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04章：使用若依开发框架

黄子杰
河南方和信息科技有限公司

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
2025/10/21

本章概要

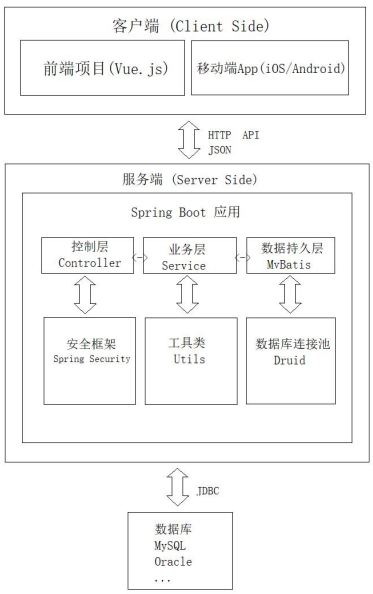
- 项目特点与优势
- 整体架构解析
  - 逻辑视图
- 目录结构
  - 前端
  - 后端
- 架构核心组成
  - 前端 (Front-end)
  - 后端 (Back-end)
  - 核心模块
- 前后端交互流程

项目特点与优势

- 职责分离： 前端专注于展示和交互，后端专注于业务逻辑和数据，分工明确，便于开发和维护。
- 技术栈灵活： 前后端可以独立选择技术栈，后端API可以同时为Web、App、小程序等多种客户端提供服务。
- 性能提升： 前端是静态资源，可以由CDN或高性能Web服务器承载，减轻后端服务器压力。
- 并行开发： 前后端开发人员可以并行工作，只需提前定义好API接口文档即可，提升开发效率。
- 高可扩展性： 微服务化改造时，后端可以很容易地拆分为多个独立的服务。

整体框架解析

逻辑视图



目录结构

前端目录结构

```
src/
    api/                        // 所有后端API的请求封装
    components/                 // 全局可复用的Vue组件
    layout/                     // 整体布局组件
    router/                     // 路由配置
    store/                      // Vuex状态管理
    utils/                      // 工具类，如request.js(封装Axios)
    views/                      // 页面视图组件
```


目录结构

后端目录结构

```
src/main/java/
    com.ruoyi.web.controller    // 控制层
    com.ruoyi.system.service    // 业务层接口
    com.ruoyi.system.service.impl // 业务层实现
    com.ruoyi.system.mapper     // 数据持久层 (MyBatis Mapper)
    com.ruoyi.system.domain     // 实体层
    com.ruoyi.framework.web     // Web相关配置(包括安全配置)
resources/
    mapper/                     // MyBatis的XML文件
    application.yml            // 主配置文件
```

核心架构组成

前端核心架构组成

- 前端 (Front-end)

  技术栈： Vue 2.x / Vue 3.x + Element UI + Axios + Vue Router + Vuex。

  核心职责：

  用户界面(UI)： 负责渲染所有页面，如表单、表格、图表等。

  用户交互： 处理用户的点击、输入等操作。

  路由管理： 通过Vue Router管理单页面应用(SPA)的路由跳转。

  状态管理： 使用Vuex管理全局状态，如用户登录信息、权限数据等。

  API调用： 使用Axios库发起HTTP请求，与后端RESTful API进行数据交互。

- 独立部署

  前端项目会使用npm run build打包成静态资源（HTML, CSS, JS），
  部署在Nginx或Apache等Web服务器上。

核心架构组成

后端核心架构组成

- 后端 (Back-end)

  技术栈： Spring Boot + Spring Security + MyBatis + Redis + Maven + Druid。

  分层架构：

  控制层(Controller)： 接收前端的HTTP请求，进行参数校验，调用业务层处理，并返回JSON格式的数据给前端。系统中使用了@RestController。

  业务层(Service)： 实现核心业务逻辑，处理来自控制层的请求，协调多个数据模型的操作。

  数据持久层(Mapper)： 基于MyBatis，负责与数据库进行交互，执行SQL查询、插入、更新和删除操作。

  实体层(Entity)： 对应数据库中的表结构，是数据传递的载体。

- 核心模块

  安全框架(Spring Security)： 若依的权限核心。

  用户认证(Authentication)： 验证用户名和密码（通常配合JWT令牌）。

  用户授权(Authorization)： 判断已登录的用户是否有权限访问某个API或操作某个菜单。通过@PreAuthorize注解实现方法级别的权限控制。

核心架构组成

后端核心架构组成

- 核心模块

  令牌机制(JWT)： 用户登录成功后，后端生成一个JWT令牌返回给前端。前端在后续的每次请求中都在HTTP Header中携带此令牌。后端通过验证JWT来确认用户身份，实现无状态的会话管理。

  缓存(Redis)： 用于缓存用户信息、字典数据、系统配置等，提升系统性能。同时也可用于管理在线用户等。

  数据库(MySQL)： 存储核心业务数据，如用户、角色、菜单、部门、日志等。

前后端交互流程

用户登录交互流程

- 请求：用户在登录页输入用户名和密码，点击登录。
- 前端处理： Vue组件通过Axios向后台发送一个POST请求，例如 /auth/login，并将用户名和密码放在请求体中。
- 后端验证：

  LoginController 接收到登录请求。

  Spring Security 的认证管理器进行用户名和密码校验。

  校验成功后，生成一个JWT令牌，并将用户权限等信息存入Redis。

  将用户信息、令牌等封装成JSON数据返回给前端。
- 前端接收：前端收到成功的响应后，将JWT令牌存储在本地（如LocalStorage或Vuex），并跳转到系统主页。
- 后续请求： 前端在调用任何其他API（如获取用户列表）时，自动在HTTP请求的Header中（通常是Authorization: Bearer <token>）附加JWT令牌。
- 权限校验： 后端的Spring Security过滤器会拦截每个请求，验证JWT令牌的有效性，并根据用户角色和权限判断是否允许访问该API。

需要注意的几点

关于权限

- 关于@PermitAll和@Anonymous

  在若依（RuoYi）框架中，权限控制是一个双重体系：

  1.Spring Security 的 URL 路径过滤

  2.Spring Security 方法级注解（如 @PreAuthorize）

- @PermitAll 注解（通常是 javax.annotation.security.PermitAll）只是一个标准的 JSR-250 注解，它本身并不能绕过 Spring Security 的 HTTP 请求级别安全。它主要被 @PreAuthorize 等注解的解析器识别，用于方法级别的权限控制。
- 当我们的请求到达时，会先经过 Spring Security 的过滤器链。如果过滤器链判断该请求需要认证，那么请求在到达我们的 @RestController 之前就会被拦截，根本不会有机会去检查 @PermitAll 注解。

需要注意的几点

测试工具

- http://localhost:8080/swagger-ui/index.html
- Authorize的获取

需要注意的几点


```
0 0 2 * * ?              每天凌晨2点执行
0 */5 * * * ?            每5分钟执行一次
0 0 18 * * MON-FRI 周一到周五晚上6点执行
0 0 12 1 * ?             每月1号中午12点执行
0 15 10 ? * 6L           每月最后一个周五早上10:15执行
秒 分 时 日 月 周 年(可选)
 * * * * * * *
 | | | | | | └── 年 (1970-2099)
 | | | | | └──────── 周 (1-7 或 SUN-SAT)
 | | | | └────────── 月 (1-12 或 JAN-DEC)
 | | | └──────────── 日 (1-31)
 | | └────────────── 时 (0-23)
 | └──────────────── 分 (0-59)
 └────────────────── 秒 (0-59)
```


本章作业

- 在本地初始化若依框架(前后端分离)系统

  要求:

  1.下载并初始化前端项目(包含各种组件)

  2.下载并初始化后端项目(maven)

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