

Springboot vue 前后端项目

前端项目构建

```
npm install -g vue-cli
npm i -g @vue/cli-init
vue init webpack {项目名}
cd {项目名}
npm run dev
npm i element-ui -S
npm i axios -S
```

Main.js

引入 element

vue.prototype 封装请求

配置请求转发 config目录下的index proxyTable

动态路由 路由全局守卫

菜单渲染

后端项目构建

新建一个SpringBoot 项目

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
  <version>2.5.5</version>
</dependency>
<dependency>
  <groupId>org.mybatis.spring.boot</groupId>
  <artifactId>mybatis-spring-boot-starter</artifactId>
  <version>2.2.0</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
  <version>2.5.5</version>
</dependency>
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-data-redis</artifactId>
  <exclusions>
    <exclusion>
      <groupId>io.lettuce</groupId>
      <artifactId>lettuce-core</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

```

<dependency>
    <groupId>redis.clients</groupId>
    <artifactId>jedis</artifactId>
    <version>3.7.0</version>
</dependency>
<dependency>
    <groupId>com.alibaba</groupId>
    <artifactId>druid</artifactId>
    <version>1.2.6</version>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-cache</artifactId>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.26</version>
</dependency>

```

```

# Mysql
spring.datasource.type=com.alibaba.druid.pool.DruidDataSource
spring.datasource.url=jdbc:mysql://localhost:3306/sysmedia?
serverTimezone=UTC&useUnicode=true&characterEncoding=utf8&useSSL=false
spring.datasource.username=root
spring.datasource.password=123456
# mybatis
mybatis.config-location=classpath:/mybatis-config.xml
#redis
spring.redis.database=0
spring.redis.host=
spring.redis.port=6379
spring.redis.password=123456
spring.redis.jedis.pool.max-active=8
spring.redis.jedis.pool.max-idle=8
spring.redis.lettuce.pool.max-wait=-1ms
spring.redis.jedis.pool.min-idle=0
# 缓存
spring.cache.cache-names=menus_cache
spring.cache.redis.time-to-live=1800s
#端口
server.port=8888

```

SpringSecurity 类

1. userdetail的重写
2. 动态配置权限 FilterInvocationSecurityMetadataSource 和 AccessDecisionManager
3. 返回授权是失败的信息 重写 AccessDeniedHandler
4. 配置 Spring Security 继承 WebSecurityConfigurationAdapter @EnableGlobalMethodSecurity
5. 在Security配置类中注入 userdetail和passwordencoder

缓存菜单

1. @Service
2. @Transactional

3. @CacheConfig(name="")
4. 方法上@CacheEnable(key="#root.menuName")

email

子线程发送

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-mail</artifactId>
</dependency>
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
```

在线聊天

webSocket

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-websocket</artifactId>
</dependency>
```