

项目介绍

文件结构

后端结构

```
1  com.ruoyi
2  |— common          // 工具类
3  |   |— annotation   // 自定义注解
4  |   |— config       // 全局配置
5  |   |— constant     // 通用常量
6  |   |— core         // 核心控制
7  |   |— enums        // 通用枚举
8  |   |— exception    // 通用异常
9  |   |— filter       // 过滤器处理
10 |   |— utils        // 通用类处理
11 |— framework       // 框架核心
12 |   |— aspectj      // 注解实现
13 |   |— config       // 系统配置
14 |   |— datasource   // 数据权限
15 |   |— interceptor  // 拦截器
16 |   |— manager      // 异步处理
17 |   |— security     // 权限控制
18 |   |— web          // 前端控制
19 |— ruoyi-generator  // 代码生成（可移除）
20 |— ruoyi-quartz    // 定时任务（可移除）
21 |— ruoyi-system    // 系统代码
22 |— ruoyi-admin     // 后台服务
23 |— ruoyi-xxxxxx    // 其他模块
```

前端结构

```
1  |— build           // 构建相关
2  |— bin            // 执行脚本
3  |— public         // 公共文件
4  |   |— favicon.ico // favicon图标
5  |   |— index.html  // html模板
```

	≡	RuoYi	
8		assets	// 主题 字体等静态资源
9		components	// 全局公用组件
10		directive	// 全局指令
11		layout	// 布局
12		plugins	// 通用方法
13		router	// 路由
14		store	// 全局 store管理
15		utils	// 全局公用方法
16		views	// view
17		App.vue	// 入口页面
18		main.js	// 入口 加载组件 初始化等
19		permission.js	// 权限管理
20		settings.js	// 系统配置
21		.editorconfig	// 编码格式
22		.env.development	// 开发环境配置
23		.env.production	// 生产环境配置
24		.env.staging	// 测试环境配置
25		.eslintignore	// 忽略语法检查
26		.eslintrc.js	// eslint 配置项
27		.gitignore	// git 忽略项
28		babel.config.js	// babel.config.js
29		package.json	// package.json
30		vue.config.js	// vue.config.js
31			

配置文件

通用配置 application.yml

```
1 # 项目相关配置
2 ruoyi:
3   # 名称
4   name: RuoYi
5   # 版本
6   version: 3.3.0
7   # 版权年份
8   copyrightYear: 2021
9   # 实例演示开关
10  demoEnabled: true
11  # 文件路径 示例 ( Windows配置D:/ruoyi/uploadPath, Linux配置 /home/ruoyi/uploadPath )
12  profile: D:/ruoyi/uploadPath
```

15	<code># 验证码类型 math 数组计算 captcha 与 验证码</code>
16	<code>captchaType: math</code>
17	
18	<code># 开发环境配置</code>
19	<code>server:</code>
20	<code># 服务器的HTTP端口，默认为8080</code>
21	<code>port: 8080</code>
22	<code>servlet:</code>
23	<code># 应用的访问路径</code>
24	<code>context-path: /</code>
25	<code>tomcat:</code>
26	<code># tomcat的URI编码</code>
27	<code>uri-encoding: UTF-8</code>
28	<code># tomcat最大线程数，默认为200</code>
29	<code>max-threads: 800</code>
30	<code># Tomcat启动初始化的线程数，默认值25</code>
31	<code>min-spare-threads: 30</code>
32	
33	<code># 日志配置</code>
34	<code>logging:</code>
35	<code>level:</code>
36	<code>com.ruoyi: debug</code>
37	<code>org.springframework: warn</code>
38	
39	<code># Spring配置</code>
40	<code>spring:</code>
41	<code># 资源信息</code>
42	<code>messages:</code>
43	<code># 国际化资源文件路径</code>
44	<code>basename: i18n/messages</code>
45	<code>profiles:</code>
46	<code>active: druid</code>
47	<code># 文件上传</code>
48	<code>servlet:</code>
49	<code>multipart:</code>
50	<code># 单个文件大小</code>
51	<code>max-file-size: 10MB</code>
52	<code># 设置总上传的文件大小</code>
53	<code>max-request-size: 20MB</code>
54	<code># 服务模块</code>
55	<code>devtools:</code>
56	<code>restart:</code>
57	<code># 热部署开关</code>
58	<code>enabled: true</code>



```
61 # 地址
62 host: localhost
63 # 端口，默认为6379
64 port: 6379
65 # 数据库索引
66 database: 0
67 # 密码
68 password:
69 # 连接超时时间
70 timeout: 10s
71 lettuce:
72   pool:
73     # 连接池中的最小空闲连接
74     min-idle: 0
75     # 连接池中的最大空闲连接
76     max-idle: 8
77     # 连接池的最大数据库连接数
78     max-active: 8
79     # #连接池最大阻塞等待时间（使用负值表示没有限制）
80     max-wait: -1ms
81
82 # token配置
83 token:
84   # 令牌自定义标识
85   header: Authorization
86   # 令牌密钥
87   secret: abcdefghijklmnopqrstuvwxyz
88   # 令牌有效期（默认30分钟）
89   expireTime: 30
90
91 # MyBatis配置
92 mybatis:
93   # 搜索指定包别名
94   typeAliasesPackage: com.ruoyi.**.domain
95   # 配置mapper的扫描，找到所有的mapper.xml映射文件
96   mapperLocations: classpath*:mapper/**/*.xml
97   # 加载全局的配置文件
98   configLocation: classpath:mybatis/mybatis-config.xml
99
100 # PageHelper分页插件
101 pagehelper:
102   helperDialect: mysql
103   reasonable: true
104   supportMethodsArguments: true
```



```
107 // Swagger 配置
108 swagger:
109     # 是否开启swagger
110     enabled: true
111     # 请求前缀
112     pathMapping: /dev-api
113
114 # 防止XSS攻击
115 xss:
116     # 过滤开关
117     enabled: true
118     # 排除链接（多个用逗号分隔）
119     excludes: /system/notice/*
120     # 匹配链接
121     urlPatterns: /system/*,/monitor/*,/tool/*
```

数据源配置 application-druid.yml

```
1 # 数据源配置
2 spring:
3     datasource:
4         type: com.alibaba.druid.pool.DruidDataSource
5         driverClassName: com.mysql.cj.jdbc.Driver
6         druid:
7             # 主库数据源
8             master:
9                 url: jdbc:mysql://localhost:3306/ry-vue?useUnicode=true&charac
10                 username: root
11                 password: password
12             # 从库数据源
13             slave:
14                 # 从数据源开关/默认关闭
15                 enabled: false
16                 url:
17                 username:
18                 password:
19             # 初始连接数
20             initialSize: 5
21             # 最小连接池数量
22             minIdle: 10
23             # 最大连接池数量
24             maxActive: 20
```



```
27 # 配置间隔多久进行一次检测，间隔多久又会自动加入被污染的连接，默认为毫秒
28 timeBetweenEvictionRunsMillis: 60000
29 # 配置一个连接在池中最小生存的时间，单位是毫秒
30 minEvictableIdleTimeMillis: 300000
31 # 配置一个连接在池中最大生存的时间，单位是毫秒
32 maxEvictableIdleTimeMillis: 900000
33 # 配置检测连接是否有效
34 validationQuery: SELECT 1 FROM DUAL
35 testWhileIdle: true
36 testOnBorrow: false
37 testOnReturn: false
38 webStatFilter:
39     enabled: true
40 statViewServlet:
41     enabled: true
42     # 设置白名单，不填则允许所有访问
43     allow:
44     url-pattern: /druid/*
45     # 控制台管理用户名和密码
46     login-username:
47     login-password:
48 filter:
49     stat:
50         enabled: true
51         # 慢SQL记录
52         log-slow-sql: true
53         slow-sql-millis: 1000
54         merge-sql: true
55     wall:
56         config:
57             multi-statement-allow: true
```

代码生成配置 generator.yml

```
1 # 代码生成
2 gen:
3     # 作者
4     author: ruoyi
5     # 默认生成包路径 system 需改成自己的模块名称 如 system monitor tool
6     packageName: com.ruoyi.system
7     # 自动去除表前缀，默认是false
8     autoRemovePre: false
```



核心技术

TIP

- 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

后端技术

SpringBoot框架

1、介绍

Spring Boot 是一款开箱即用框架，提供各种默认配置来简化项目配置。让我们的 Spring 应用变的更轻量化、更快的入门。在主程序执行 main 函数就可以运行。你也可以打包你的应用为 jar 并通过使用 java -jar 来运行你的Web应用。它遵循"约定优先于配置"的原则，使用 SpringBoot 只需很少的配置，大部分的时候直接使用默认的配置即可。同时可以与 Spring Cloud 的微服务无缝结合。

提示

Spring Boot2.x 版本环境要求必须是 jdk8 或以上版本，服务器 Tomcat8 或以上版本

2、优点

- 使编码变得简单：推荐使用注解。
- 使配置变得简单：自动配置、快速集成新技术能力 没有冗余代码生成和XML配置的要求
- 使部署变得简单：内嵌Tomcat、Jetty、Undertow等web容器，无需以war包形式部署
- 使监控变得简单：提供运行时的应用监控
- 使集成变得简单：对主流开发框架的无配置集成。
- 使开发变得简单：极大地提高了开发快速构建项目、部署效率。

Spring Security安全控制

三 RuoYi

决方案的安全框架。

2、功能

Authentication 认证，就是用户登录

Authorization 授权，判断用户拥有什么权限，可以访问什么资源
安全防护，跨站脚本攻击， session 攻击等
非常容易结合 Spring 进行使用

3、 Spring Security 与 Shiro 的区别

相同点

- 1、认证功能
- 2、授权功能
- 3、加密功能
- 4、会话管理
- 5、缓存支持
- 6、rememberMe功能

....

不同点

优点：

- 1、Spring Security基于Spring开发，项目如果使用Spring作为基础，配合Spring Security做权限更加方便。而Shiro需要和Spring进行整合开发
- 2、Spring Security功能比Shiro更加丰富，例如安全防护方面
- 3、Spring Security社区资源相对比Shiro更加丰富

缺点：

- 1) Shiro的配置和使用比较简单，Spring Security上手复杂些
- 2) Shiro依赖性低，不需要依赖任何框架和容器，可以独立运行。Spring Security依赖Spring容器

前端技术

- npm：node.js的包管理工具，用于统一管理我们前端项目中需要用到的包、插件、工具、命令等，便于开发和维护。

≡

RuoYi

- vue-cli：Vue的脚手架工具，用于自动生成Vue项目的目录及文件。
- vue-router：Vue提供的前端路由工具，利用其我们实现页面的路由控制，局部刷新及按需加载，构建单页应用，实现前后端分离。
- vuex：Vue提供的状态管理工具，用于统一管理我们项目中各种数据的交互和重用，存储我们需要用到数据对象。
- element-ui：基于MVVM框架Vue开源出来的一套前端ui组件。

← 环境部署

后台手册 →