

《Ai Agent》第3-4节：根据 Ai Agent 案例，设计库表

来自：码农会锁

小傅哥

2025年06月14日 16:29

本章重点：★★☆☆☆

课程视频：<https://t.zsxq.com/Tx438>

代码分支：<https://gitcode.net/KnowledgePlanet/ai-agent-station-study/-/tree/3-4-agent-table-design>

工程代码：<https://gitcode.net/KnowledgePlanet/ai-agent-station-study>

版权说明：©本项目与星球签约合作，受《中华人民共和国著作权法实施条例》。版权法保护，禁止任何理由和任何方式公开(public)源码、资料、视频等小傅哥发布的星球内容到Github、Gitee等各类平台，违反可追究进一步的法律责任。

作者：小傅哥

博客：<https://bugstack.cn>

沉淀、分享、成长，让自己和他人都能有所收获！😊

一、本章诉求

二、拆分设计

三、库表细节

1. 智能体

2. 功能表

3. 工具表

4. 链接表

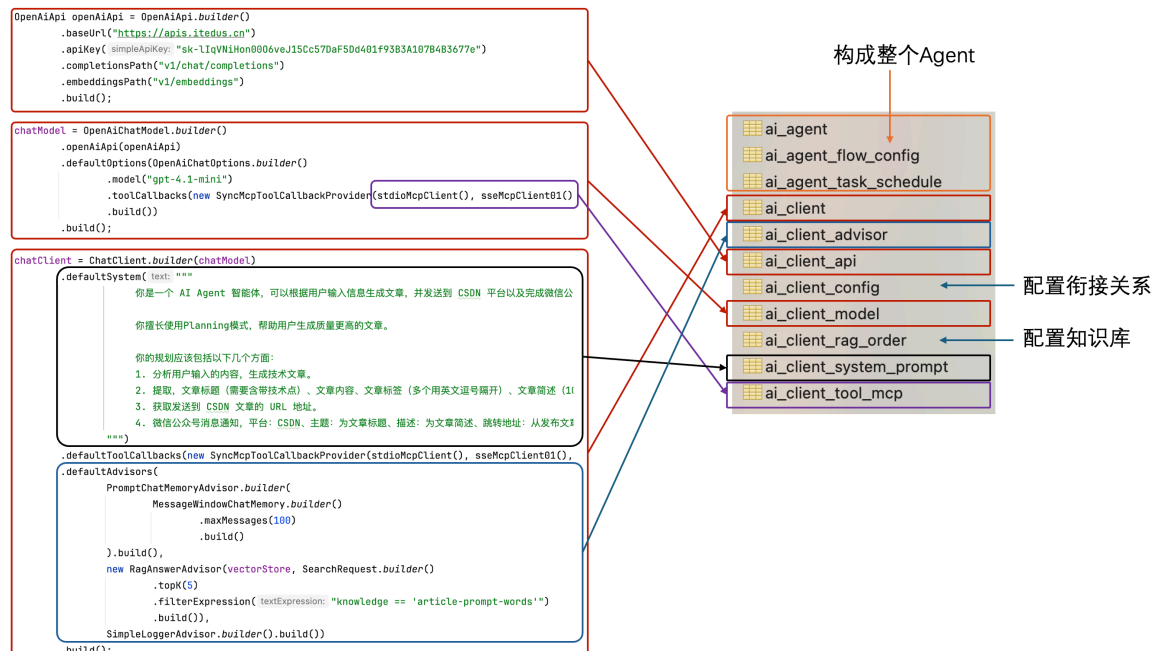
四、读者作业

一、本章诉求

根据 Ai Agent 的代码案例，设计用于解耦，硬编码流程的库表。在后续的代码开发中，根据库表配置的 Ai Agent 流程所需的，模型、提示词、顾问、工具等，动态实例化出 Ai Agent 服务。

二、拆分设计

如图，为对应的Ai Agent 案例代码，映射出要拆分的库表设计；



首先，整个代码构建的整个 Ai Agent 最小化单元服务，我们可以根据这样的服务信息设计出库表结构。

第一步，从上到下，OpenAiApi 是最基础单元结构，可以被多个 OpenAiChatModel 使用，它可以被拆分出第一张表。

第二步，构建 OpenAiChatModel，这个阶段，需要 openAiApi、model对话模型、tool mcp 工具。其中model对话模型时一种固定固定资源，可以直接放到 ai_client_model 模型中，而 openAiApi、mcp 工具，都属于复杂配置，则需要额外的外部关联来衔接。也就是后面的 ai_client_config 配置，用于配置衔接关系。

第三步，ChatClient 对话客户端，这部分的实例化过程都是和外部其他的资源关联，本身表设计只要有一个客户端的唯一id和客户端的描述介绍即可。

第四步，给 mcp 增加一个表，mcp 服务是非常重要的，有 mcp 才有 agent 服务。mcp 的启动有 stido、sse 两种方式，每种方式都有对应的配置文件 json 数据。

第五步，defaultSystem 系统提示词，需要单独拆分出来。提示词等于智能体的大脑，有人说，其实 Ai Agent 就是 prompt 的堆叠，所以写提示词是很重要的。

第六步，advisor 顾问角色，在 Spring Ai 框架中，以顾问的方式，访问记忆上下文，知识库资源，所以这部分也要单独设计库表。

第七步，设计一个 ai_client_config，用于配置；api、model、client、prompt、mcp、advisor的衔接关系。

第八步，设计 ai_agent、ai_agent_flow_config，也就是一个 ai agent，是可以连续调用多个 ai client 客户端的。

第九步，设计 ai_agent_stask_schedule 任务，这是一种触达手段，可以把配置好的任务，让 task 定时执行，如自动发帖、系统异常巡检、舆情风险检测、系统配置变更、运营活动报表等。

第十步，ai_client_rag_order，是知识库表，用于上传知识库做一个记录，这样顾问角色就可以访问知识库内容了。

注意；chat_client 客户端的初始化过程中，也可以增加 mcp 服务，这部分在 chat_model 模型构建中，也可以增加 mcp，选择在 chat_model 增加即可。

三、库表细节

本系统是基于Spring AI框架构建的AI Agent智能体平台，通过数据库驱动的配置化方式实现AI模型、客户端、工具和智能体的解耦和自动实例化。系统支持多种AI模型接入、工具集成（MCP协议）、任务调度等功能。



1. 智能体

1.1 ai_agent

```
CREATE TABLE `ai_agent` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `agent_id` varchar(64) NOT NULL COMMENT '智能体ID',  
  `agent_name` varchar(50) NOT NULL COMMENT '智能体名称',  
  `description` varchar(255) DEFAULT NULL COMMENT '描述',  
  `channel` varchar(32) DEFAULT NULL COMMENT '渠道类型(agent, chat_stream)',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_agent_id` (`agent_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='AI智能体配置表'.
```

表说明：AI智能体配置表是系统的核心表，用于存储智能体的基本信息。每个智能体代表一个可独立运行的AI服务单元，可以通过不同的渠道（如agent或chat_stream）进行交互。智能体是整个系统的顶层实体，可以关联多个客户端组成 workflow。

1.2 ai_agent_task_schedule

```
CREATE TABLE `ai_agent_task_schedule` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `agent_id` bigint NOT NULL COMMENT '智能体ID',
  `task_name` varchar(64) DEFAULT NULL COMMENT '任务名称',
  `description` varchar(255) DEFAULT NULL COMMENT '任务描述',
  `cron_expression` varchar(50) NOT NULL COMMENT '时间表达式(如: 0/3 * * * * *)',
  `task_param` text COMMENT '任务入参配置(JSON格式)',
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:无效,1:有效)',
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',
  PRIMARY KEY (`id`),
  KEY `idx_agent_id` (`agent_id`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='智能体任务调度配置表'
```

表说明：智能体任务调度配置表用于设置智能体的定时任务。通过cron表达式配置执行频率，可以实现智能体的自动化运行，如定期生成报告、监控系统状态、自动回复等功能。任务参数以JSON格式存储，支持灵活的参数配置。

1.3 ai_agent_flow_config

```
CREATE TABLE `ai_agent_flow_config` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',
  `agent_id` bigint NOT NULL COMMENT '智能体ID',
  `client_id` bigint NOT NULL COMMENT '客户端ID',
  `sequence` int NOT NULL COMMENT '序列号(执行顺序)',
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_agent_client_seq` (`agent_id`,`client_id`,`sequence`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='智能体-客户端关联配置表'
```

表说明：智能体-客户端关联表定义了智能体的工作流程。通过sequence字段指定多个客户端的执行顺序，实现复杂的智能体 workflow。一个智能体可以按顺序调用多个客户端，形成完整的处理链路，支持复杂的多步骤AI任务处理流程。

2. 功能表

2.1 ai_client_api

```
CREATE TABLE `ai_client_api` (
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '自增主键ID',
  `api_id` varchar(64) NOT NULL COMMENT '全局唯一配置ID',
  `base_url` varchar(255) NOT NULL COMMENT 'API基础URL',
  `api_key` varchar(255) NOT NULL COMMENT 'API密钥',
  `completions_path` varchar(255) NOT NULL COMMENT '补全API路径',
  `embeddings_path` varchar(255) NOT NULL COMMENT '嵌入API路径',
  `status` tinyint NOT NULL DEFAULT '1' COMMENT '状态: 0-禁用, 1-启用',
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',
  PRIMARY KEY (`id`),
  UNIQUE KEY `uk_api_id` (`api_id`),
  KEY `idx_status` (`status`)
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='OpenAI API配置表'
```

表说明：OpenAI API配置表存储与AI服务提供商的连接信息，是系统与外部AI服务通信的基础。包含API密钥、基础URL和各种API路径配置，支持不同的API端点如补全和嵌入。这是系统最基础的配置单元，可被多个模型共享使用。

2.2 ai_client_model

```
CREATE TABLE `ai_client_model` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '自增主键ID',  
  `model_id` varchar(64) NOT NULL COMMENT '全局唯一模型ID',  
  `api_id` varchar(64) NOT NULL COMMENT '关联的API配置ID',  
  `model_name` varchar(64) NOT NULL COMMENT '模型名称',  
  `model_type` varchar(32) NOT NULL COMMENT '模型类型: openai、deepseek、claude',  
  `status` tinyint NOT NULL DEFAULT '1' COMMENT '状态: 0-禁用, 1-启用',  
  `create_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_model_id` (`model_id`),  
  KEY `idx_api_config_id` (`api_id`),  
  KEY `idx_status` (`status`)  
) ENGINE=InnoDB AUTO_INCREMENT=2 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='聊天模型配置表';
```

表说明：聊天模型配置表定义了系统可用的AI模型。通过关联API配置，指定使用的具体模型（如GPT-4、Claude等）。支持多种模型类型（openai、deepseek、claude等），使系统能够灵活选择不同的AI模型来处理不同类型的任务，提高系统的适应性和性能。

2.3 ai_client

```
CREATE TABLE `ai_client` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `client_id` varchar(64) NOT NULL COMMENT '客户端ID',  
  `client_name` varchar(50) NOT NULL COMMENT '客户端名称',  
  `description` varchar(1024) DEFAULT NULL COMMENT '描述',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `client_id` (`client_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='AI客户端配置表'
```

表说明：AI客户端配置表是对话客户端的基本信息表。客户端是系统中的功能单元，可以独立处理特定类型的AI任务。客户端通过关联配置表与模型、提示词、工具等资源建立连接，形成完整的功能单元。多个客户端可以组合成智能体工作流，实现复杂的业务逻辑。

3. 工具表

3.1 顾问角色

```
CREATE TABLE `ai_client_advisor` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `advisor_id` varchar(64) NOT NULL COMMENT '顾问ID',  
  `advisor_name` varchar(50) NOT NULL COMMENT '顾问名称',  
  `advisor_type` varchar(50) NOT NULL COMMENT '顾问类型(PromptChatMemory/RagAnswer/SimpleLoggerAdvisor等)',  
  `order_num` int DEFAULT '0' COMMENT '顺序号',  
  `ext_param` varchar(2048) DEFAULT NULL COMMENT '扩展参数配置, json 记录',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_advisor_id` (`advisor_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='顾问配置表';
```

表说明：顾问配置表定义了系统中的顾问角色。在Spring AI框架中，顾问是访问记忆上下文和知识库资源的重要组件。支持多种顾问类型，如对话记忆管理(PromptChatMemory)、知识库检索(RagAnswer)和日志记录(SimpleLoggerAdvisor)等。顾问可以增强AI客户端的能力，提供上下文管理和知识检索等功能。

3.2 ai_client_system_prompt

```
CREATE TABLE `ai_client_system_prompt` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `prompt_id` varchar(64) NOT NULL COMMENT '提示词ID',  
  `prompt_name` varchar(50) NOT NULL COMMENT '提示词名称',  
  `prompt_content` text NOT NULL COMMENT '提示词内容',  
  `description` varchar(1024) DEFAULT NULL COMMENT '描述',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_prompt_id` (`prompt_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='系统提示词配置表';
```

表说明：系统提示词配置表存储AI模型的系统提示词。提示词是智能体的“大脑”，决定了AI的行为方式和回答风格。通过精心设计的提示词，可以引导AI生成更符合预期的回答。系统提示词是AI Agent的核心组成部分，对AI的输出质量有决定性影响。

3.3 ai_client_tool_mcp

```
CREATE TABLE `ai_client_tool_mcp` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `mcp_id` varchar(64) NOT NULL COMMENT 'MCP名称',  
  `mcp_name` varchar(50) NOT NULL COMMENT 'MCP名称',  
  `transport_type` varchar(20) NOT NULL COMMENT '传输类型(sse/stdio)',  
  `transport_config` varchar(1024) DEFAULT NULL COMMENT '传输配置(sse/stdio)',  
  `request_timeout` int DEFAULT '180' COMMENT '请求超时时间(分钟)',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `uk_mcp_id` (`mcp_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=11 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='MCP客户端配置表';
```

表说明：MCP客户端配置表定义了系统中的工具集成配置。MCP (Model-Control-Protocol) 是实现AI Agent功能的关键组件，使AI能够调用外部工具和服务。支持多种传输类型 (sse/stdio)，可以配置不同的通信方式。MCP工具使AI能够执行实际操作，如查询数据、调用API等，是实现真正Agent功能的核心。

4. 链接表

```
CREATE TABLE `ai_client_config` (  
  `id` bigint NOT NULL AUTO_INCREMENT COMMENT '主键ID',  
  `source_type` varchar(32) NOT NULL COMMENT '源类型 (model、client)',  
  `source_id` varchar(64) NOT NULL COMMENT '源ID (如 chatModelId、chatClientId 等)',  
  `target_type` varchar(32) NOT NULL COMMENT '目标类型 (model、client)',  
  `target_id` varchar(64) NOT NULL COMMENT '目标ID (如 openAiApiId、chatModelId、systemPromptId、advisorId 等)',  
  `ext_param` varchar(1024) DEFAULT NULL COMMENT '扩展参数 (JSON格式)',  
  `status` tinyint(1) DEFAULT '1' COMMENT '状态(0:禁用,1:启用)',  
  `create_time` datetime DEFAULT CURRENT_TIMESTAMP COMMENT '创建时间',  
  `update_time` datetime DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT '更新时间',  
  PRIMARY KEY (`id`),  
  KEY `idx_source_id` (`source_id`),  
  KEY `idx_target_id` (`target_id`)  
) ENGINE=InnoDB AUTO_INCREMENT=9 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci COMMENT='AI客户端统一关联'
```

AI客户端统一关联配置表是系统的核心关联表，用于配置各组件之间的关系。通过source_type/source_id和target_type/target_id字段，灵活定义不同组件间的关联关系，如模型与API的关联、客户端与模型的关联、客户端与提示词的关联等。这种设计使系统具有高度的灵活性和可配置性，支持复杂的组件组合。

四、读者作业

简单作业：完成库表设计，创建库表，理解库表字段。

复杂作业：把数据库表在工程中写好crud代码，后续我们进行业务功能开发。