

SpringAI联手GPT-4o：定制化面相分析师

一. 框架介绍

该项目从著名的 Python 项目（例如 LangChain 和 Llamaindex）中汲取灵感，但 Spring AI 并非这些项目的直接移植。该项目的成立基于这样的信念：下一波生成式 AI 应用将不仅面向 Python 开发人员，还将遍及多种编程语言。Spring AI 的**核心是提供抽象，作为开发 AI 应用程序的基础**。这些抽象具有多种实现，只需进行最少的代码更改即可轻松交换组件。

****功能特性：**

- 支持所有主流模型提供商，例如 OpenAI、Microsoft、Amazon、Google 和 Huggingface。
- **支持的基础模型类型包括聊天、文本到图像、音频转录、文本到语音等。**
- 支持主流的矢量数据库，例如 Apache Cassandra、Azure Vector Search 等。
- FunctionCall 函数调用能力的支持。
- 结构化输出的能力支持。
- 数据工程的 ETL 框架 (RAG)。

项目地址：[github.com/spring-proj...](<http://cxyroad.com/> "https://github.com/spring-projects/spring-ai")

二. 多模态特性

多模态是指模型同时理解和处理来自各种来源的信息的能力，包括文本、图像、音频和其他数据格式。

Spring AI Message API 提供了支持多模式 LLM 所需的所有抽象。

目前最新版本支持的模型列表：

- **Open AI – (GPT-4-Vision and GPT-4o models)**

- Ollama – (LlaVa and Baklava models)

- Vertex AI Gemini – (gemini-pro-vision model)

- Anthropic Claude 3

- AWS Bedrock Anthropic Claude 3

三. 多模态应用

3.1 照片面相分析

先看效果

分析结果：

根据这张照片的分析，这位男性可能正在一个公园或绿地进行锻炼。由于他手持水瓶，肩上搭着毛巾，身穿运动服装，可以推测他刚刚进行了一次运动或锻炼，可能是跑步或者其他有氧运动，并且正在休息和补充水分。

面相分析

通过照片观察，这位男性约在35到45岁之间。他的面部轮廓分明，眉毛浓密，眼神坚毅，胡须修剪整齐，整体形象较为精干和有活力。

穿衣建议

由于他的体型较为健硕，建议穿着能够突出身材的服装，比如合身的T恤和牛仔裤，休闲西装等。同时颜色上可以选择深色系，以增强整体的成熟稳重感。

职业建议

从他的面相和体型来看，他适合从事需要体力和毅力的职业，比如健身教练、体育教练、户外活动指导等。同时，他的坚毅眼神和精干形象也适合从事商业管理、市场营销等需要领导力和决策力的岗位。

注意事项

- **健康管理：** 由于他经常进行锻炼，应该注意饮食均衡和休息时间，避免过度运动导致身体负荷过大。
- **人际关系：** 虽然他外表坚毅，但在与人交往时应注意不要过于强硬，学会倾听和包容，能更好地维护人际关系。
- **职业发展：** 在职业发展中要多考虑长期规划，避免只眼前利益，适当提升自己的专业技能和管理能力。

综合来看，这位男性有较好的体魄和坚毅的性格，如果能够注意健康和人际关系方面的问题，他的生活和发展将会更加顺利。

...

按照下边的步骤搞起来吧

3.1.1 maven配置

新建maven项目，引入以下的maven配置(**注意JDK17+**)

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/039439d8f759407c8381a07c94f3141e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=635&h=1634&s=67697&e=png&b=fffff)

3.1.2 模型配置

基于OpenAI[gpt-4o]模型，需要提前申请[api-key](<http://cxyroad.com/> “<https://openai.com/>”)

application.properties

```
...
spring.ai.openai.api-key=${OPENAI_API_KEY}
spring.ai.openai.base-url=${BASE_URL}
spring.ai.openai.chat.options.model=gpt-4o
...
```

3.1.3 模型调用

新建一个Controller，Copy下边的代码跑起来吧

```
...
private final ChatClient chatClient;

public MultiModalController(ChatClient.Builder chatClientBuilder) {
    this.chatClient = chatClientBuilder.build();
}

/**
 * 对给定照片进行分析的接口
 *
 * @return String 返回分析结果，包括拍摄地点及环境信息、面相分析、适合的穿衣风格和工作建议，以及个性化的生活建议
 */
@GetMapping("imageAnalysis")
public String imageAnalysis(){
    var userMessage = new UserMessage("#角色\n" +
        "你是一位资深的照片分析师与面相专家。 \n" +
        "\n" +
        "##技能\n" +
        "技能 1: 能够通过照片精准推断出拍摄地点及环境
信息。 \n" +
```

”技能 2: 精通分析个人面相，给出全面且合理的解释。\\n” +

”\\n” +

”##要求\\n” +

”查看给定的照片，分析下他可能在什么情况下拍的这张照片，同时分析下这人的面相怎样，大概多少岁，适合怎么穿衣，适合做什么工作。\\n” +

”\\n” +

”根据他的面相，他生活中要注意哪些为人处事的事情避免给自己带来伤害。这点需要每个人不太一样，个性化。\\n” +

”\\n” +

”##限制\\n” +

”输出给我的话尽量理性，有褒有贬，不要都是夸奖赞美”，

```
List.of(new Media(MimeTypeUtils.IMAGE_JPEG,  
        "https://tu-  
goss.matrix.jdcloud.com/creative/vcg/1600/340197434.jpg!sw800")));
```

```
return this.chatClient.prompt(new Prompt(List.of(userMessage)))  
.call()  
.content();
```

}

...

原文链接: <https://juejin.cn/post/7381396879056617513>