beansmile

重塑AI应用开发范式

Dify.ai理念与落地实践

主讲人：Leif • 时间：2024.12

beansmile

Dify_实战指南

作者: Beansmile

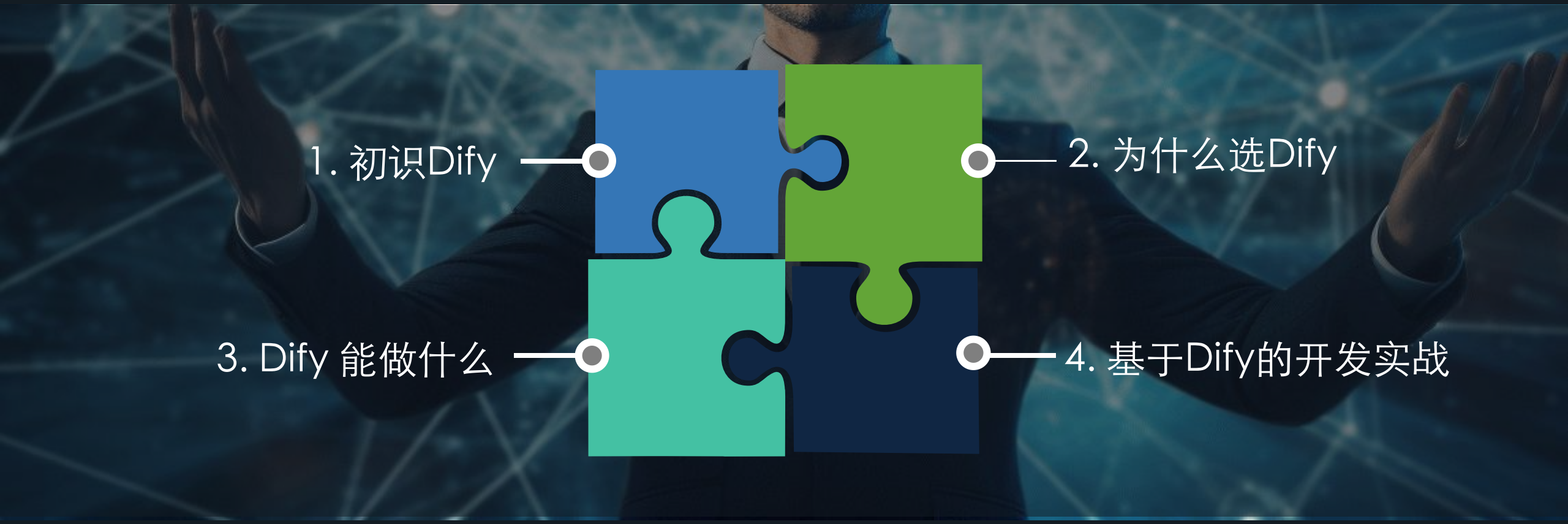
出版日期: 2024-11-05

描述: 在这本书中，我们将为你揭示 Dify 的实战之道。通过真实的项目案例，展示如何巧妙运用 Dify 加速开发进程，让 AI 的魔力自然而然地融入应用之中。

主讲人：Leif • 时2024.12

《Dify_ 实战指南》 Ebook 目录

beansmile



目录
CONTENTS

01

AI应用开发现状与痛点

AI 应用开发现状与痛点

OpenAI 出了个新模型，去适配下

这个原有系统需要增加一个AI 新功能，能不能明天上线

这个AI功能跟上一个差不多，调整下口令就好了，今晚能不能上线

开发者最大的痛就是： 你要的东西那么急，臣妾做不到啊

AI 应用开发现状与痛点

模型迭代适配

大模型快速迭代带来模型版本更新频繁、性能特性变化、接口参数调整以及兼容性维护成本高的问题。这些影响导致适配代码需频繁更新、测试成本增加和版本管理复杂。

接口标准化问题

接口规范不统一、参数命名不一致、返回格式差异大以及错误处理机制不同，导致差异性问题。这些问题使得代码难以复用、维护成本高、开发效率低。

多模型管理

集成复杂主要体现在需要对接多套接口、认证机制不同、计费模式各异以及性能特征差异。这些导致代码结构复杂、切换成本高和难以统一管理的挑战。

大模型应用开发现状与痛点

开发效率问题

效率瓶颈的表现包括重复开发工作多、通用功能需反复实现、缺乏统一开发框架和工具链不完善。这些问题导致开发周期长、人力成本高和质量难以保证。

业务适配

适配过程面临场景理解不足、业务逻辑复杂、定制需求多和集成难度大的问题。主要挑战包括需求理解困难、开发周期长以及维护成本高。

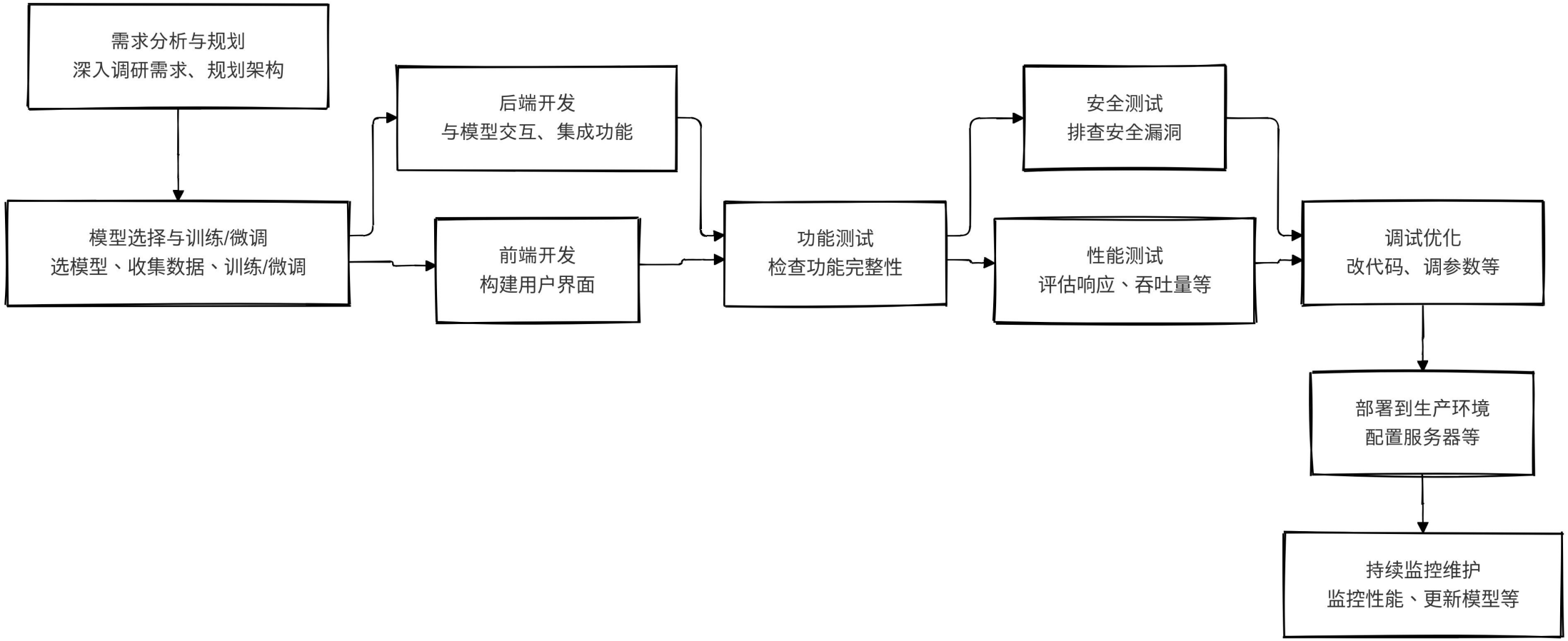
其他

.....

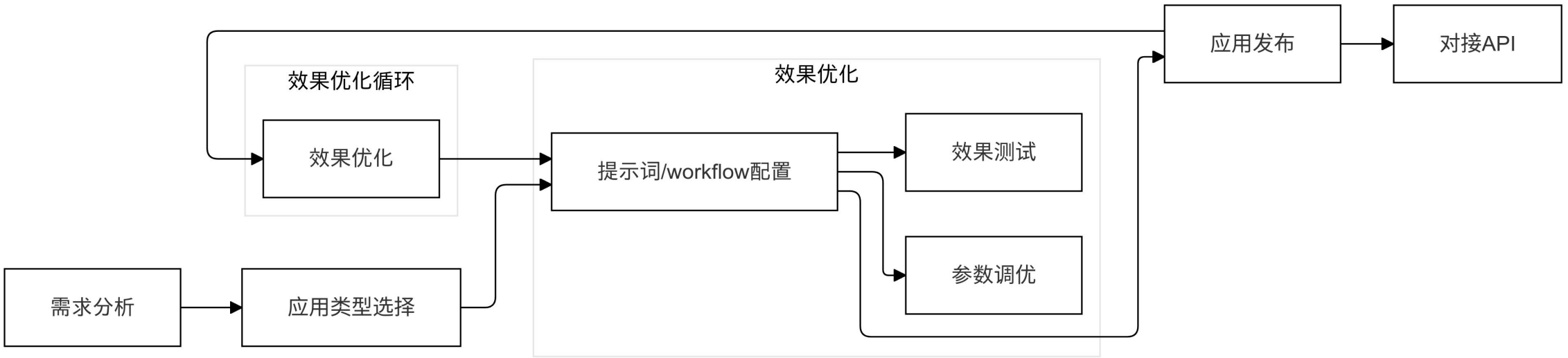
02

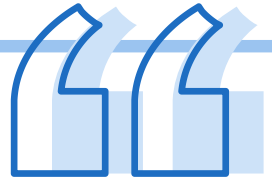
对比传统开发与Dify的开发新模式

传统大模型应用开发流程



Dify新模式下的开发





维度	传统开发	Dify 模式
开发周期	周	天
技术门槛	较高	较低
维护成本	较高	低
扩展能力	较高	高
模型集成与管理	复杂度较高	低
部署	取决于技术架构	简单

03

Dify 应用场景分析

常用场景分析

POC 快速验证

通过Dify，用户可以快速搭建和测试大模型应用，验证其在特定业务需求中的性能。

多模型效果对比

在处理复杂任务或数据时，经常需要对多种模型进行比较，以找到最佳解决方案。Dify允许用户同时运行多个模型，并根据预设指标进行效果评估。

企业级解决方案

Dify可作为企业内部的语言模型基础设施，支持将AI功能无缝集成到现有业务系统中，如CRM、供应链管理、企业资源计划（ERP）等。

数据分析

Dify提供了日志的标注与监测服务，可集成外部Ops工具对应用进一步分析。

beansmile

04

从理念到实践：某文旅项目

某文旅项目景点调研方案

某文旅项目附近景点调研

项目实现目标

通过对选址周边的景点进行信息搜集调研，然后对调研的数据进行结构化的输出

核心功能

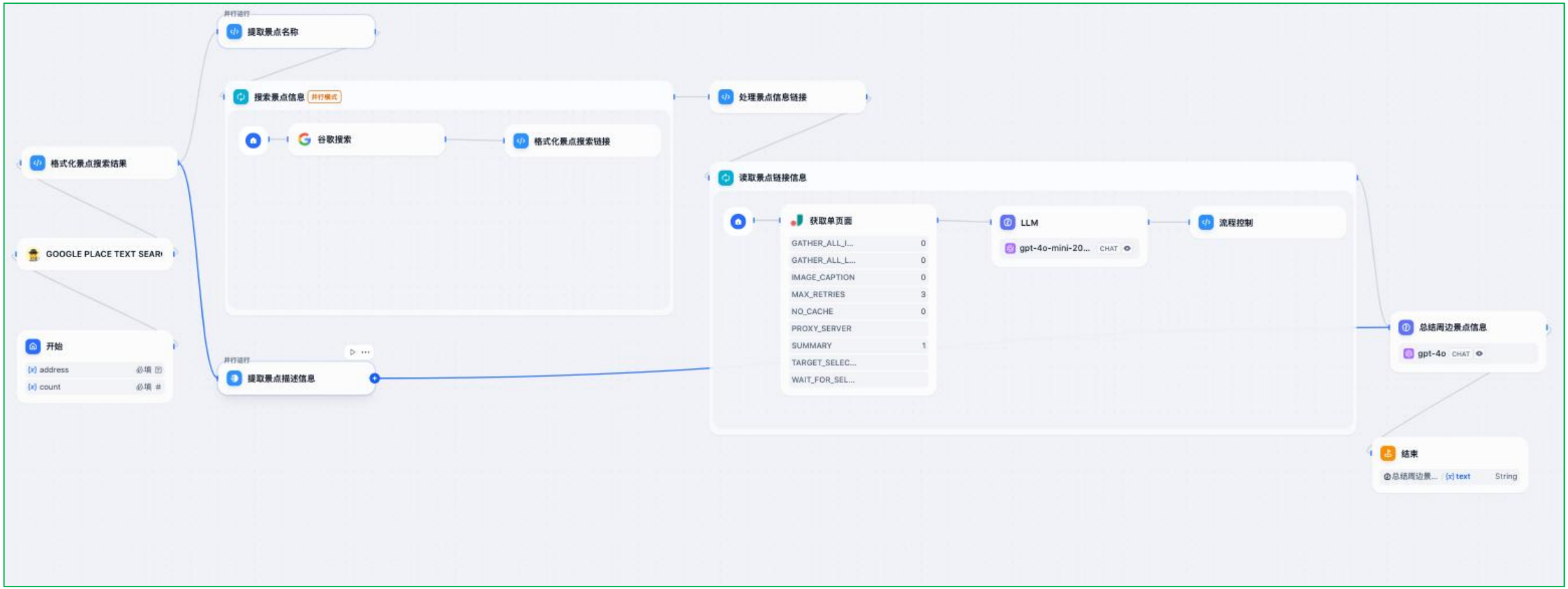
1. 通过 Places API 获取附近旅游景点地址与名称

2. 通过联网调用 Google 搜索景点信息

3. 迭代提取 所有景点信息的网页搜索到的信息

4. 使用大模型总结分析

Workflow 展示

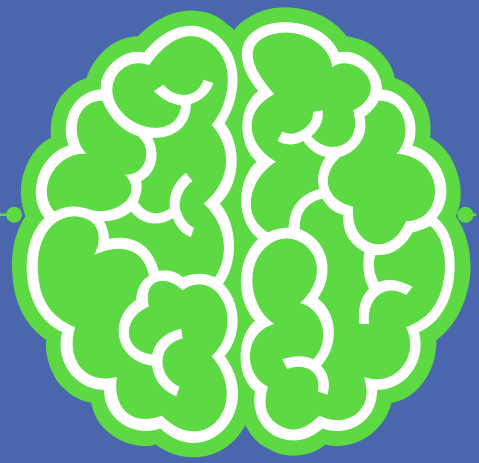

并行运行
提取景点名称
搜索景点信息
并行模式
谷歌搜索
格式化景点搜索链接
处理景点信息链接
格式化景点搜索结果
GOOGLE PLACE TEXT SEARC
读取景点链接信息
获取单页面
GATHER_ALL_I... 0
GATHER_ALL_L... 0
IMAGE_CAPTION 0
MAX_RETRIES 3
NO_CACHE 0
PROXY_SERVER
SUMMARY 1
TARGET_SELEC...
WAIT_FOR_SEL...
LLM
gpt-4o-mini-20... CHAT
流程控制
总结周边景点信息
gpt-4o CHAT
开始
address 必填
count 必填 #
并行运行
提取景点描述信息
结束
总结周边景... {x} text String

Workflow 主要节点

工具

迭代

模板转换



并行

代码执行

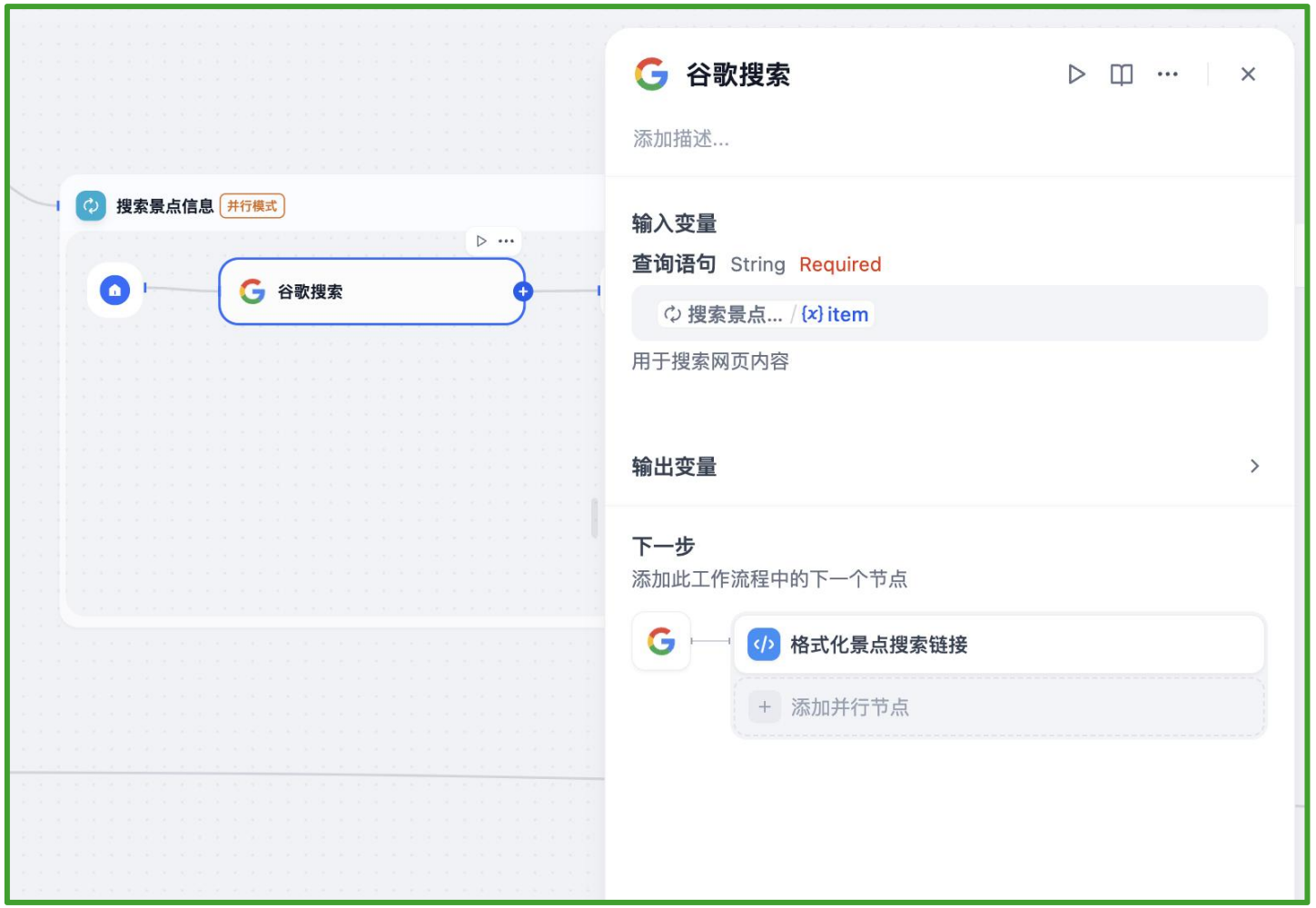
LLM

自定义工具

Dify 提供的自定义工具，需要遵循OpenAPI-Swagger 规范的。本案例，需要对某地点选址周边的景点进行调研，通过自定义工具，接入Google Place API，获取周边景点信息

怎么为API写一个满足OpenAPI-Swagger的Schema

遇事不决，可问 AI



内置工具

Dify 提供了很多的工具，例如抓取网页，google 搜索，调用SD画图等等。

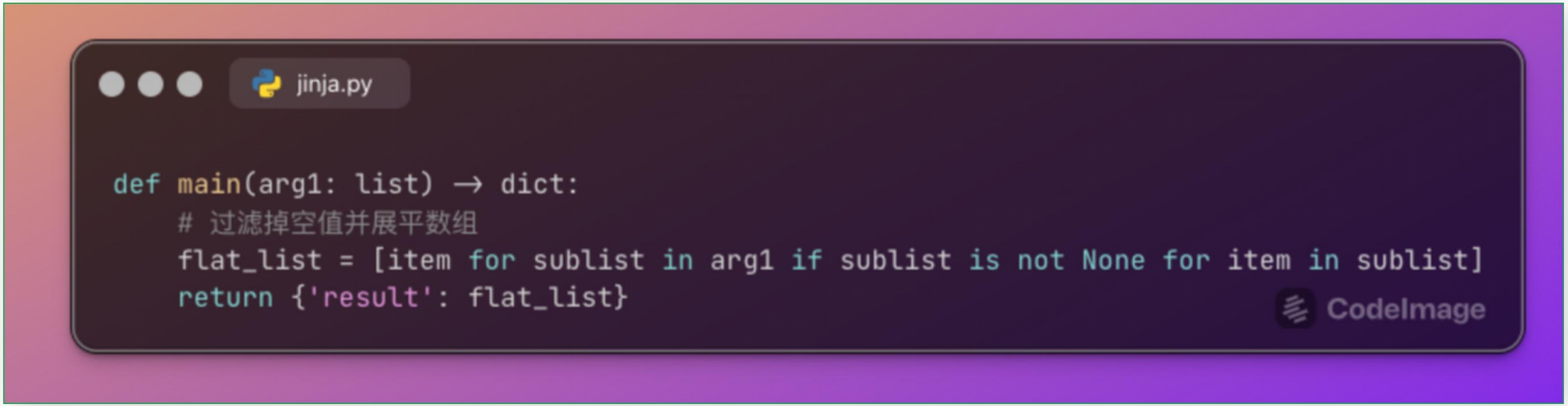
在本实例中，我们使用了google 搜索来搜索景点信息。

使用并行节点

Dify 提供四种并行的方法：
普通并行、嵌套并行、迭代中并行和条件并行。
在这个案例中我们使用的简单并行跟迭代中并行两种方式。

迭代

Dify 迭代节点，支持对数组对象进行迭代执行具体任务。



代码执行

在工作流中，经常要面对非结构化的数据处理，如JSON字符串的解析、提取、转换等。在这个里面我们就对数据进行处理，返回下几个迭代节点期待的一个一维数组。

Jinja模板转换

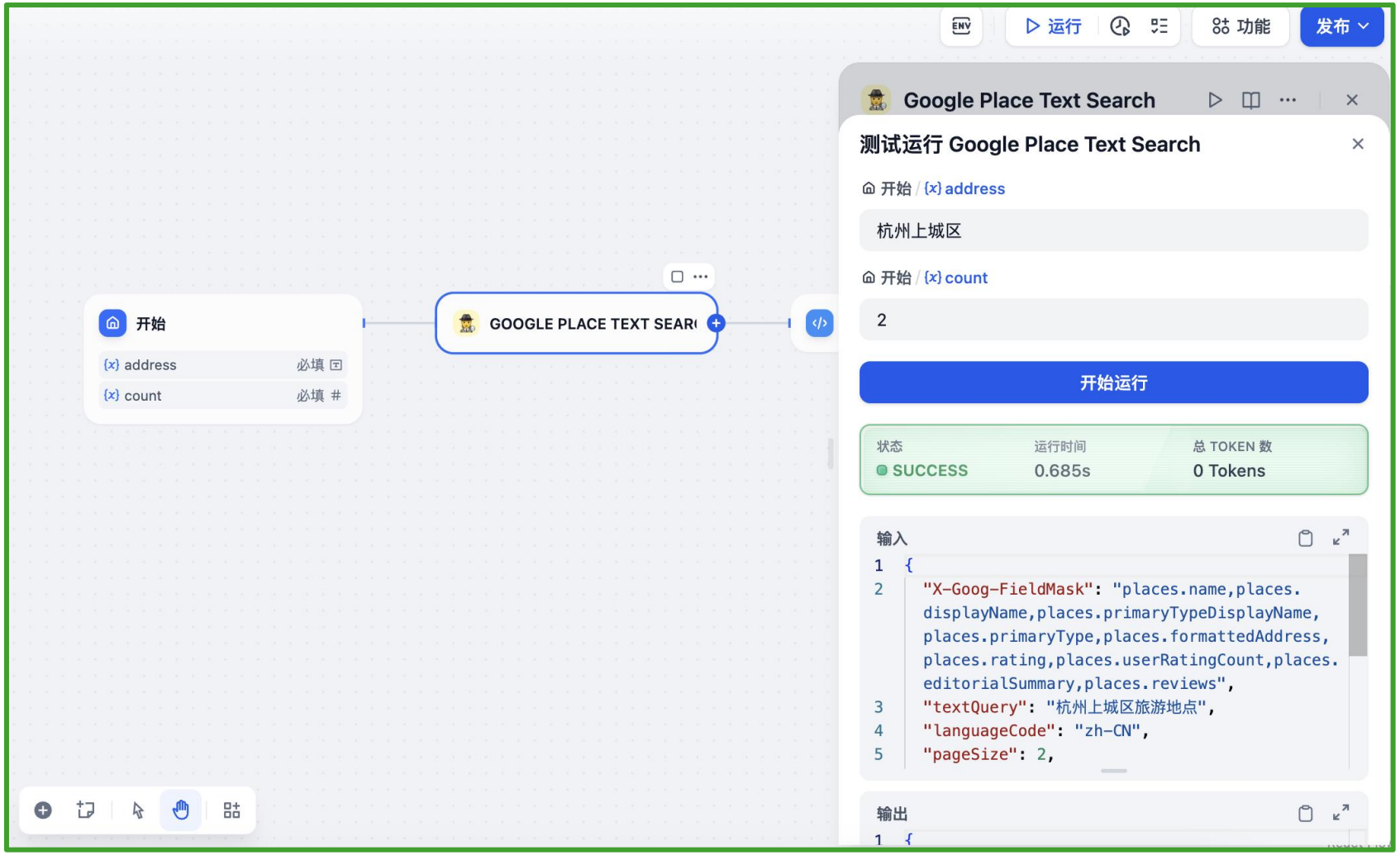
jinja.py

```
{% for place in arg1.places %}
  {% if place.displayName %}name: {{ place.displayName.text }},{% endif %}
  {% if place.formattedAddress %}address: {{ place.formattedAddress }},{% endif %}
  {% if place.rating %}rating: {{ place.rating }},{% endif %}
  {% if place.userRatingCount %}userRatingCount: {{ place.userRatingCount }},{% endif %}
  {% if place.editorialSummary %}editorialSummary: {{ place.editorialSummary.text }},{% endif %}
  {% if place.reviews %}reviews: {% for review in place.reviews %}
    {% if review.originalText %}rating: {{ review.rating }},review: {{ review.originalText.text }}{% endif %}{% if not loop.last %},{% endif %}
  {% endfor %}{% else %}{% endif %}
{% endfor %}
```

CodeImage

在此案例中，我们将json 数据 转成成文本，并将更语义的句子结果提交给大模型进行处理。 我么也可以根据自己需要转成 html 或者是 markdown 等等

单点调试



工作流支持节点的单步调试，在单步调试中你可以重复测试当前节点的执行是否符合预期。

单步测试运行后可以查看运行状态、输入/输出、元数据信息。

Test run

Dify workflow 的test run，有当前任务的详细运行情况，包括当前运行节点，每个节点的输入输出，错误信息等

beansmile

谢谢大家

主讲人：Leif • 时间：2024.12