

河南中医药大学信息技术学院（智能医疗行业学院）产教协同课程建设成果
智能医学工程专业《医疗信息系统开发》课程

第09-基于人工智能算法开发软件功能

王烈

河南经方云科技有限公司

河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

<https://aitcm.hactcm.edu.cn>

2025/11/21



本章概要

本章内容

- 使用场景
- 常用技术
- JAVA代码生成
- 怎么和大模型融合



使用场景

- 项目骨架生成

快速创建可运行的基础项目 Spring Initializr

- 样板代码生成

消除重复的 Getter/Setter/Builder 等代码 Lombok, IDE 内置功能

- 模型转换生成

自动生成对象之间的转换代码

- API 相关生成

保证 API 契约一致，生成客户端/服务端代码 Swagger Codegen, OpenAPI Generator, gRPC protoc

- 持久层生成

根据数据库表生成 Entity, Mapper, XML 等 MyBatis Generator, MyBatis-Plus



常用技术

- springBoot +Mybatis/MyBatisPlus+mysql
- MyBatis Generator/MyBatisPlus
- 模板引擎 FreeMarker、 velocity
- 接口文档 swagger

代码生成-myBatis

```
<artifactId>genDemo</artifactId>
<version>0.0.1-SNAPSHOT</version>
<name>genDemo</name>
<description>genDemo</description>
<properties>
  <maven.compiler.source>17</maven.compiler.source>
  <maven.compiler.target>17</maven.compiler.target>
  <mybatis.version>3.5.10</mybatis.version>
  <mybatis.generator.version>1.4.2</mybatis.generator.version>
  <mysql.version>8.0.33</mysql.version>
</properties>

<dependencies>
  <dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>${mybatis.version}</version>
  </dependency>
  <dependency>
    <groupId>org.mybatis.generator</groupId>
    <artifactId>mybatis-generator-core</artifactId>
    <version>1.4.2</version>
  </dependency>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>${mysql.version}</version>
  </dependency>
</dependencies>
```

```
CREATE TABLE user (
  id BIGINT AUTO_INCREMENT PRIMARY KEY,
  username VARCHAR(50) NOT NULL UNIQUE,
  email VARCHAR(100) NOT NULL,
  password VARCHAR(255) NOT NULL,
  created_time DATETIME DEFAULT CURRENT_TIMESTAMP,
  updated_time DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
  status TINYINT DEFAULT 1 COMMENT '0:禁用, 1:正常'
);

INSERT INTO user (username, email, password) VALUES
('zhangsan', 'zhangsan@example.com', '123456'),
('lisi', 'lisi@example.com', '123456');
```

代码生成-myBatis

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE generatorConfiguration
    PUBLIC "-//mybatis.org//DTD MyBatis Generator Configuration 1.0//EN"
    "http://mybatis.org/dtd/mybatis-generator-config_1_0.dtd">

<generatorConfiguration>
    <!-- 数据库驱动路径 -->
    <classPathEntry location="C:/path/to/mysql-connector-java-8.0.33.jar"/>

    <context id="MySQL" targetRuntime="MyBatis3">
        <!-- 生成的 Java 文件的编码 -->
        <property name="javaFileEncoding" value="UTF-8"/>

        <!-- 注释生成器 -->
        <commentGenerator>
            <property name="suppressDate" value="true"/>
            <property name="addRemarkComments" value="true"/>
        </commentGenerator>

        <!-- 数据库连接信息 -->
        <jdbcConnection>
            driverClass="com.mysql.cj.jdbc.Driver"
            connectionURL="jdbc:mysql://localhost:3306/test_db"
            userId="root"
            password="password">
        </jdbcConnection>

        <!-- Java 类型解析器 -->
        <javaTypeResolver>
            <property name="forceBigDecimals" value="false"/>
        </javaTypeResolver>
    </context>
</generatorConfiguration>
```

代码生成-myBatis

```
<!-- 生成实体类配置 -->
<javaModelGenerator targetPackage="com.example.entity" targetProject="src/main/java">
  <property name="enableSubPackages" value="true"/>
  <property name="trimStrings" value="true"/>
</javaModelGenerator>

<!-- 生成 SQL Map XML 文件配置 -->
<sqlMapGenerator targetPackage="mapper" targetProject="src/main/resources">
  <property name="enableSubPackages" value="true"/>
</sqlMapGenerator>

<!-- 生成 Mapper 接口配置 -->
<javaClientGenerator type="XMLMAPPER" targetPackage="com.example.mapper" targetProject="src/main/java">
  <property name="enableSubPackages" value="true"/>
</javaClientGenerator>

<!-- 指定要生成的表 -->
<table tableName="user" domainObjectName="User">
  <property name="useActualColumnNames" value="false"/>
  <!-- 主键自增 -->
  <generatedKey column="id" sqlStatement="MySql" identity="true"/>
  <!-- 字段映射配置 -->
  <columnOverride column="created_time" property="createTime"/>
  <columnOverride column="updated_time" property="updateTime"/>
</table>
</context>
</generatorConfiguration>
```

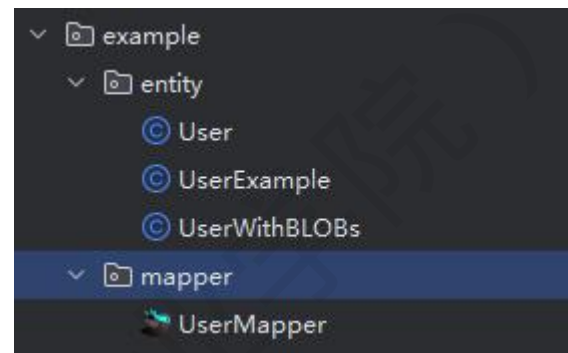
代码生成-myBatis

```
import org.mybatis.generator.api.ShellRunner;

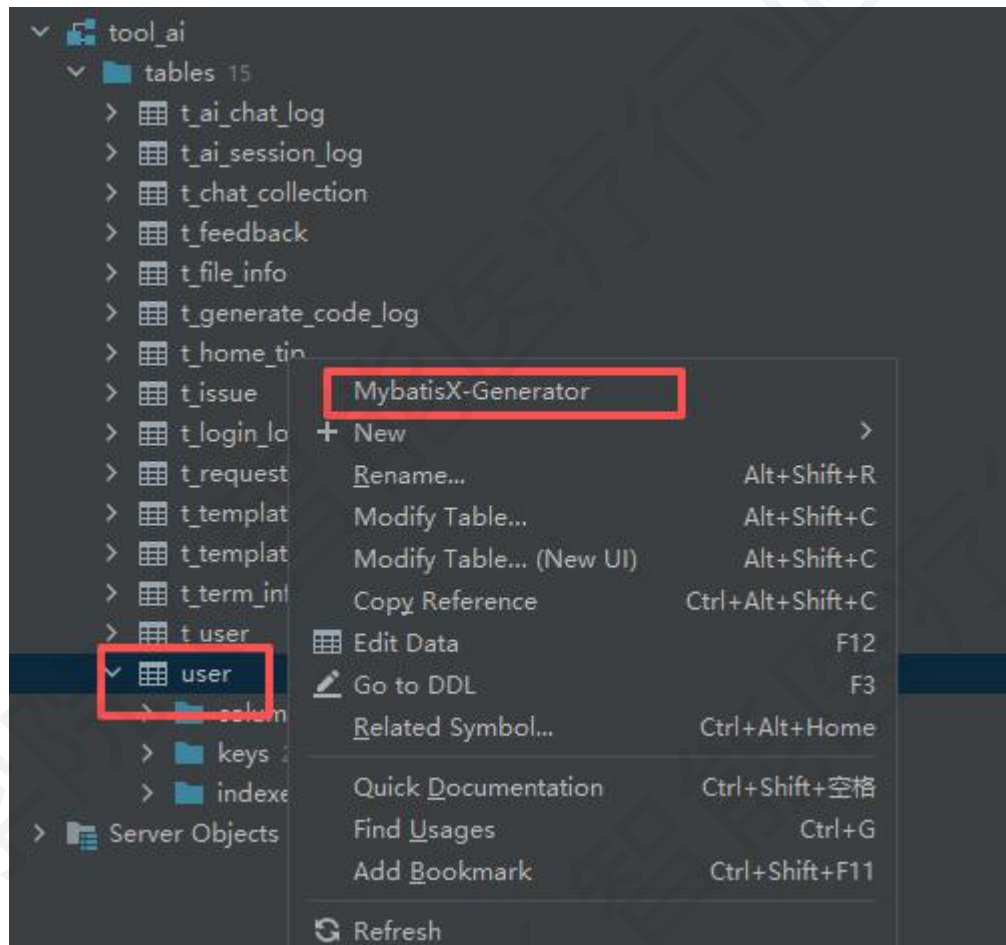
public class GenDemoApplication {

    public static void main(String[] args) {

        String config = "/generatorConfig.xml";
        String[] arg = {"-configfile", config, "-overwrite"};
        ShellRunner.main(arg);
    }
}
```



开发工具中的实现



开发工具中的实现

 Generate Options

annotation

☐ None

☒ Mybatis-Plus 2

☐ Mybatis-Plus 3

☐ JPA

options

☒ Comment

☐ Actual Column

☒ Model

☒ toString/hashCode>equals

☐ Actual Column Annotation

☐ Lombok

☐ JSR310: Date API

template

☐ custom-model-swagger

☒ mybatis-plus2

☐ default-all

☐ mybatis-plus3

☐ default-empty

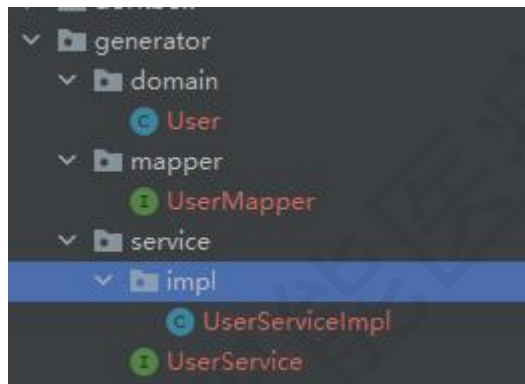
	config name	module path	base path	package name
	serviceImpl	E:/wl/work/project/jingfang-cloud...	src/main/java	generator.service.impl
	mapperInterface	E:/wl/work/project/jingfang-cloud...	src/main/java	generator.mapper
	serviceInterface	E:/wl/work/project/jingfang-cloud...	src/main/java	generator.service
	mapperXml	E:/wl/work/project/jingfang-cloud...	src/main/resources	mapper

Previous

Finish

Cancel

开发工具中的实现



```
9 usages
@TableName(value = "user")
public class User implements Serializable {
    /**
     *
     */
    3 usages
    @TableId
    private Long id;

    /**
     *
     */
    3 usages
    private String username;

    /**
     *
     */
    3 usages
    private String email;

    /**
     *
     */
    3 usages
    private String password;

    /**
     *
     */
    3 usages
    private Date createTime;
}
```

模板引擎-FreeMarker语法

程序开发指南

- + 入门
- + 数据模型
- + 配置(Configuration)
- + 其它

模板语言参考

- + 内建函数参考
- 指令参考
 - Alphabetical index
 - assign
 - attempt, recover
 - compress
 - escape, noescape
 - flush
 - ftl
 - function, return
 - global
 - if, else, elseif
 - import
 - include
 - list, else, items, sep, break
 - local
 - macro, nested, return
 - noparse
 - nt
 - setting
 - stop
 - switch, case, default, break
 - t, lt, rt
 - User-defined directive (<@...>)
 - visit, recurse, fallback

概要

```
<#if condition>
...
<#elseif condition2>
...
<#elseif condition3>
...
<#else>
...
</#if>
```

这里:

- `condition`, `condition2`, 等: 将被计算成布尔值的表达式。

`elseif` 和 `else` 是可选的。

描述

你可以使用 `if`, `elseif` 和 `else` 指令来条件判断是否越过模板的一个部分。 `condition` 必须计算成布尔值, 否则错误将会中止模板处理。 `elseif` 和 `else` 必须出现在 `if` 内部 (也就是, 在 `if` 的开始标签和结束标签之间)。 `if` 中可以包含任意数量的 `elseif` (包括0个) 而且结束时 `else` 是可选的。比如:

只有 `if` 没有 `elseif` 和 `else`:

```
TEMPLATE
<#if x == 1>
  x is 1
</#if>
```

只有 `if` 没有 `elseif` 但是有 `else`:

```
TEMPLATE
```

<http://freemarker.foofun.cn/>

案例演示

<https://boot3.jeecg.com/>

代码在线预览

```
ccc2
├── controller
│   └── HousekeepingAppointmentController.java
├── entity
│   └── HousekeepingAppointment.java
├── mapper
│   └── HousekeepingAppointmentMapper.java
│       └── xml
│           └── HousekeepingAppointmentMapper.xml
├── service
│   ├── IHousekeepingAppointmentService.java
│   └── impl
│       └── HousekeepingAppointmentServiceImpl.java
├── uniapp
│   ├── HousekeepingAppointmentForm.vue
│   └── HousekeepingAppointmentList.vue
├── uniapp3
│   ├── HousekeepingAppointmentData.ts
│   ├── HousekeepingAppointmentForm.vue
│   └── HousekeepingAppointmentList.vue
├── vue3
│   ├── HousekeepingAppointmentList.vue
│   ├── HousekeepingAppointment.api.ts
│   ├── HousekeepingAppointment.data.ts
│   └── V20251120_1_menu_insert_HousekeepingAppointment
└── components
    └── HousekeepingAppointmentForm.vue
```

```
46 */
47 @Tag(name="家政预约表")
48 @RestController
49 @RequestMapping("/ccc2/housekeepingAppointment")
50 @Slf4j
51 public class HousekeepingAppointmentController extends JeecgController<HousekeepingAppointment, IHousekeep:
52     @Autowired
53     private IHousekeepingAppointmentService housekeepingAppointmentService;
54
55 /**
56  * 分页列表查询
57  *
58  * @param housekeepingAppointment
59  * @param pageNo
60  * @param pageSize
61  * @param req
62  * @return
63  */
64 //@@AutoLog(value = "家政预约表-分页列表查询")
65 @Operation(summary="家政预约表-分页列表查询")
66 @GetMapping(value = "/list")
67 public Result<IPage<HousekeepingAppointment>> queryPageList(HousekeepingAppointment housekeepingAppointm
68     @RequestParam(name="pageNo", defaultValue="1") Integer pageNo,
69     @RequestParam(name="pageSize", defaultValue="10") Integer pageSize,
70     HttpServletRequest req) {
71
72
73     QueryWrapper<HousekeepingAppointment> queryWrapper = QueryGenerator.initQueryWrapper(housekeepingAp
74     Page<HousekeepingAppointment> page = new Page<HousekeepingAppointment>(pageNo, pageSize);
75     IPage<HousekeepingAppointment> pageList = housekeepingAppointmentService.page(page, queryWrapper);
76     return Result.OK(pageList);
77 }
78
79 /**
80  * 添加
```



使用大模型

- 怎么让大模型按照需求返回想要的结果呢
- 返回的结果我们应该处理



总结与实验预告

- 本节重点：
怎么生成通用性的代码
- 实验2预告：
和大模型融合生成业务代码

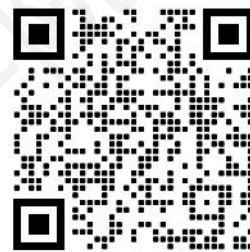


本章作业

- 自己使用MyBatisPlus生成以下CURD代码

信创智能医疗系统研发课程体系

河南中医药大学信息技术学院（智能医疗行业学院）



河南中医药大学信息技术学院（智能医疗行业学院）智能医疗教研室

河南中医药大学医疗健康信息工程技术研究所