项目git地址

<https://gitee.com/ybbag/deep-project.git>

项目分支说明正式（master），开发（dev）

## **技术选型**

1、系统环境

Java EE 8

Servlet 3.0

Apache Maven 3

2、主框架

Spring Boot 2.2.x

Spring Framework 5.2.x

Spring Security 5.2.x

3、持久层

Apache MyBatis 3.5.x

Hibernate Validation 6.0.x

Alibaba Druid 1.2.x

4、视图层

Vue 2.6.x

Axios 0.21.x

Element 2.15.x

## **功能**

用户管理：用户是系统操作者，该功能主要完成系统用户配置。

部门管理：配置系统组织机构（公司、部门、小组），树结构展现支持数据权限。

岗位管理：配置系统用户所属担任职务。

菜单管理：配置系统菜单，操作权限，按钮权限标识等。

角色管理：角色菜单权限分配、设置角色按机构进行数据范围权限划分。

字典管理：对系统中经常使用的一些较为固定的数据进行维护。

参数管理：对系统动态配置常用参数。

通知公告：系统通知公告信息发布维护。

操作日志：系统正常操作日志记录和查询；系统异常信息日志记录和查询。

登录日志：系统登录日志记录查询包含登录异常。

在线用户：当前系统中活跃用户状态监控。

定时任务：在线（添加、修改、删除)任务调度包含执行结果日志。

代码生成：前后端代码的生成（java、html、xml、sql)支持CRUD下载 。

系统接口：根据业务代码自动生成相关的api接口文档。

服务监控：监视当前系统CPU、内存、磁盘、堆栈等相关信息。

缓存监控：对系统的缓存信息查询，命令统计等。

在线构建器：拖动表单元素生成相应的Vue代码。

连接池监视：监视当期系统数据库连接池状态，可进行分析SQL找出系统性能瓶颈

合成功能基础数据：对所有query进行管理

合成功能模块拼接：对每个query进行视频拼接

合成内容：实现1、导入合成。2.模块拼接后最终合成

合成功能小模块视频管理：管理所有合成的小视频模块

合成功能小模块文字管理：管理所有对应的视频文字

合成功能医生视频管理：管理所有合成中拼接的医生视频

合成功能封面图管理：关联最终合成的视频的封面图（内置会把图片转成mp4）

合成功能医生管理：合成视频关联的医生关系

合成功能病种分类：合成视频关联的病种分类

合成功能资源库管理：只是把连接加上可以进行管理深读本地的数据图

# **环境部署**

JDK >= 1.8 (推荐1.8版本)

Mysql >= 5.7.0 (推荐5.7版本)

Redis >= 3.0

Maven >= 3.0

Node >= 12

不建议使用cnpm（可能会出现奇怪的问题）

## **Nginx配置**

worker\_processes 1;

events {

worker\_connections 1024;

}

http {

include mime.types;

default\_type application/octet-stream;

sendfile on;

keepalive\_timeout 65;

server {

listen 80;

server\_name localhost;

charset utf-8;

location / {

root /home/ruoyi/projects/ruoyi-ui;

try\_files $uri $uri/ /index.html;

index index.html index.htm;

}

location /prod-api/ {

proxy\_set\_header Host $http\_host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header REMOTE-HOST $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_pass http://localhost:8080/;

}

error\_page 500 502 503 504 /50x.html;

location = /50x.html {

root html;

}

}

}

# **项目介绍**

### **后端结构**

com.ruoyi

├── common // 工具类

│ └── annotation // 自定义注解

│ └── config // 全局配置

│ └── constant // 通用常量

│ └── core // 核心控制

│ └── enums // 通用枚举

│ └── exception // 通用异常

│ └── filter // 过滤器处理

│ └── utils // 通用类处理

├── framework // 框架核心

│ └── aspectj // 注解实现

│ └── config // 系统配置

│ └── datasource // 数据权限

│ └── interceptor // 拦截器

│ └── manager // 异步处理

│ └── security // 权限控制

│ └── web // 前端控制

├── ruoyi-generator // 代码生成

├── ruoyi-quartz // 定时任务

├── ruoyi-system // 系统代码

├── ruoyi-admin // 后台服务

├── ruoyi-sdgather // 合成服务

|—— ruoyi-chatgpt // chatgpt 服务

### **前端结构**

├── build // 构建相关

├── bin // 执行脚本

├── public // 公共文件

│ ├── favicon.ico // favicon图标

│ └── index.html // html模板

│ └── robots.txt // 反爬虫

├── src // 源代码

│ ├── api // 所有请求

│ ├── assets // 主题 字体等静态资源

│ ├── components // 全局公用组件

│ ├── directive // 全局指令

│ ├── layout // 布局

│ ├── plugins // 通用方法

│ ├── router // 路由

│ ├── store // 全局 store管理

│ ├── utils // 全局公用方法

│ ├── views // view

│ ├── App.vue // 入口页面

│ ├── main.js // 入口 加载组件 初始化等

│ ├── permission.js // 权限管理

│ └── settings.js // 系统配置

├── .editorconfig // 编码格式

├── .env.development // 开发环境配置

├── .env.production // 生产环境配置

├── .env.staging // 测试环境配置

├── .eslintignore // 忽略语法检查

├── .eslintrc.js // eslint 配置项

├── .gitignore // git 忽略项

├── babel.config.js // babel.config.js

├── package.json // package.json

└── vue.config.js // vue.config.js

## **核心技术**

前端技术栈 ES6、vue、vuex、vue-router、vue-cli、axios、element-ui

后端技术栈 SpringBoot、MyBatis、Spring Security、Jwt