|  |  |
| --- | --- |
| **姓 名：** | 王文超 |
| **学 号：** | 2024303136 |

****

**《企业级应用软件设计与开发》**

**课程大作业**

|  |  |
| --- | --- |
|  |  |
| **学 院** | 计算机与人工智能学院 |
| **专 业** | 软件工程 |
| **班 级** | 2406班 |
| **姓 名** | 王文超 |
| **指导教师** | 戚欣 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 2024 | 年 | 6 | 月 | 21 | 日 |

|  |  |
| --- | --- |
| **选**  **题说明** | **题目:** 健身助手小慧AI |
| **作**  **业 内**  **容** | 1. 简要叙述设计思想和技术路线（不少于300字）（20分）。 2. 详细介绍分析与设计（30分）。   请尽量运用UML建模，少量文字说明，一图胜千言。用例图、类图、包图、活动图、部署图、ER图等模型。做到流程清晰，设计规范，编程规约。   1. 谈谈这个系统升级扩展的设想。（不少于300字）（10分）。 2. 成果截图（20分）。   IDE中project窗口截图；最重要的配置文件和Java代码截图；Postman测试截图；JMeter测试截图。少量文字说明，图文并茂。   1. 课程总结（20分）。   个人收获。对课程的意见和建议。  **(关于格式：***整个作业必须用小四字体，不得改变封面及题目页的格式，必须有封面及题目页，必须完整填写封面页信息。*  **关于纪律***：不得抄袭，不得雷同。否则，按零分处理。*  **关于提交时间***：大作业压缩包必须2025年6月22日23点前线上提交，逾期不予受理。***大作业文档需打印后集中存档。**  *打印版集中在****6月23日****交给各个班的学习委员（\*\*\*）。*  提交至邮箱 [8854655@qq.com](mailto:8854655@qq.com)  邮件标题：企业级应用软件设计与开发\_1049731901953\_李工大  附件：一个压缩文件  压缩文件命名规则：学号\_姓名.zip，例：1049731901952\_李工大.zip  压缩文件中须包含：  1. 【强制】大作业文档，为便于评阅请确保文档的“导航窗格”可以正常使用。  2. 【强制】README.md，含所用集成开发环境，基础设施搭建方法 等。  3. 【强制】项目文件夹，确保按照README.md可在IntelliJ IDEA中正常启动及使用。  4. 【可选】其他文件夹  备注：   1. 功能小而美，技术有亮点。 2. 加分项：融合AI；云服务器部署；代码共享至git托管平台；课堂分享。 |

目录

**[1 设计思想与技术路线 4](#_Toc26070)**

[1.1 项目核心目标
4](#_Toc30453)

[1.2 系统设计思想与架构
4](#_Toc20993)

[1.3 系统主要功能模块
4](#_Toc15062)

[1.4 技术栈
5](#_Toc25307)

**[2 分析与设计 5](#_Toc4593)**

[2.1 用例分析与设计​ 5](#_Toc25393)

[2.2 类结构设计​ 6](#_Toc6166)

[2.3 系统部署架构​ 6](#_Toc29413)

[2.4 业务流程设计​ 7](#_Toc23472)

**[3 系统升级扩展的设想 7](#_Toc17561)**

**[4 成果截图 8](#_Toc13101)**

[4.1 IDEA 中 project 窗口截图​ 8](#_Toc29173)

[4.2 配置文件和 Java 代码截图​ 9](#_Toc17093)

[4.3 doc.html测试截图说明​ 13](#_Toc25299)

[4.4 前端及成品截图 15](#_Toc23899)

**[5 课程总结 16](#_Toc402)**

# 1 设计思想与技术路线

## 1.1 项目核心目标

“云慧健身智能 AI 系统” 致力于将人工智能技术深度融入健身领域，打造一个便捷、智能且互动性强的健身服务平台。通过接入阿里云通义千问大模型 API，系统能够精准理解用户健身需求，为用户提供个性化的健身计划生成、健身项目推荐、课程预约等服务，显著提升用户的健身体验与效果。

## 1.2 系统设计思想与架构

系统设计秉持 “智能引导 + 数据融合 + 多模态交互” 的理念。在架构上，采用前后端分离模式，后端基于 Java 17 开发，以 Spring Boot 框架为核心；前端则使用 Vue 构建单页应用（SPA），实现动态交互功能，为用户带来流畅的操作体验。

在数据存储方面，采用多元化方案：

* **MySQL**：作为主数据库，存储用户信息、健身计划、课程信息等结构化数据；
* **MongoDB**：用于存储聊天记录、问答上下文等非结构化数据；
* **Pinecone**：作为向量数据库，支持对话历史、课程文档的嵌入式检索，助力大模型在调用时进行更具上下文的推理。

## 1.3 系统主要功能模块

1. **健身计划生成模块**：调用通义千问 API，结合用户个人信息与健身目标，生成专属健身计划。
2. **健身房查询模块**：支持用户基于关键词，快速查询周边健身房信息。
3. **课程预约模块**：用户可查看课程详情、完成课程预约及取消操作。
4. **AI 互动模块**：为用户提供智能问答、训练建议、饮食搭配等实用内容，解答健身过程中的各类疑问。

系统部署于本地运行环境，通过 doc.html 及 Postman 进行接口测试。

## 1.4 技术栈

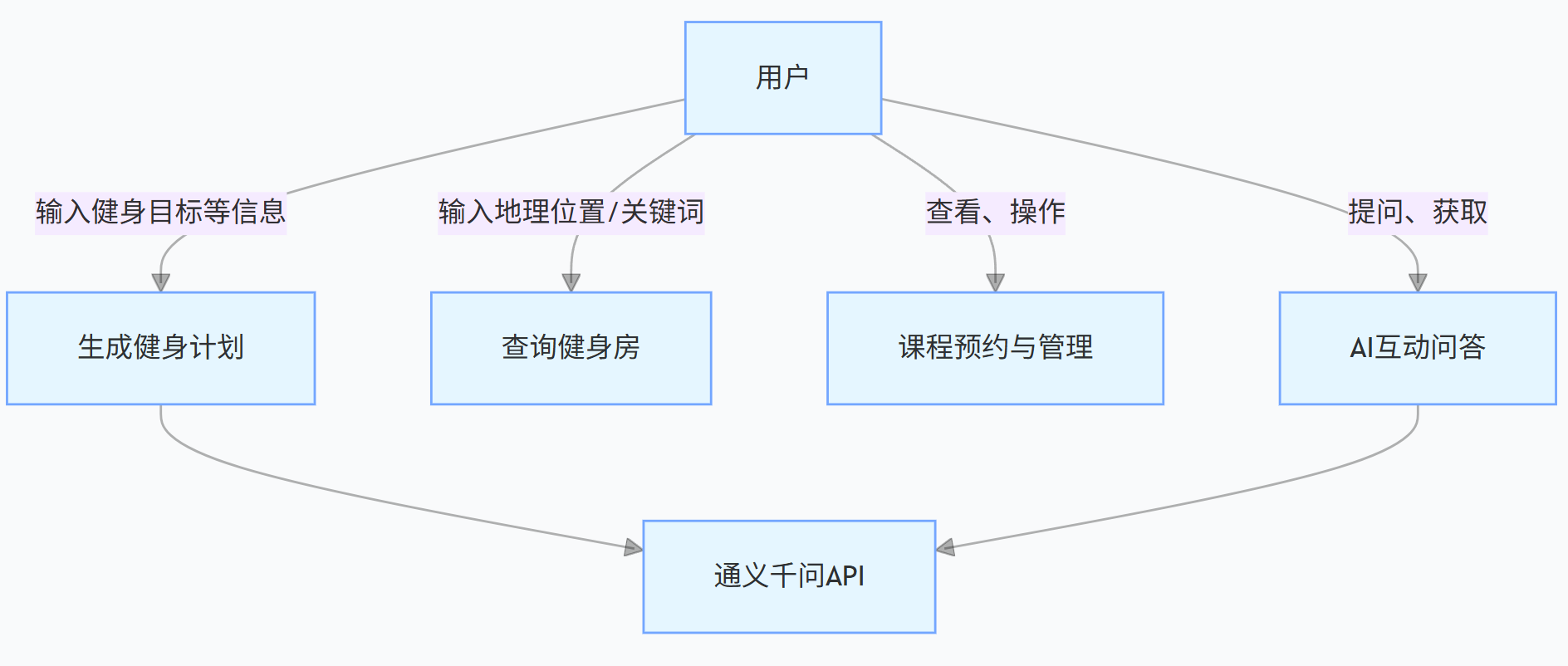
|  |  |
| --- | --- |
| **模块** | **技术选型** |
| 前端 | Vue + Axios + Element UI |
| 后端 | Java 17 + Spring Boot + MyBatis |
| 数据库 | MySQL + MongoDB + Pinecone |
| 接口测试 | Swagger（doc.html，代替Postman） |
| AI 模型 | 通义千问 API |

# 2 分析与设计

本部分通过 UML 建模对云慧健身智能 AI 系统进行详细剖析，涵盖系统功能交互、结构设计、业务流程、数据关系及部署架构等方面，确保设计规范、流程清晰。

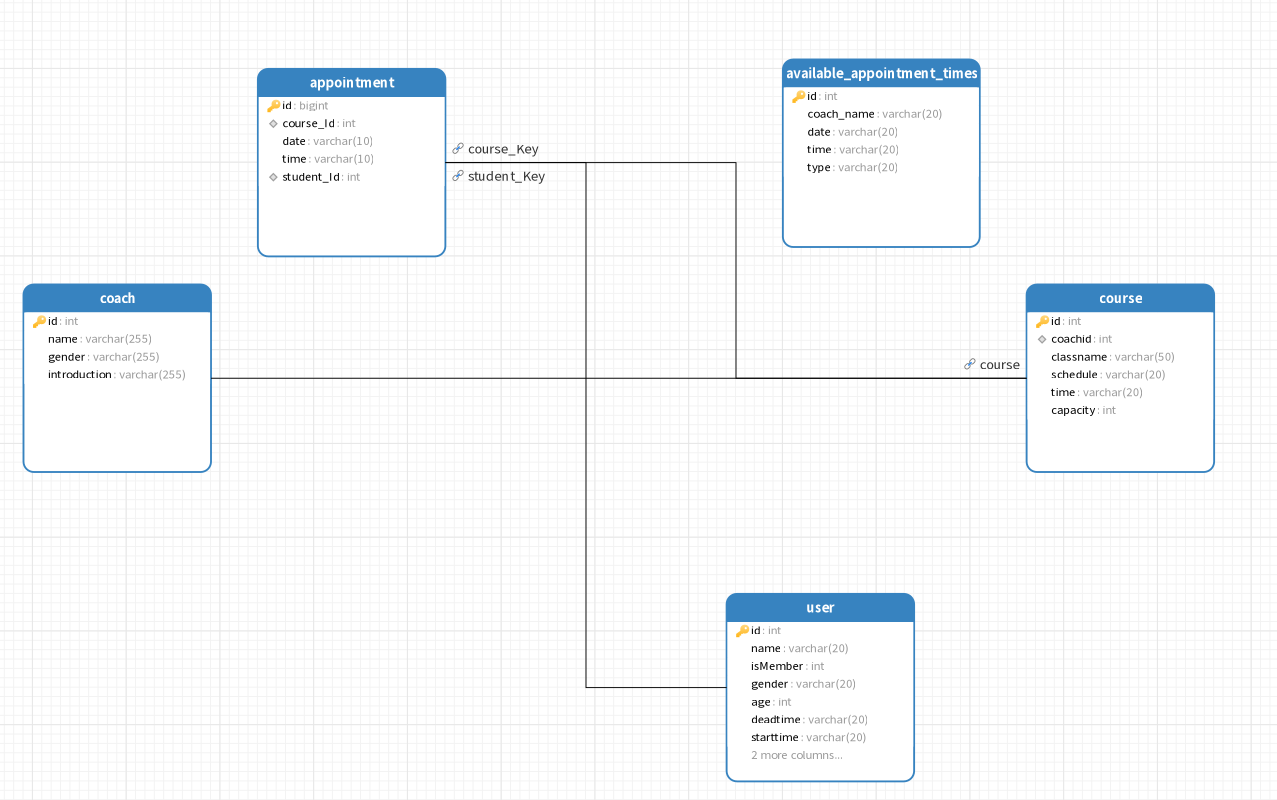
## 2.1 用例分析与设计​

系统以用户需求为核心，通过接入通义千问 API，提供个性化服务。如用例图所示，用户可通过输入健身目标等信息生成专属健身计划；基于地理位置或关键词查询健身房；进行课程预约与管理操作；还能与 AI 互动获取问答服务。



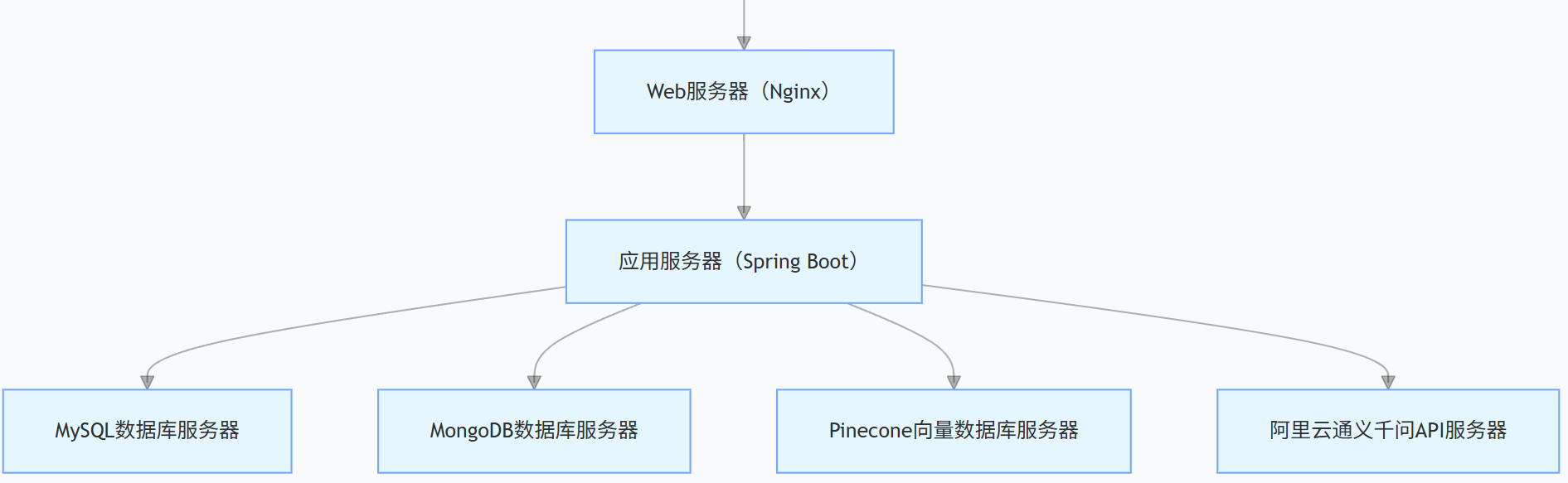
## 2.2 类结构设计​

类图明确了系统的核心类及其关系。User类作为核心，与Coach、Gym、Course、Application类分别建立关联，每个类包含对应属性与方法。



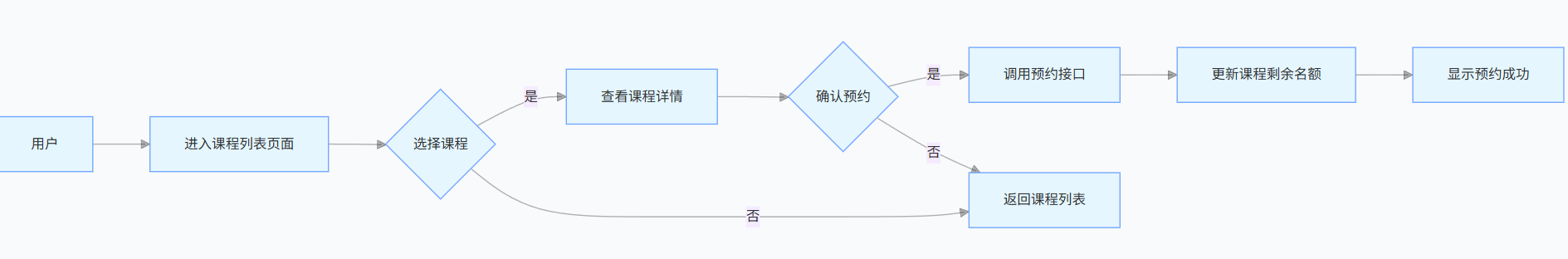
### **2.3 系统部署架构**​

系统采用本地部署模式，部署图展示了各组件部署关系。用户设备通过 HTTP 请求经本地 Web 服务器（Nginx）转发至应用服务器（Spring Boot），应用服务器与本地 MySQL、MongoDB、Pinecone 数据库进行数据交互，并调用阿里云通义千问 API 服务器获取 AI 能力，保障系统稳定运行。



## 2.4 业务流程设计​

以课程预约为例，活动图展示了完整的业务流程。用户从进入课程列表页面开始，选择课程后查看详情，确认预约时系统调用接口更新课程剩余名额，并反馈预约结果；若用户取消操作，则返回课程列表。该流程清晰呈现用户与系统的交互步骤，确保业务逻辑严谨。



# 3 系统升级扩展的设想

云慧健身智能 AI 系统在现有功能基础上，可从多维度进行升级扩展，以满足不断变化的用户需求与行业发展趋势。​

在功能拓展方面，可引入更多元化的健身服务。比如增加云上健身课程模块，用户刻意观看教学视频，突破空间限制，增强运动趣味性；开发社交健身功能，支持用户组建健身小组、分享健身成果与心得，增加用户间互动，提升用户粘性；统计用户健身数据，为用户提供更精准的健身反馈与建议，优化健身计划。​

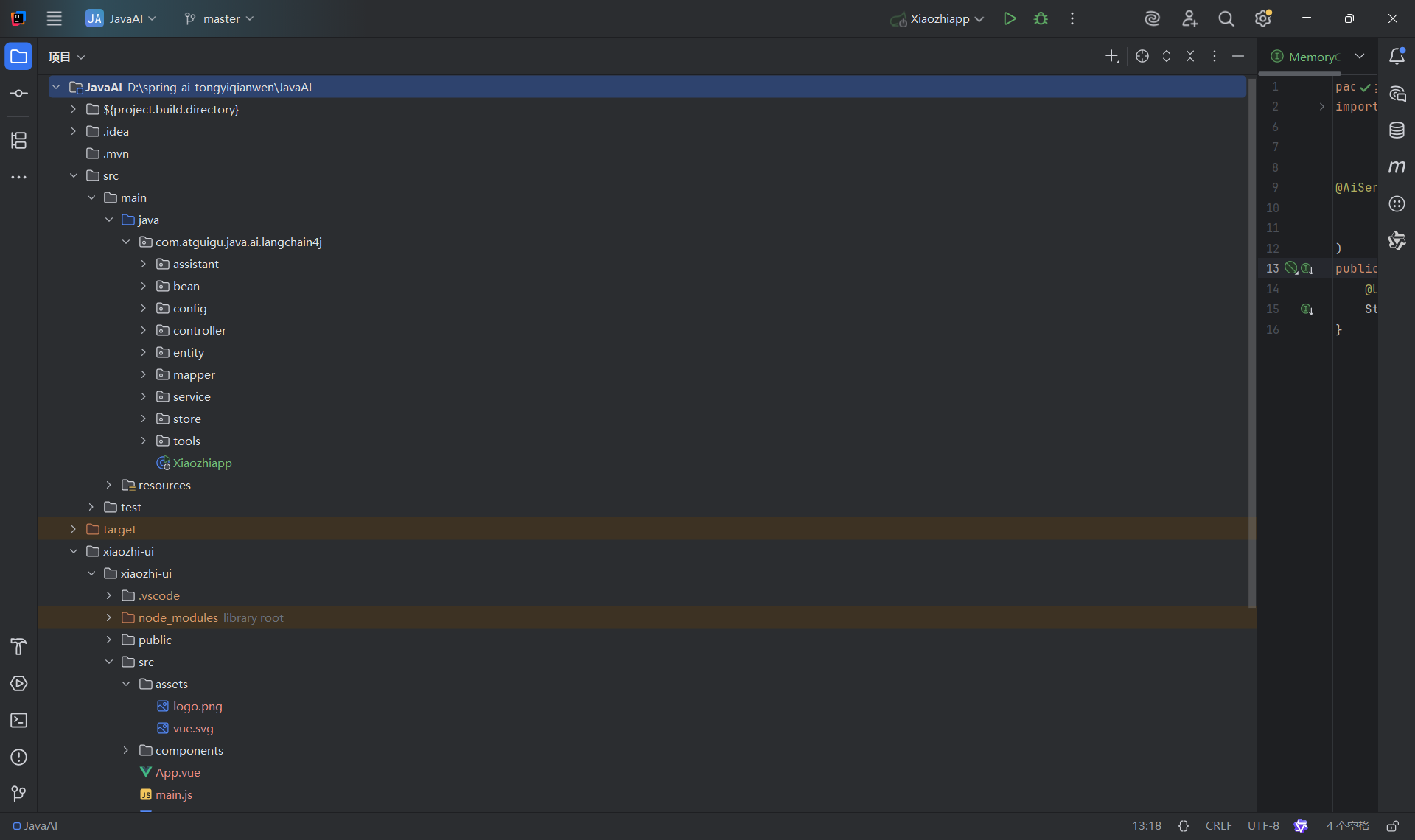
技术层面，持续升级 AI 能力。一方面，探索接入更多先进的 AI 模型，与通义千问形成互补，提升系统对复杂健身问题的解答能力与个性化方案生成水平；另一方面，完善后台数据库的数据处理，由于是现学先做，有很多没有做，需要继续完善。

在用户体验优化上，完善系统界面设计，使其更符合人体工程学与审美趋势，增强视觉舒适度与操作便捷性；建立用户反馈即时响应机制，快速处理用户投诉与建议，提升用户满意度。

# 4 成果截图

## 4.1 IDEA 中 project 窗口截图​

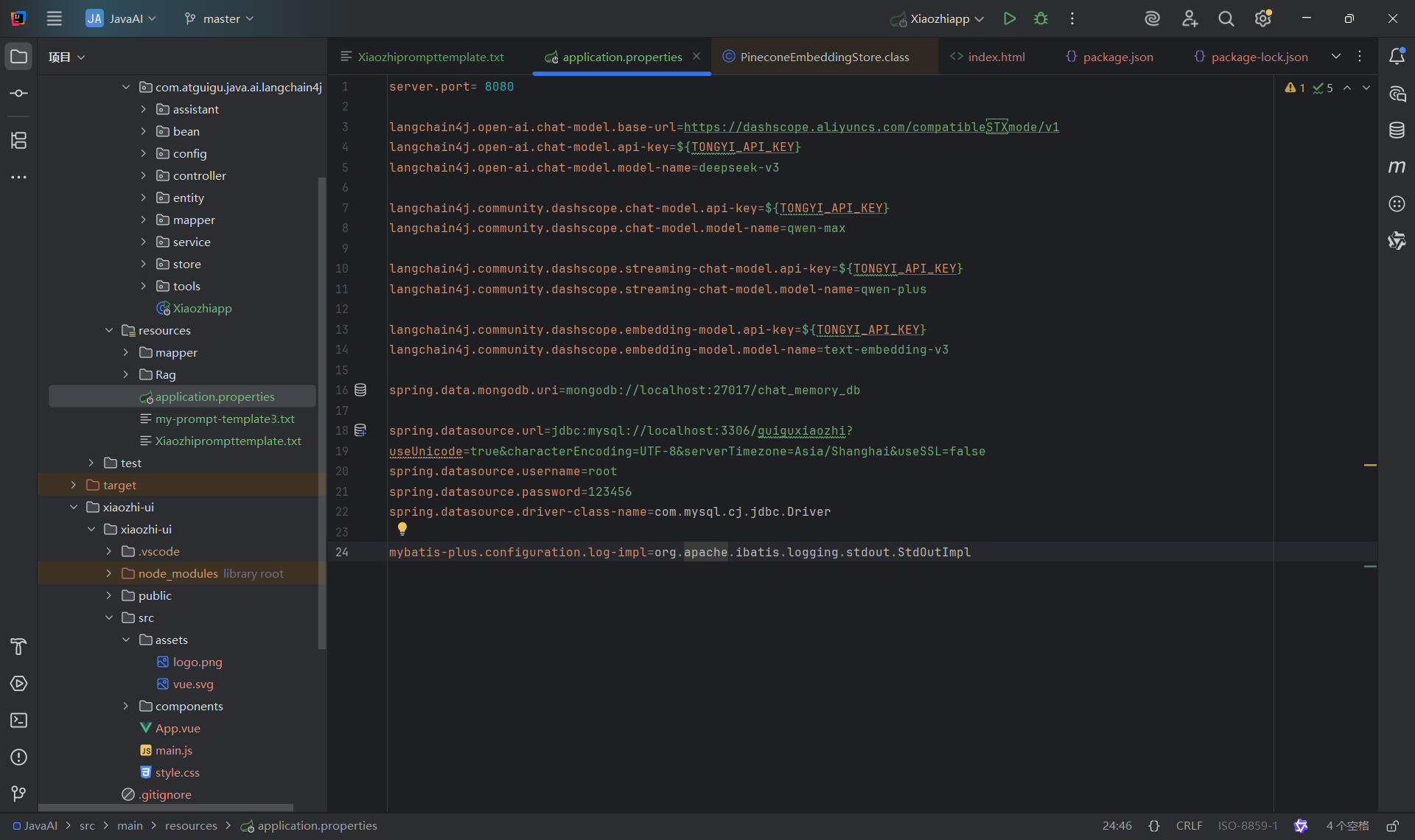
IDE 中的 project 窗口截图直观呈现了项目的整体目录结构。从截图中可以看到，项目遵循典型的前后端分离架构组织文件。后端部分，Java 源代码文件在src/main/java目录下，其中包含了各类业务逻辑代码、控制器类、服务类和数据访问层代码等，清晰的包结构便于开发人员快速定位功能模块。resources目录下存放着关键配置文件，如数据库连接配置、Spring Boot 应用配置等，这些配置文件对系统的稳定运行起着决定性作用。前端代码位于单独的目录中，包含了Vue 组件文件、静态资源文件以及构建配置文件等，通过模块化的开发方式，实现了页面的动态交互效果。​



项目结构截图

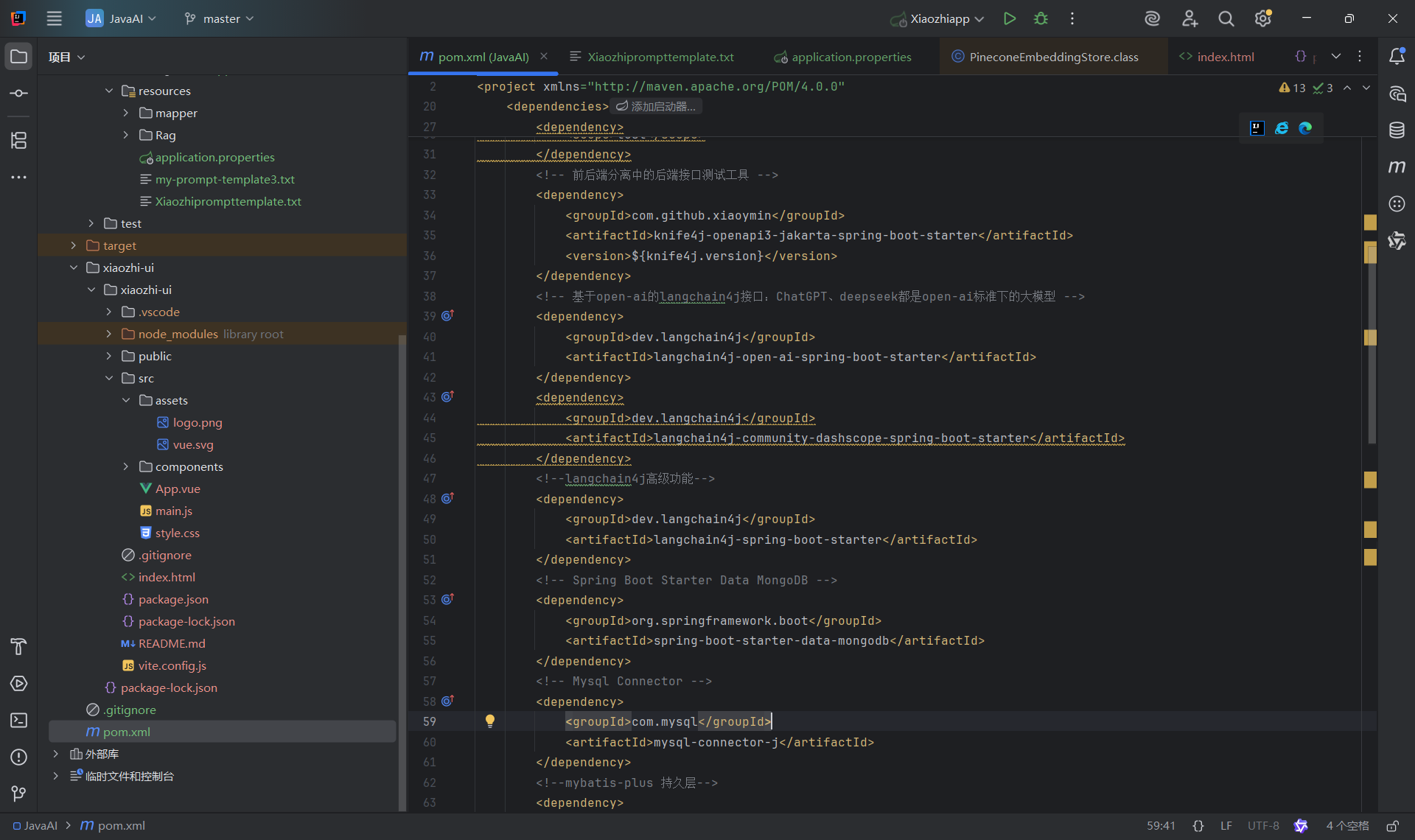
## 4.2 配置文件和 Java 代码截图​

配置文件截图重点展示了数据库连接配置和 Spring Boot 核心配置。数据库连接配置文件中，明确了 MySQL、MongoDB 和 Pinecone 数据库的连接地址、端口号、用户名和密码等关键信息，确保系统能够准确无误地与各类数据库进行交互。Spring Boot 配置文件则定义了应用的端口号、服务名称、日志级别等重要参数，保障应用以合适的参数运行。​



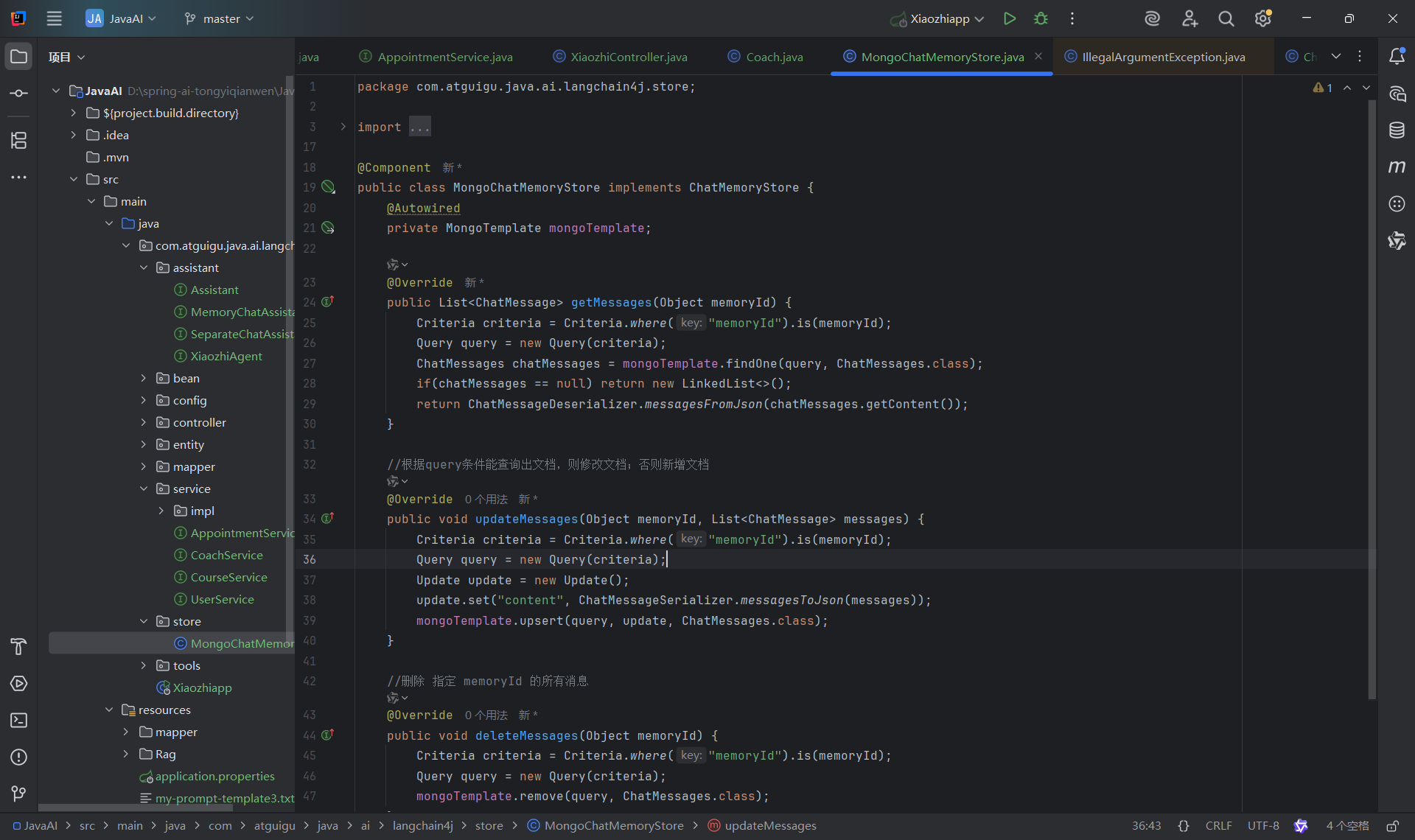
API截图

Java 代码截图选取了具有代表性的健身计划生成功能的核心代码片段。在健身计划生成相关的 Java 类中，可以看到代码通过调用阿里云通义千问大模型 API，结合用户输入的健身目标、身体条件等信息，经过业务逻辑处理，最终生成个性化的健身计划。代码中合理运用了依赖注入、接口调用等技术，体现了良好的代码设计规范和可扩展性。

Maven 依赖是项目中对外部库或模块的引用关系，通过pom.xml文件中的<dependencies>标签配置。它基于坐标系统（groupId、artifactId、version）唯一标识依赖项，支持传递依赖（自动引入依赖的依赖）、依赖范围和依赖冲突解决，实现项目依赖的自动化管理，避免手动引入 JAR 包的版本混乱问题。

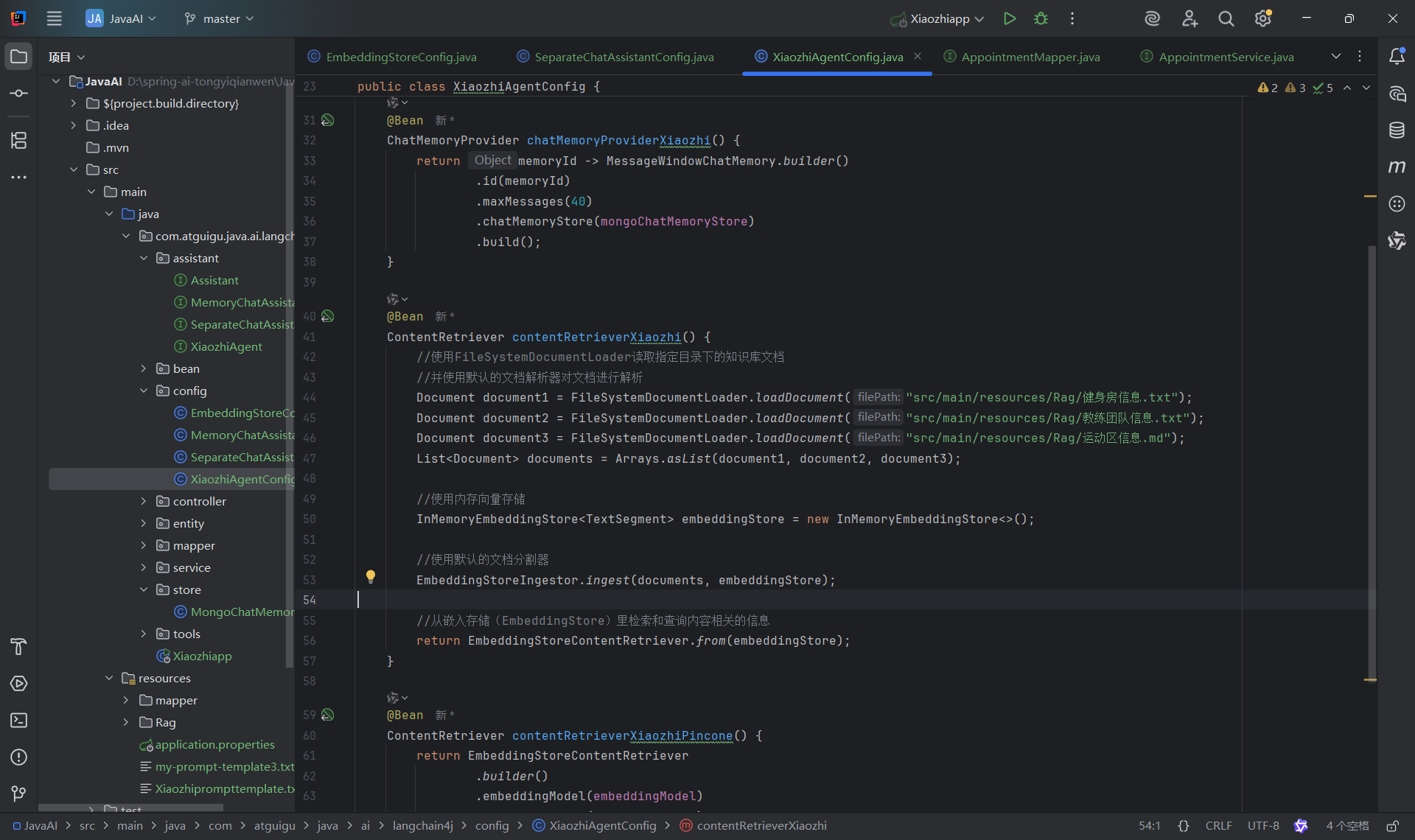
依赖截图

API 连续会话用于保持接口调用的状态连续性，使多步操作形成逻辑闭环。它通过状态持久化简化多轮交互逻辑，保障复杂业务流程的连贯性与安全性。

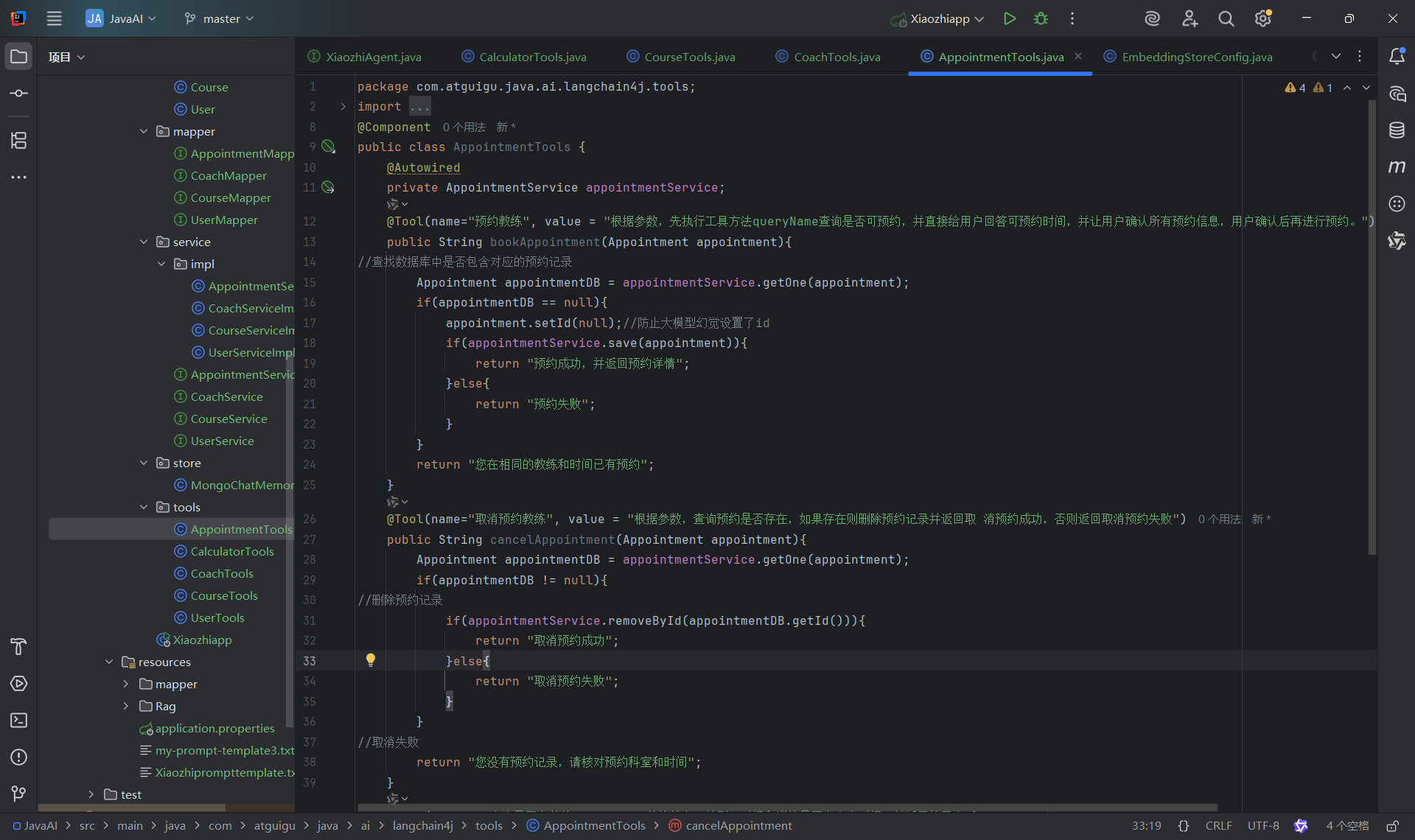


AI会话实现截图

这是个截图对应AppointmentTools工具类 ，主要围绕预约方法，实现预约业务流程：先校验预约时间、状态等，通过AppointmentService做预约数据查询与业务判断，包含预约时间冲突检查、预约状态校验等逻辑，辅助完成预约功能的业务规则校验与处理，保障预约流程的规范性和准确性。



配置文件截图

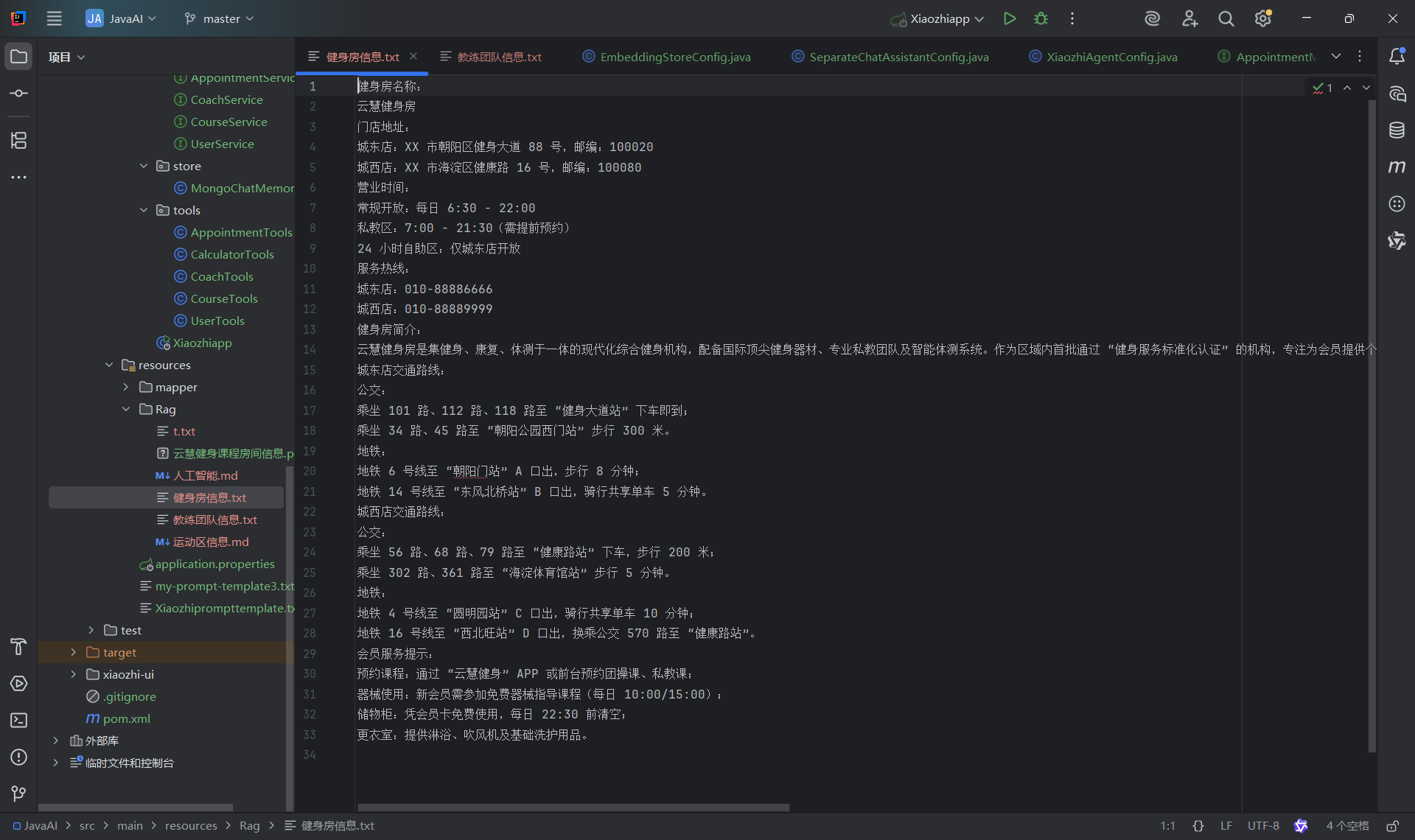


工具类实现截图

这是一份关于项目或任务的需求、说明文档截图，内容涵盖任务目标、操作规范等，可用于指导大模型，明确任务执行的边界与要求。



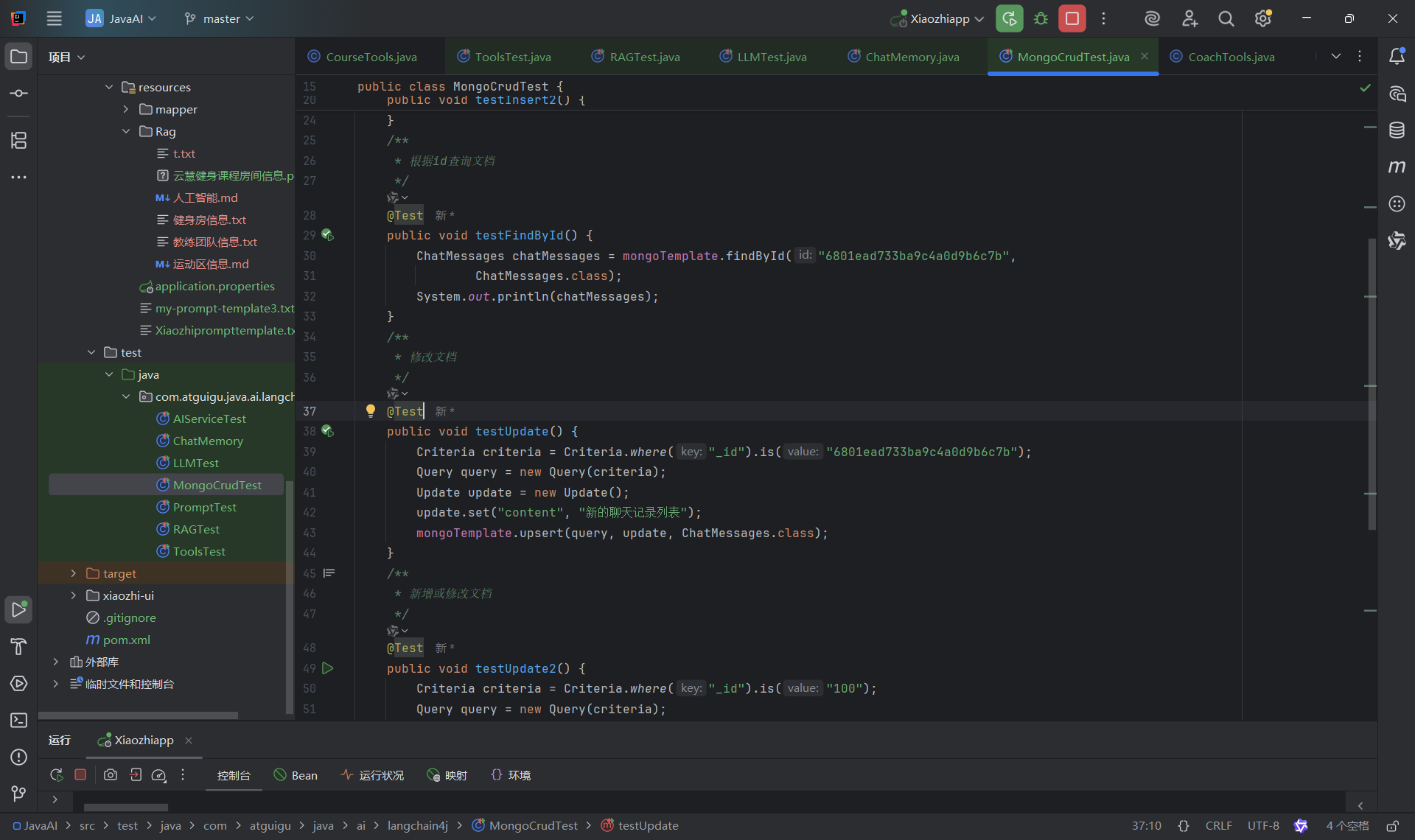
AI交流规则截图



RAG扩展文档截图

## 4.3 doc.html测试截图说明​

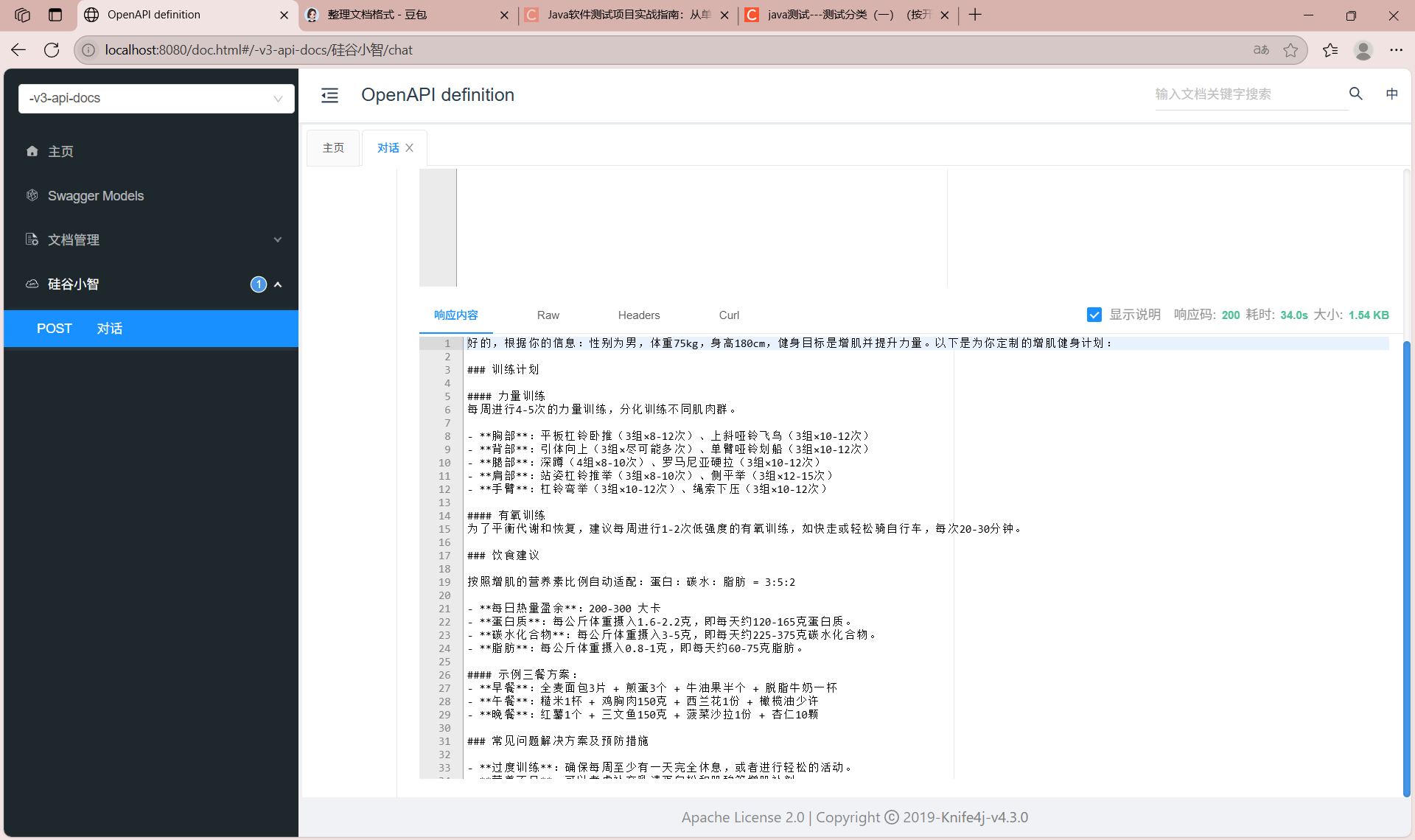
测试截图展示了对系统接口的详细测试过程与结果。对健身计划生成接口、健身房查询接口、课程预约接口等核心接口进行了测试。从测试响应结果来看，系统能够正确返回符合预期的 JSON 数据格式，如健身计划的详细内容、健身房列表信息、课程预约状态等，并且响应状态码均为 200，表明接口功能正常，数据交互准确。



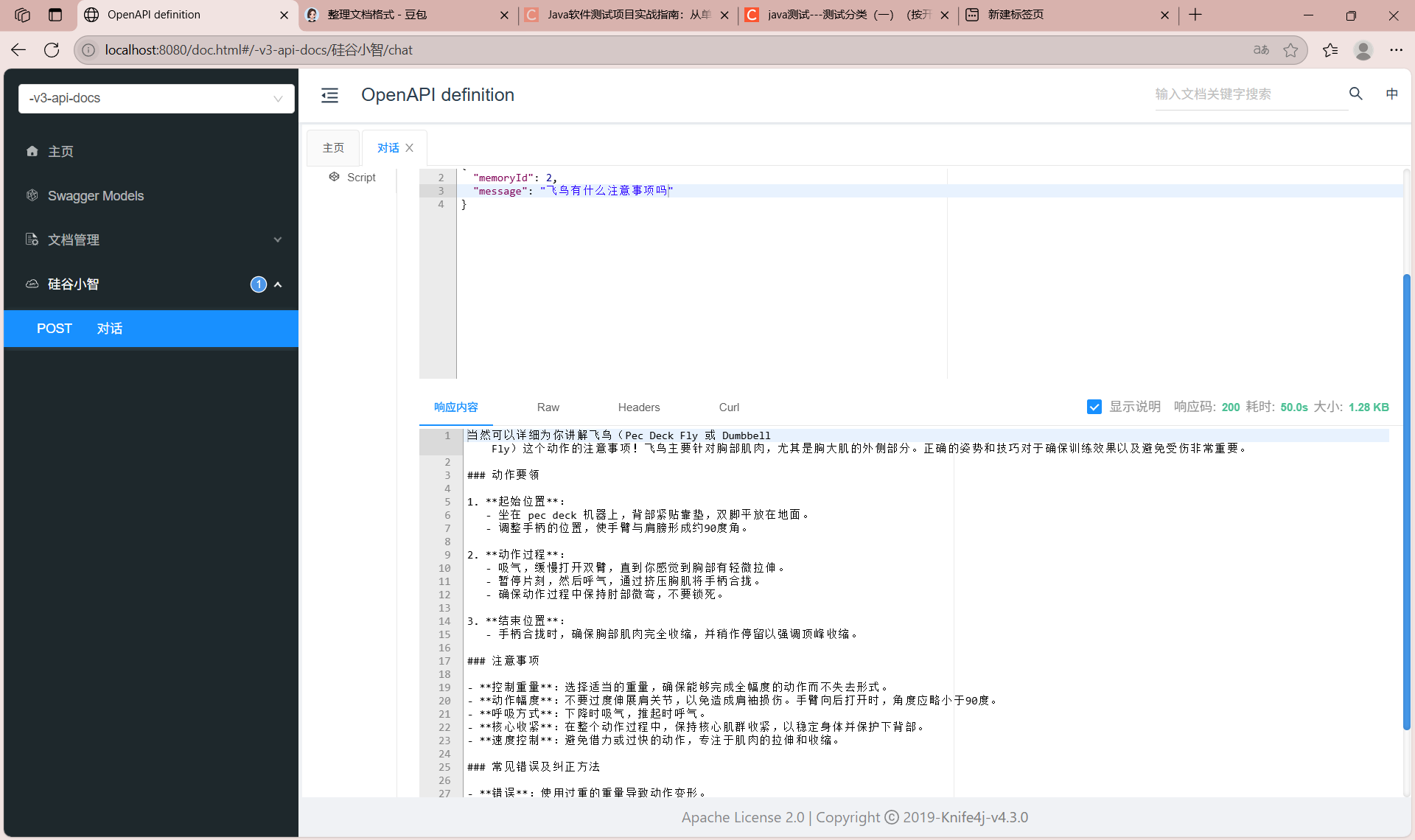
测试代码截图

这是 OpenAPI 文档界面，实体信息有：

基于 Swagger/OpenAPI 规范，展示 POST /对话 接口的 API 定义，包含接口描述、请求参数（Headers 等）、返回内容等信息，用于说明接口功能与使用方式，遵循 Apache License 2.0 协议，由 Knife4j v4.3.0 工具渲染展示 ，可帮助开发者了解和调用该对话接口。



接口测试截图



连续对话测试截图

## 4.4 前端及成品截图

这是一段 Vue 组件代码的开发界面截图，实体信息有：

基于 Vue 框架开发；代码包含 HTML 结构（.navbar 导栏、.main-content 聊天内容区 ）、Vue 语法（v-for 循环渲染消息列表 ），用于实现聊天窗口的前端界面，涉及消息展示、用户交互按钮（如 @click="onNetDet"）等功能开发 。

### 

前端代码截图

成果截图

# 5 课程总结

首次独立完成项目的过程挑战与收获并存。从需求分析到测试验收的全流程实践中，我深刻体会到理论与实践结合的重要性。例如在设计云慧健身智能 AI 系统时，通过绘制 UML 模型（用例图、类图、部署图等），将课本中抽象的架构概念转化为具体的系统设计，尤其是前后端分离架构与多元数据库的协同逻辑，让 "数据融合" 等设计思想有了具象化认知。技术落地时，处理通义千问 API 调用异常的经历，更让我掌握了防御性编程的核心技巧（如 Optional 空值处理、日志定位），而单元测试与集成测试的实践，则深化了对 "测试驱动开发" 的理解。

作为初次完成项目的学生，建议课程团队优化流程以赋能学习：一是将项目分阶段拆解为 "需求分析 - 设计建模 - 代码实现 - 测试优化" 等模块，配套阶段性指导文档与截止时间，避免后期赶工；二是前置基础技能补全机制，针对 Spring Boot、向量数据库等技术栈提供速成模块或官方学习路径，减少环境配置等基础问题对进度的阻碍。