人员来说，使用和上手还是太难，学习成本也比较高。

（2）开发能力的门槛高：智能体构建目前最关键的 3 个能力是提示词设计、工作流和插件，其中提示词设计和工作流设计对于初学者而言属于技术活，有一定的准入门槛。

（3）AI 应用的效果难以达到预期：目前普通用户开发的 Agent 在解决实际问题时，效果并不是那么让人满意，可能还不如直接使用 ChatGPT 这类现成的产品。

（4）智能体的商业变现模式不清晰：开发者难以投入比较多的时间打磨和迭代智能体应用。企业知识库是智能体商业化的关键，但是开发者缺乏专业、丰富、基于大模型驱动的企业知识库能力支持。

3. 扣子的核心产品能力

扣子的核心产品能力主要包括如下几部分。

1）灵活的工作流设计

扣子平台的工作流功能可以用来处理逻辑复杂且有较高稳定性要求的任务流。可以说，扣子平台如果没有工作流，那么扣子平台的价值将直接降低。AI 要解决真实世界的实际问题，不仅需要强大的大脑，还需要强大的分解任务能力。一个任务往往包含非常多的子任务，如果 AI 不具备一步一步拆解并分步完成任务的能力，那么这个 AI 只能作为闲聊的百科全书。扣子因为有了工作流这个功能，作为开发工具来说有了质的飞跃。

下发一个任务给 AI 后，它要完成这个任务，可能需要用到多种工具，我们只需要给 AI 配置不同的工具。扣子工作流的节点相当于不同的工具，而扣子提供了大量灵活可组合的工作流节点，包括大语言模型、自定义代码、判断逻辑、图像处理等。无论你是否有编程基础，都可以通过拖曳的方式快速搭建一个工作流。

假设某连锁品牌的旗下门店要做个主题活动，需要一张宣传海报。传统方法要完成这件事，门店要么申报总部，让总部设计部门设计下发，要么门店自己去打印店出钱设计。如果使用扣子工作流来实现，门店只需要告诉扣子要什么海报、内容是什么，设计好的工作流就能按照预设的流程、风格等信息，为门店自动生成一张宣传海报。在这个过程中，不仅不需要人工干预，还能确保设计标准符合总部要求。扣子平台提供的可视化拖曳的工作流编排功能，降低了普通用户使用的门槛。

2）无限拓展的插件能力

插件能力是构建 AI 应用必不可少的功能。我们都知道，大模型只是提供了一些类似文本生成的能力，但是大模型并不具备搜索引擎、网页内容获取等能力。在构建具体应用时，我们除了需要大模型的能力外，还需要多种构建产品工程的原子能力，这些就是通过插件来实现的，插件本质是各种 API 服务。扣子通过调用 API 实现各种功能，极大地拓展了智能体的能力边界。目前，扣子已经集成了上百种各式各样的插件，涵盖范围极广，包括图像、文本、搜索、数据分析、语音识别等，你可以直接将这些插件添加到智能体中。例如，使用新闻插件打造一个可以播报最新时事新闻的 AI 新闻播音员。你也可以利用已有的 API 能力，通过参数配置的方式快速创建一个插件让智能体调用。自定义开发的插件也可以发布到商店，供其他用户使用。

3）知识库的能力

先问读者一个问题：AI 有没有短板？有，还很多。最大的短板就是知识不属于用户，且学习的知识未必是最新的，例如刚刚发生的事，你问 AI，它就回答不出来。

那么怎么解决这个短板？答案就是：知识库。

扣子提供了强大的知识库功能来管理和存储数据，支持智能体与自己的数据进行交互。无论是内容量巨大的本地文件还是某个网站的实时信息，都可以上传到知识库中。这样，智能体就可以使用知识库中的内容回答问题了。知识库的作用是让模型获得并学习更多的专业知识，从而能够解决一些专业问题。在一些垂直应用场景，提供知识库是非常有必要的，因此知识库也是构建强大智能体的重要技能。目前，知识库支持本地文档、网页链接、笔记、在线文档、数据表、图片等格式的数据上传。另外，知识库还提供了多样化的检索能力。知识库这一功能特别设计用来解决大模型可

能出现的幻觉问题以及专业领域知识的不足问题，显著提升了大模型回复的准确性。

4）持久化的记忆能力

由于大模型存在上下文限制，大模型的记忆能力有限。为了让产品具备一些长期记忆能力，扣子提供了变量、数据库等功能。例如，扣子提供了方便 AI 交互的数据库记忆能力，可持久记住用户对话的重要参数或内容。创建一个数据库来记录阅读笔记，包括书名、阅读进度和个人注释。有了数据库，智能体就可以通过查询数据库中的数据来提供更准确的答案。

传统软件开发中的数据库是一门专业学科，例如 Oracle、MySQL、SQL Server 等。如果放在过去，普通人想利用数据库解决数据管理问题，可以说是不可能的事。然而，在 Coze 的 AI 加持下，数据管理就不再是难事了。上面提到的各种数据库，都可以不用懂，我们只需要用自然语言（口语化）告诉 AI 要记录什么，它就能帮你转换成数据库的语言进行记录，至于数据库内部是如何运行的，我们无须关心。

5）大模型能力

目前扣子平台提供了字节内部的豆包大模型服务，同时也提供了阿里通义千问、Kimi、DeepSeek、百川智能、智谱等第三方模型服务，用户可以根据自己的需求选择接入相应的大模型版本。扣子平台全面支持 DeepSeek 系列模型，用户在扣子智能体或工作流模型节点的模型列表中，能直接选择 DeepSeek 的 R1 或 V3 模型，感受其强大的功能。

6）Prompt 提示词编排能力

Prompt 提示词编排能力是用户创建智能体最基础的技能。很多智能体的能力，基本通过精心的编排和设计提示词，就可以达到相对较好的生成效果。扣子提供了提示词优化功能，并且通过一些专业提示词写作的引导，帮助没有掌握提示词创作技巧的用户快速掌握这一技能。

7）多 Agent 协作能力

在扣子平台上，除了单 Agent 模式外，还有多 Agent 模式，多 Agent 模式可以更加全面地处理更复杂的工作。

单 Agent 模式的优点是简单直接，但可能在处理复杂任务时效率不高，因为单 Agent 模式设定的智能体通常只能解决单一问题。例如一个图片处理智能体，如果想让这个智能体既能处理文本，又能处理图片，虽然并非不可能实现，但目前的 AI 容易出现工作任务理解不清而导致错误的问题。

那么，如何更好地解决这种多任务组合的问题？

答案是使用多 Agent 模式。多 Agent 模式允许多个 Agent 协作完成复杂任务，每个 Agent 都可以根据特定的提示和技能独立操作。这些 Agent 通过设置的“跳转条件”相互连接，基于特定关键词或用户请求在不同 Agent 之间切换，以高效完成复杂的任务流程。

我们可以先制作多个单 Agent，每一个 Agent 都只负责某一个具体工作，例如负责产品介绍的智能体、负责处理售后的智能体、负责品牌宣传的智能体。将这些智能体统一放在多 Agent 模式下，作为一个个独立的“员工”。在此基础上，配置一个管理者智能体，让它根据用户提出的问题进行判断，识别出这个问题到底是谁的工作内容、谁来服务客户等。管理 Bot 识别好任务后，就能把具体任务分配给具体智能体，从而更加高效地完成任务。

多 Agent 的优势如下：

- 统一管理多个智能体，让它们协同工作，避免工作内容冲突。
- 避免多个入口，如果没有多 Agent 模式，制作海报需要单独找到制作海报的 Bot，写文案要单独找写文案的 Bot。但是有了多 Agent 模式，只需要一个入口，就能把这些功能集合起来，发挥 AI 最大的功效。
- 在多 Agent 模式下，不仅能够发挥单个 Agent 的效果，还能单独调用数据库、插件等。这种模式的灵活性和扩展性，极大地提升了工作流的效率和 Agent 的功能，适合需要多功能处理和高度定制的 AI 应用场景。

2.3　扣子的版本和商业化模式

1. 扣子的版本

扣子提供基础版与专业版，界面和设计流程都是一样的。扣子基础版面向尝鲜体验的个人和企业开发者，全部功能免费使用，但有一定的限量额度，超过后不可再使用。例如，基础版提供有限的模型使用权限、有限的智能体调用次数，知识库空间默认只有 1GB 且不支持扩容；基础版每个工作流每天最多试运行 500 次，超出后会报错等。

扣子专业版在基础版的功能之上，支持更大的团队规模、更高的知识库空间以及更大的协作编辑容量，并提供专业完善的售后服务体系，满足开发者和企业用户的业务需求。扣子专业版面向对稳定性和用量有更高需求的专业开发者，支持更大的团队空间规模，使用更多的模型，提供更大的免费知识库空间，不限制调用请求频率和总量，费用按实际用量计算，不限制试运行工作流的次数。

2. 扣子是如何实现商业化的

扣子主要向应用的开发者收费，其具体的商业变现模式是：为开发者提供免费的有限功能和有限服务的版本，同时提供更高级的功能和更多的服务来收费。其中，官方将免费的版本称为“基础版”，将付费的版本称为“专业版”。该模式是比较典型的 SaaS（软件即服务）增值付费模式。

扣子基础版给开发者提供的免费能力包括：基础的开发能力、有限的模型使用权限以及有限的智能体应用使用次数（用户通过豆包、扣子应用商店等平台使用智能体应用的次数）。

基础版的有限模型使用权限包括：只能使用扣子平台提供的有限模型，不支持接入火山方舟模型资源。

而扣子专业版则提供更高的能力。当基础版的功能无法满足开发者的需求时，开发者需要购买扣子专业版。以下几种情形下，开发者可以考虑购买专业版：

（1）基础的应用开发能力无法满足开发者的需求，例如开发者需要更高的团队空间、知识库空间以及更高的扣子 API 调用次数。

（2）需要使用更多的模型或更高的模型版本，扣子基础版只提供了几个有限的模型和版本，而火山方舟上还有更多丰富的模型和版本，开发者如果想要使用更多厂商的模型和版本，则需要购买专业版。在扣子专业版账号下搭建智能体和应用时，支持使用 Kimi、通义千问等免费模型，也可

以通过火山引擎方舟平台接入更多的模型资源，以满足不同的需求。

（3）智能体的使用量限制无法满足开发者的需求，例如开发者需要有更高的智能体使用次数，则需要购买专业版。否则，智能体使用量达到一定限制后，智能体将不可使用。

3. 扣子专业版的付费模式设计

以下为扣子的基础版和专业版的权益差异对比，从整个权益体系的设计来看，其中核心的付费项目主要是 Bot 的使用消耗，以及大模型的使用消耗。至于基础开发能力等权益，本身不产生付费，只是超过免费门槛之后就需要使用专业版。购买专业版之后，通过 Bot 使用量和模型使用量 Token 消耗来计费。

扣子专业版的计费模式便捷且灵活，其支持的计费模式包括按量计费、包年包月和资源包。你可以根据业务用量、并发峰值合理评估产品需求，选择合适的计费方式，最大化节约成本。

扣子专业版支持的计费方式如下：

- 按量计费：按照各计费项的实际用量结算费用，先使用，后付费，适用于业务用量经常有变化的场景。
- 资源包：预先购买扣子资源包，在费用结算时，优先从资源包抵扣用量，先购买，后抵扣。超出资源包抵扣额度的用量，自动转为按量计费，根据你的业务量，购买适合业务额度的资源包。资源包用于抵扣扣子专业版的各项消耗。相较于按量计费模式，扣子资源包的性价比更高。

使用扣子专业版的过程中，涉及的计费项目如下：

- 智能体调用：包括智能体开发者在内的任意用户向智能体发送的一次有效对话请求计为一次智能体调用。由扣子专业版统计智能体调用次数并收费。
- 知识库空间：每个扣子专业版主账号可免费享有 10GB 知识库空间，支持以包年包月的方式额外购买知识库空间容量。
- 模型：使用大模型（例如豆包）的调用费用。按模型的 Token 使用量计费，不同模型的收费标准有所不同。
- 实时音视频：涉及实时音视频通话或流媒体传输的费用，按使用时长计费。
- 智能语音：包括语音合成、语音识别等费用，语音合成按字符数计费，语音识别按时长计费。

扣子资源包支持的抵扣范围包括：按量计费产生的智能体调用次数、超额使用的知识库空间容量、模型调用费（不包括微调模型）、实时音视频调用费、智能语音调用费。用户在调用智能体时产生的模型调用费，主要根据模型的 Token 使用量来收费。那么，这里的 Token 是什么意思呢？Token 是用来计量大模型输入、输出的基本单位，也可以直观地理解为“字”或“词”。但是，目前并没有统一计量标准，比如有的大模型 1 Token 约等于 1 个汉字，而有的大模型 1 Token 约等于 185 个汉字。简单来说，Token 的作用就是把文字拆分成模型能理解的小片段。例如，你写的“你好，老宋！”会被拆成“你”“好”“，”“老”“宋”“！”这些小块。Token 在大模型中的作用很大，模型通过 Token 能理解你写的文字，处理复杂语言，并更快地处理文字。

2.4　扣子平台的目标群体

扣子的产品定位是即使没有编程基础的用户也可以使用的 AI 应用开发平台。因此，其主要目标群体自然包括具有编程能力的研发人员群体，以及有 AI 应用开发需求的非程序员群体。然而，考虑到其使用目前仍存在一定门槛，个人认为，目前扣子的主要目标客户群体包括 B 端企业开发者和 C 端个人开发者。

1. B 端企业开发者

（1）中大型企业的研发部门和数字化降本增效团队：其核心诉求是利用扣子的能力为企业提供提升效率的 AI 应用工具。扣子的核心吸引力在于可以减少他们的开发成本，同时满足他们多种多样的业务需求。

（2）创业公司和中小企业：扣子可用于快速搭建提升创业公司和中小企业生产效率的 AI 应用工具。作为提升经营和业务效率的工具，扣子可以帮助企业解决智能客服、获客转换等问题。

2. C 端个人开发者

（1）关注 AI 的程序员群体：特别是 GitHub 圈内喜欢关注技术动态、研究新兴技术领域的技术人员，以及有提升研发效率需求的个人开发者。

（2）AI 科技爱好者：对 AI 科技领域有浓厚兴趣的用户，愿意钻研和了解 AI 工具和产品，属于 AI 深度爱好者。

（3）AI 效率需求群体：有提升个人办公、学习、生活、工作等效率需求的群体，包括办公白领、大学生等。

2.5　扣子平台架构

扣子为 AI 应用开发人员提供了一站式全链路的能力，如图 2-2 所示。扣子平台设计了一个空间的概念。在空间中，项目开发分为智能体和 AI 应用两类。资源库则包括插件、数据库、提示词、知识库等。

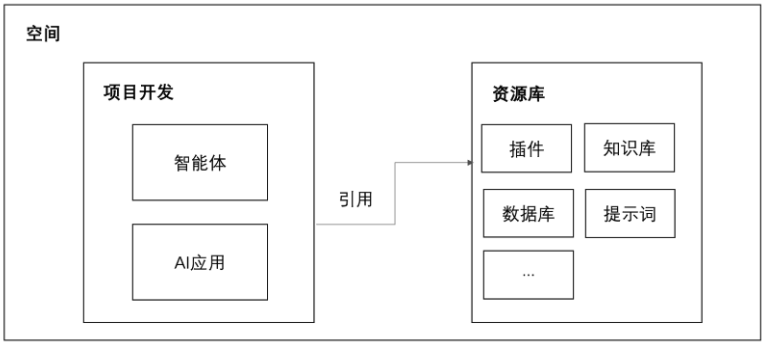

图 2-2　扣子平台架构

- 空间：空间是扣子平台中资源组织的最顶层单元。在不同的空间内，资源和数据是相互隔离的。一个空间内可创建多个智能体和 AI 应用，并包含一个资源库。在资源库中创建的资源可以被相同空间内的智能体和 AI 应用所使用。
- 项目：项目分为智能体和 AI 应用两种类型。
- 智能体：智能体（Agent）是一个能够独立完成任务的自动化程序。它可以根据用户指令自主调用模型、知识库和插件等，最终完成任务。例如，一个智能体可以成为虚拟助理，帮你安排日程。在基于大模型的智能体中，大模型充当着智能体的“大脑”的角色，同时还有 4 个关键部分：规划决策（Planning）、记忆（Memory）、工具（Tools）、行动（Action）。智能体的交互均是基于 LUI（Language User Interface，语言用户界面）的。
- AI 应用：AI 应用是基于大模型技术开发的应用程序。它能够处理复杂任务，分析数据，并做出决策。例如，AI 应用可以用来进行实时翻译或数据分析。AI 应用通常提供 GUI（Graphical User Interface，图形化用户界面）交互。
- 资源库：你可以在资源库内创建、发布、管理共享资源，例如插件、知识库、数据库、提示词等。这些资源可以被同一空间内的智能体和 AI 应用所使用。
- 资源存放位置的两种形式，一个是空间的资源库，另一个是 AI 应用中的项目资源库。
  - 空间资源库：在空间资源库内创建的资源可以被空间内的 AI 应用项目和智能体项目使用，属于空间内的共享资源。
  - 项目资源库：在 AI 应用项目中也可以创建资源，但这些资源是 AI 应用项目自有的资源，默认不可以被其他项目使用，也不会展示在空间资源库内。如果需要共享，可以将这些资源转移或复制到空间资源库。

2.6 扣子快速开发入门

2.6.1 设计思路

本入门案例非常简单，初学者容易上手。扣子平台旨在助力用户迅速搭建基于大模型的各类智能体，包括最近火爆出圈的 DeepSeek 模型。扣子现已推出满血版 DeepSeek 全家桶，支持免费体验 R1、V3 模型。此外，扣子支持 Function Calling（工具调用），让大模型拥有智能化调用工具的能力，为你的智能体添加私有知识和多种技能，极大地拓展智能体的能力边界，一个应用就能满足多种场景需求。

众所周知，智能体是基于对话的 AI 项目，它的交互均是基于 LUI 的。AI 应用由工作流串起来，工作流就是把 AI 应用的逻辑分成几个步骤，每个步骤称为一个节点。本案例构建的示例工作流节点如图 2-3 所示，除了开始和结束节点外，这个工作流中只添加了一个大模型节点来处理用户任务。你可以将用户输入的内容传输给大模型进行处理并返回结果。包含大模型节点的工作流可单独指定模型的各项配置参数，通过附加的提示词约束模型的行为，使智能体在指定场景下的运行过程更稳定、输出内容更符合预期效果。

图 2-3　工作流架构案例

2.6.2　接入 DeepSeek 大模型

扣子平台默认只能使用预置的模型，例如豆包模型。如果你需要使用其他模型，可以在火山引擎的火山方舟平台创建模型推理接入点。然后在扣子平台中创建智能体并为智能体设置模型，或者在工作流的大模型节点中进行模型选择，才可以在模型列表中选择模型推理接入点所对应的模型。

我们可以通过火山方舟平台接入 DeepSeek，流畅地使用 DeepSeek 满血版，并能感受到大模型的高性能和高稳定性。首先需要注册火山引擎账号，官网地址为 https://www.volcengine.com/。打开页面后，单击右上角的“注册/登录”（支持手机号或邮箱）。账号注册完成后，需要完成实名认证（对接模型需要先完成实名认证）。登录火山引擎官网后，如图 2-4 所示，通过搜索“火山方舟”，单击“管理控制台”按钮进入火山方舟管理控制台。

图 2-4　搜索“火山方舟”

如图 2-5 所示，在模型广场的搜索框输入 DeepSeek，作为字节跳动的云服务平台，火山引擎自然懂得跟上“潮流”，不仅提供豆包系列模型，还提供 DeepSeek 模型。

图 2-5　火山方舟的“模型广场”

如图 2-6 所示，我们在页面左下角依次单击“系统管理”→“开通管理”菜单，打开“开通管理”页面，在其中选择需要开通的大模型。例如，我们在页面右侧操作窗口单击“开通服务”，开通 DeepSeek-R1 大模型。注意，开通过程中有个实名认证，按它的要求进行操作即可。

图 2-6　火山方舟的“开通管理”

接着通过“火山方舟沟里控制台”页面左侧导航栏找到“在线推理”菜单项，打开“在线推理”页面，如图 2-7 所示。在“自定义推理接入点”选项页面中单击“+ 创建推理接入点”。

图 2-7　火山方舟的“在线推理”

在打开的页面上，输入接入点名称和接入点描述等相关信息，再单击“+ 添加模型”按钮，如图 2-8 和图 2-9 所示，在“选择模型”页面中选择 DeepSeek-R1 模型。

图 2-8　创建推理接入点

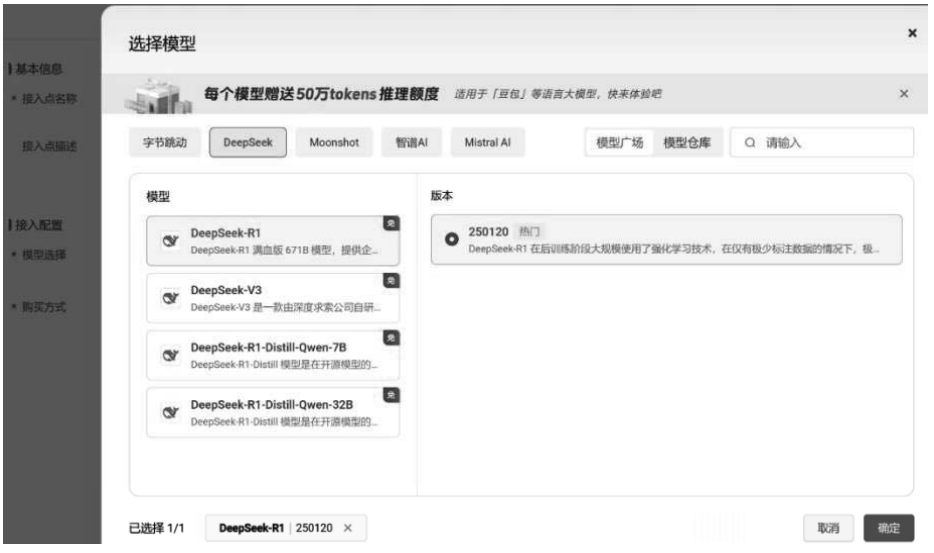

图 2-9　选择模型

如图 2-10 所示，购买方式选择“按 Token 付费”，再单击“确认接入”按钮。

图 2-10　确认接入推理接入点

回到“在线推理”页面，可以看到推理接入点已经创建完毕。如图 2-11 所示，页面上提供了接入点名称 deepseek-r1-20250120 和模型 ID（图中左上角 ep 开头的字符串）等信息。

图 2-11　接入点 API 调用信息

2.6.3　构建工作流

打开扣子官方网站，选择“工作空间”→“资源库”，在资源库页面上依次单击“+ 资源”按钮→“工作流”菜单项，如图 2-12 所示。

图 2-12　在工作空间“资源库”中单击“+ 资源”按钮创建工作流

打开的“创建工作流”窗口如图 2-13 所示，工作流名称输入“deepseek_tasks”，工作流描述输入“使用 DeepSeek 处理问题”。

单击“确认”按钮，将打开工作流编辑页面，如图 2-14 所示。后面我们在工作流上添加、配置节点等操作都是在这个页面上进行的，读者可以在扣子网站上熟悉一下这个页面。

图 2-13　工作流的基本信息

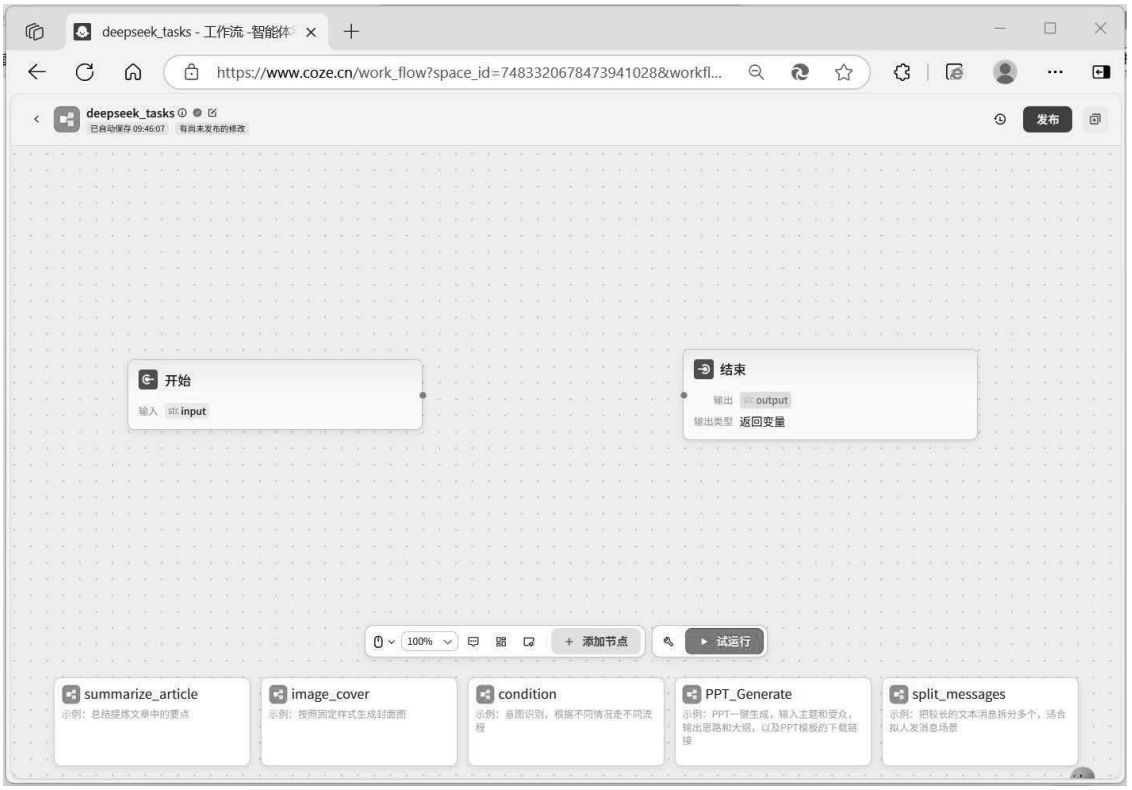

图 2-14　工作流编辑页面

每个工作流都有一个开始节点和一个结束节点，这两个节点不能删除，必须存在。开始节点定义输入参数（有哪几个输入参数，分别是什么类型，起什么作用），以便后续节点引用；结束节点定义输出参数（有哪几个输出参数，分别是什么类型，是什么意思）；而其他的节点就是处理过程，用来完成设计的功能。因此，实际上一个工作流完全可以看作一个函数，开始节点定义输入参数，结束节点定义输出参数，中间节点完成函数功能。

如图 2-15 所示，在开始节点的输入参数配置页面中，输入变量名 user_input，变量类型选择 Str.String 字符串类型。开始节点必须设定输入参数，这个界面对应机器人（智能体）对话界面，用来接收用户的输入。

图 2-15　开始节点配置

工作流接收到用户输入后，就要将输入交给大模型进行文本处理，所以开始节点后面需要增加一个大模型节点。如图 2-16 所示，在工作流编辑页面下方，单击“+ 添加节点”按钮，在节点类型

列表中选择大模型，工作流编辑页面上将增加一个大模型节点，如图 2-17 所示。

图 2-16　添加节点

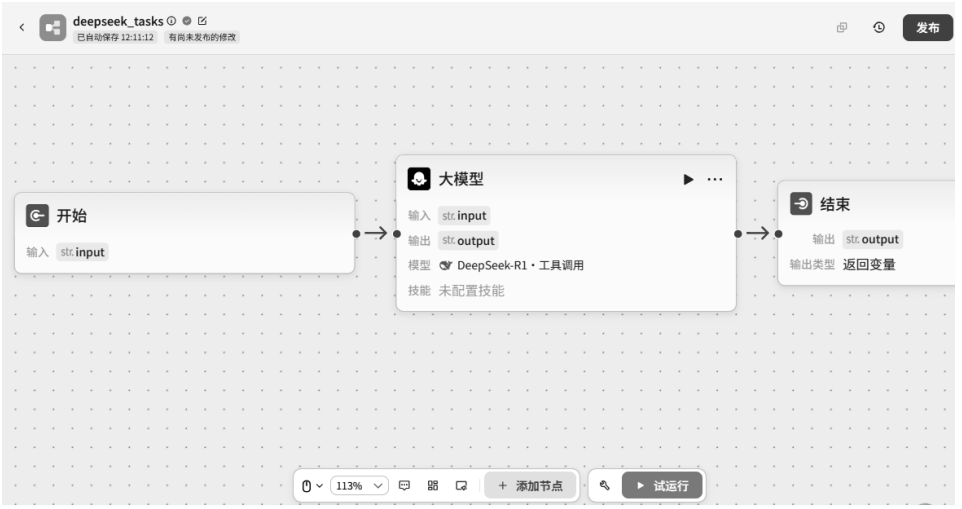

图 2-17　连接开始节点、大模型节点和结束节点

连接开始节点、大模型节点和结束节点，节点连接顺序是“开始→大模型→结束”。节点之间需要用连线表示先后关系，把鼠标放在图中的小蓝点上，按下鼠标右键拖出一条线，指向下一个节点的边框的小蓝点，就有一条线把这两个节点连接在一起，表示前后顺序关系。注意，节点只有连上线之后，后面的节点才能获取前面节点的信息。

单击大模型节点，打开大模型配置窗口，配置大模型节点，如图 2-18 所示。扣子提供了很多大模型供选择，在页面上选择模型时，就可以发现火山方舟提供的 DeepSeek-R1 满血版模型（接入点 deepseek-r1-20250120）已经接入进来了，我们选择 deepseek-r1-20250120。当然，扣子平台也会面向

用户统一提供自己的模型服务，如豆包通用模型。

图 2-18　大模型节点选择接入火山方舟的 DeepSeek

大模型节点配置窗口如图 2-19 所示，大模型节点的输入就是开始节点的输出，单击右侧的变量值输入框，就会打开一个下拉列表，选择 user_input，这就是开始节点中定义的变量 user_input。大模型必须配合提示词才能工作，且提示词的好坏影响最终的结果。注意提示词中的{{input}}表示用户输入的文本内容。然后给大模型设置输出变量 output，变量类型选择 Str.String 字符串类型。

图 2-19　大模型节点配置

结束节点配置窗口如图 2-20 所示，在工作流的结束节点中，新增变量名 output，参数值选择“大模型-output”，表示引用大模型节点的输出 output。我们可以体会到，从开始节点到大模型节点再到结束节点之间数据信息的流动。

图 2-20　结束节点配置

各节点的参数配置如表 2-1 所示。

表 2-1　各节点参数配置

节　　点	参 数 配 置
开始	输入：新增变量名 user_input，变量类型选择 Str.String
大模型	选择大模型节点的单次模式，模型选择 deepseek-r1-20250120。 输入：参数名为 input，变量值选择“开始-user_input”。 用户提示词：{{input}}，表示将用户输入的数据传入大模型进行处理。 输出：变量名为 output，变量类型选择 Str.String
结束	输出变量：参数名为 output，参数值选择“大模型-output”

配置完成后，单击工作流编辑页面（见图 2-17）底部的“试运行”按钮，测试工作流。如图 2-21 所示，用户输入“请以李白的风格写一首诗”进行测试，待所有节点都运行成功（节点会展示绿色边框）后，可以单击某个节点下方“运行成功”右边的“展开结果”，查看这个节点的运行结果。模型输出《江夜行》这首诗，读起来还真有李白的诗韵，果然是目前地表最强的大模型之一。

图 2-21　工作流试运行结果

测试工作流没有问题后，单击工作流编辑页面（见图 2-16）右上角的“发布”按钮，打开“发布”窗口如图 2-22 所示，输入版本号和版本描述。这个工作流成功发布后，再单击“工作空间”→“资源库”，在资源库页面上的资源列表中可以看到该工作流的名称，读者可以确认一下。

图 2-22　发布工作流

2.6.4　创建智能体

如图 2-23 所示，依次单击扣子平台的“工作空间”→“项目开发”，在“项目开发”页面上单击页面右上角的“+ 创建”按钮，弹出如图 2-24 所示的“创建”窗口，选择左边的创建智能体，单击“创建”按钮，创建对话式智能体。

图 2-23　单击“+ 创建”按钮

图 2-24　创建智能体

如图 2-25 所示，在“创建智能体”窗口中，输入智能体名称和智能体功能介绍的相关信息。

图 2-25　智能体基本信息

单击“确认”按钮，将打开智能体编排页面，如图 2-26 所示。在智能体编排页面（记住这个智能体编排页面和前面提到的工作流编辑页面，我们大部分工作都是在这两个页面上完成的）上，找到页面中间“技能”区域的“工作流”，在“工作流”右侧单击加号图标 + 添加工作流。

图 2-26　智能体编排页面（局部截图）

如图 2-27 所示，在添加工作流对话框的左侧单击“资源库工作流”，找到我们自建的 deepseek_tasks 工作流，并在 deepseek_tasks 右侧单击“添加”按钮。

图 2-27　添加工作流对话框（局部截图）

工作流添加进来后，我们可以直接试运行看看测试结果。在智能体编排页面右边的“预览与调试”区域，如图 2-28 所示，输入内容“给我一些学习人工智能的建议？”，预览一下智能体实现的效果。可以看到效果比较理想。

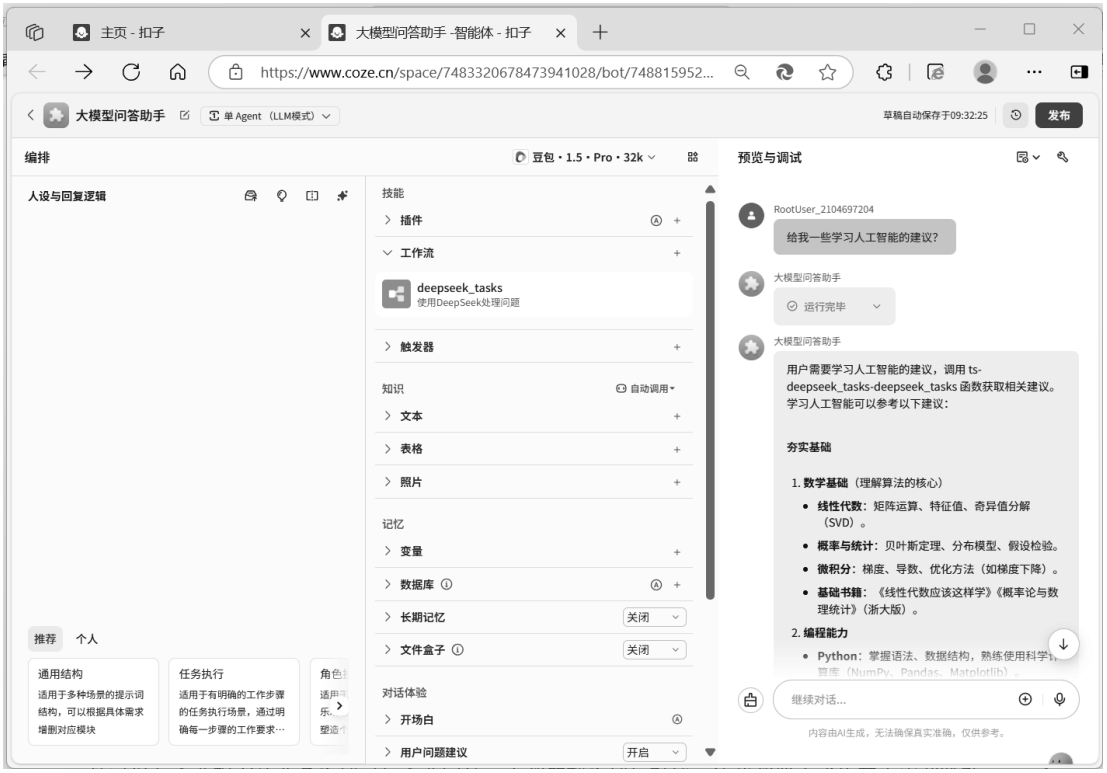

图 2-28　试运行结果

2.6.5　发布到微信公众号

扣子支持的发布渠道非常多，可以发布到抖音、飞书、钉钉、掘金社区、微信、微信公众号等还可以与企业现有的业务系统（如企业网站、内部管理系统）对接。这些部署方式均基于字节跳动提供的云端技术支持，用户无需自行搭建服务器即可完成跨平台功能扩展。这里，我们把智能体发布到微信公众号。单击智能体编排页面右上角的“发布”按钮，就会弹出发布界面。如图 2-29 所示，找到“微信订阅号”，这里状态是“未授权”，单击右侧的“配置”按钮。

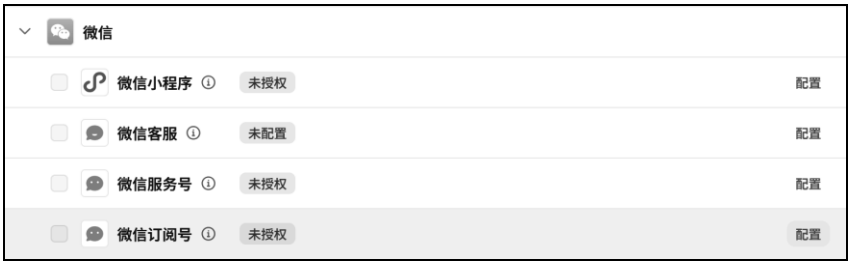

图 2-29　发布到微信订阅号

如图 2-30 所示，在“配置微信公众号（订阅号）”页面中，填写微信公众号中的“开发者 ID”，即 AppID，注意这个是微信公众号开发者 ID。

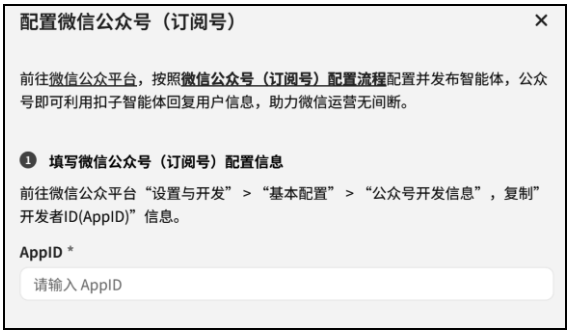

图 2-30　填写微信公众号信息

如图 2-31 所示，进入公众号管理后台，依次找到菜单“设置与开发”→“开发接口管理”。这里你必须先成为开发者，才能获得 AppID。

图 2-31　开发接口关联

如图 2-32 所示，单击“开发接口管理”→“基本配置”，即可看到开发者 ID（AppID）。复制开发者 ID 到图 2-30 所示的配置微信公众号（订阅号）页面，单击“保存”按钮。这时会出现一个二维码，如图 2-33 所示，用你的公众号主账号的微信扫一下二维码，授权一下即可。

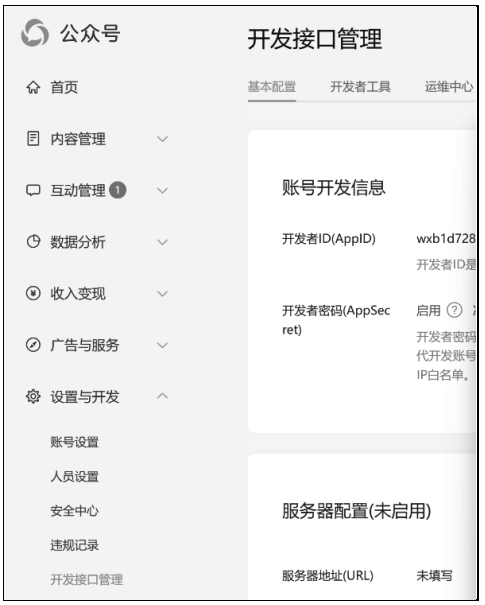

图 2-32　单击“开发接口管理”→“基本配置”

图 2-33　公众平台账号授权

微信订阅号配置成功后，单击“发布”按钮，再单击“完成”按钮即可。你可以进入公众号界面，输入几个问题测试一下。整体测试下来，回答效果符合预期，只是回复速度略慢一些。

最后还是要提醒读者注意，本节入门案例所讲解的操作步骤非常重要。完全掌握这个快速入门案例，有助于本书的顺利学习，避免因某个具体的操作细节而卡住。如果读者还没有对本节入门案例建立起整体印象并理解操作步骤，建议重新学习，并上网亲自动手实践一下。