**权限系统与RBAC模型设计实现**

摘要:

权限系统是一个基于BS架构的软件系统中必要的非功能性需求。通过对权限系统的开发，拉通系统设计方法和代码开发技术的落地，完成软件设计的全流程拉通。体现在如下几个方面：项目管理、权限模型技术现状、需求指定、逻辑建模、技术选型、架构规划、代码实施、上线部署、开源项目。

**关键词：**

# 项目管理

## Git管理文档

此处的项目管理是指个人基于此功能的全流程开发管理，以顶层设计到逐步细化为指导思想层层设计实现，在这个过程中管理好各种文档、使用工具、代码。主要借助于Git来管理文档和代码，工具使用硬盘存储。

Git

GitHub:1102739617@qq.com

约束

|  |  |  |
| --- | --- | --- |
| 文件类型 | 文档格式 | 存储 |
| 技术调研文档 | docx | 本文件 |
| 需求文档 | docx | 本文件 |
| 设计文件 | 具体格式而定 | Git |
| 代码 | 具体格式而定 | Git |

# Spring

## 获取自定义的注解类

在spring 项目启动刚刚完成后，找到添加了权限注解的类，并找出添加了权限控制注解的注释。

1. 使用applicationContext 监听器来处理
2. 使用后处理bean 来处理

利用Spring 管理每个bean的生命周期来后处理，这样权限数据会累加生成，要创建一个静态的变量来存放数据才可以。

# Spring boot

新建一个项目用作前端拉通后端调用内存数据使用。Rbacdev环境提供接口服务，数据以json格式发送和接受。下面调试环境：

1. 关闭Spring boot的安全验证。
2. 建立正对json格式的请求数据服务。
3. <https://blog.csdn.net/qq_28379809/article/details/87995524>

## yml配置

application-｛profile｝.yml格式

application.yml 引用如下格式

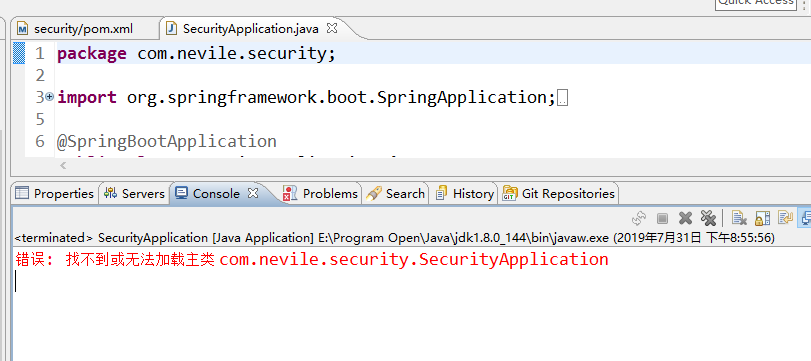
|  |
| --- |
| spring:  profiles:  active: dev |

## 日志

可参考这篇博客：<https://www.cnblogs.com/bigdataZJ/p/springboot-log.html>

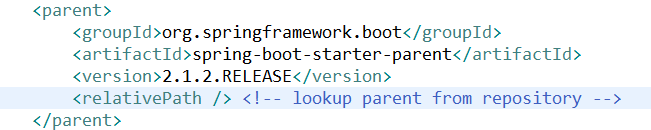
## 常见报错

使用 STS 块速构建项目，报错如下：



错误: 找不到或无法加载主类 com.nevile.security.SecurityApplication

解决：对比之前boot项目的版本2.1.2，发现现在的项目版本是2.1.6，导致启动报错，修改为2.1.2启动OK，很无语。



杀掉占用8080端口的进程：

netstat -aon|findstr "8080"

taskkill /f /pid  6764

## Spring Security

### 添加

添加，当然是为了使用。直接添加依赖，由于Spring Boot使用类自动配置，其他无需操心。

### 禁用

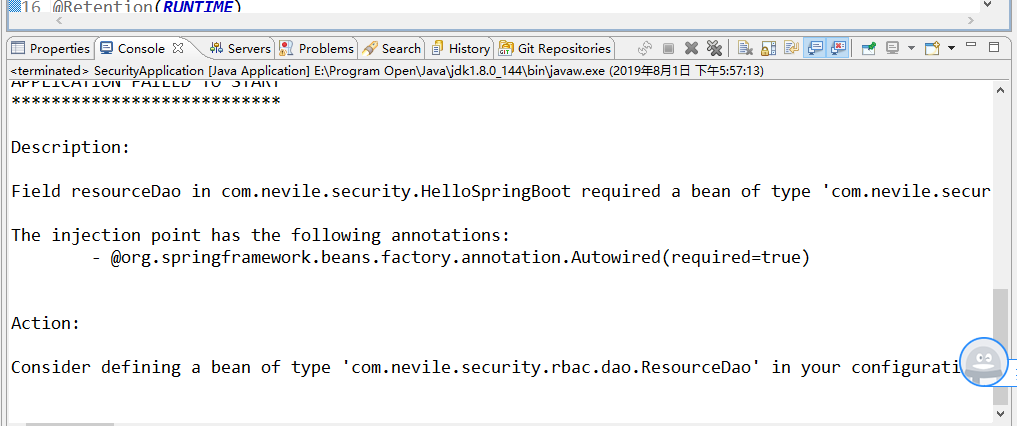
为了方便调试其他不设计安全特性的功能，或则方便安全模块功能开发，有禁用Security的需求。禁用如下：

|  |
| --- |
| springboot引入security依赖之后，默认访问资源需要登陆，不想用的话可以禁用，有两种方式  1、启动方法上如下注解  @EnableAutoConfiguration(exclude = {SecurityAutoConfiguration.class})  1  2、启动方法上如下注解  @SpringBootApplication(exclude = {SecurityAutoConfiguration.class })  ---------------------  作者：zgq\_hw  来源：CSDN  原文：https://blog.csdn.net/zgq\_hw/article/details/89475517  版权声明：本文为博主原创文章，转载请附上博文链接！ |

## Mybastis

1、启动依赖不要随便改

这个原因是我自己使用了自己的配置类，在启动main中排出了默认配置导致启动异常。



1. 在Oralc中的SQL语句注意

一段SQL结束后不要使用分号”;”，否则会报错无效字符。

使用默认的连接池如下：在Oracl的测试语句中使用select \* from dual

|  |
| --- |
| spring:  datasource:  username: springframwork  password: springframwork  url: jdbc:oracle:thin:@(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST = 127.0.0.1)(PORT = 1521))(CONNECT\_DATA =(SERVER = DEDICATED)(SERVICE\_NAME = orcl)))  driver-class-name: oracle.jdbc.driver.OracleDriver  type: com.zaxxer.hikari.HikariDataSource  hikari:  minimum-idle: 5  maximum-pool-size: 15  auto-commit: true  idle-timeout: 30000  pool-name: DatebookHikariCP  max-lifetime: 1800000  connection-timeout: 30000  connection-test-query: SELECT 1 from dual |

## Spring MVC

RESTFULL参数提交

<https://blog.csdn.net/rocling/article/details/82903574>

# SAP

使用PowerDesinger设计后台表，当然是用概览概览模型、逻辑模型、物理模型散步走的方式进行设计。

网络资料可参考：

<https://blog.csdn.net/angel_g/article/details/62418890>

添加注释：

<https://www.cnblogs.com/tymonyang/p/3916230.html>

## 权限管理模型

实现开发过程中添加注解，项目启动过程中自动刷新资源，管理员配置角色，用户登录后台验证资源所属的角色，进而保护资源（方法）。

1、想办法登记资源路径

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| spring security动态配置url权限认证 | <https://www.jianshu.com/p/0a06496e75ea> |  |  |
| 拦截器（Interceptor）和过滤器（Filter）的执行顺序和区别 | <https://blog.csdn.net/zxd1435513775/article/details/80556034> |  |  |
| spring security实现动态配置url权限的两种方法 | <https://www.cnblogs.com/xiaoqi/p/spring-security-rabc.html> |  |  |

## 菜单管理模型

# Eclipse的使用

## 代码注释

注释模板：

快捷键为：Alt+Shift+J

模板导入：**Window --> Preferences --> Java --> Code Style --> Code Templates --> Comments --> types --> Import...**

参考博客：<https://www.cnblogs.com/chenmingjun/p/8876837.html>

使用方法：鼠标点击到要注释的主体，按快捷键

## 批量修改变量

当前页面批量修改变量名称：shift+alt+r

全局修改方法名称：Ctrl+Alt+H，选中方法声明时的类文件中的位置，然后点击快捷键shift+alt+r

# 操作系统

## 关闭端口

1. Windows

netstat -nao | findstr “8080”

tasklist | findstr “3017”

# 参考文献