1. **技术框架**
2. 后端

Jdk 1.8

核心框架：Spring boot

安全框架：spring security

持久层框架：Mybatis

数据库连接池：Alibaba Druid

缓存框架：redis

日志框架：SL4J

前后端交互：json

1. 前端

Jquery

Html

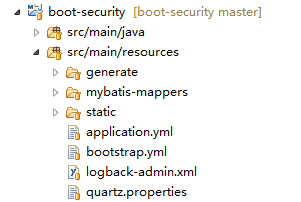
样式、弹窗：Layui

样式：Bootstrap

树形：Ztree

分页：Datatables

1. **配置文件说明**
2. 一览配置文件



1. application.yml

此文件为spring boot的基础配置文件，配置了数据源、上传文件大小、mybatis、上传文件路径、日志级别、日志目录、token过期时间、端口号，以及自定义配置参数。

1. bootstrap.yml

此配置同为spring boot的基础配置文件，加载优先于application.xml可以理解为，此文件配置系统级参数，application.yml配置应用级参数。

1. spring security配置

com.boot.security.server.config.SecurityConfig



com.boot.security.server.config.SecurityHandlerConfig

该类配置登陆成功、登陆失败、未登录返回401、退出，处理器。

1. logback-admin.xml

日志配置，此文件中的变量引用application.yml中的自定义参数。



将日志分为了三类



admin-server.log 可输出业务参数，可根据业务场景进行再分包

admin-server.log.error 所有的异常日志

admin-server.log.sql 通过mybatis生成的sql语句、参数、返回个数

1. 跨域配置



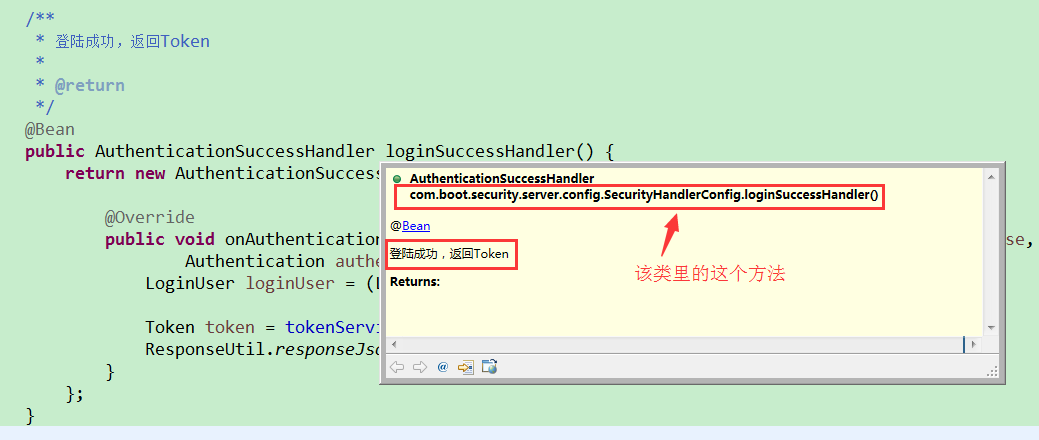
1. 全局统一的异常处理



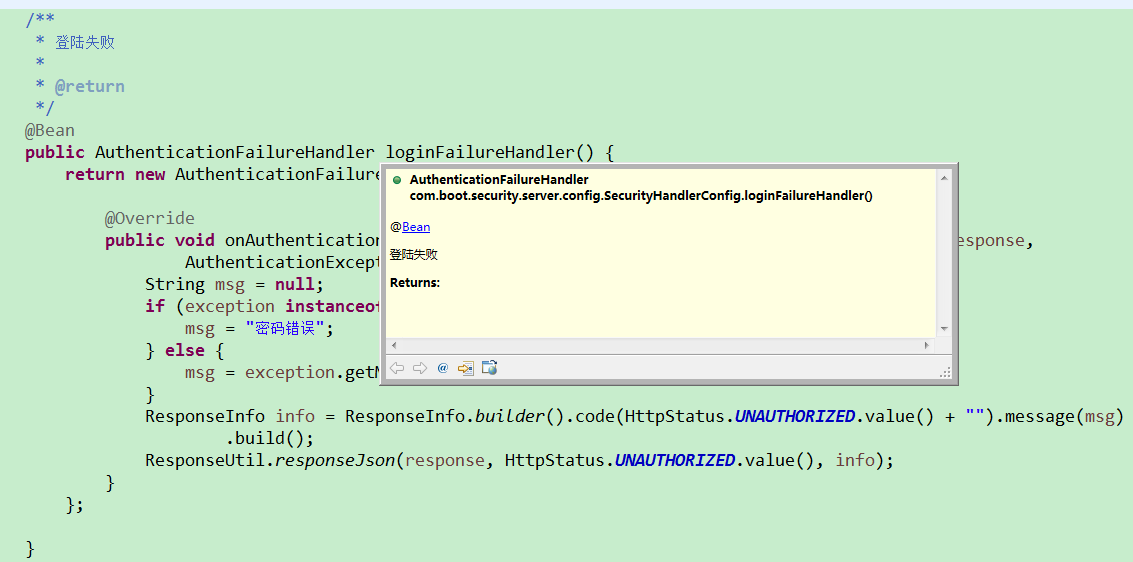
1. **登陆逻辑**
2. 登陆第一步



1. 登陆成功



1. 登陆失败



1. 校验密码

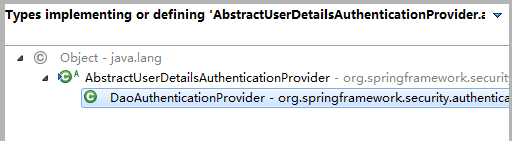
很多同学没发现密码校验在哪，其实这块是spring security底层做的校验，

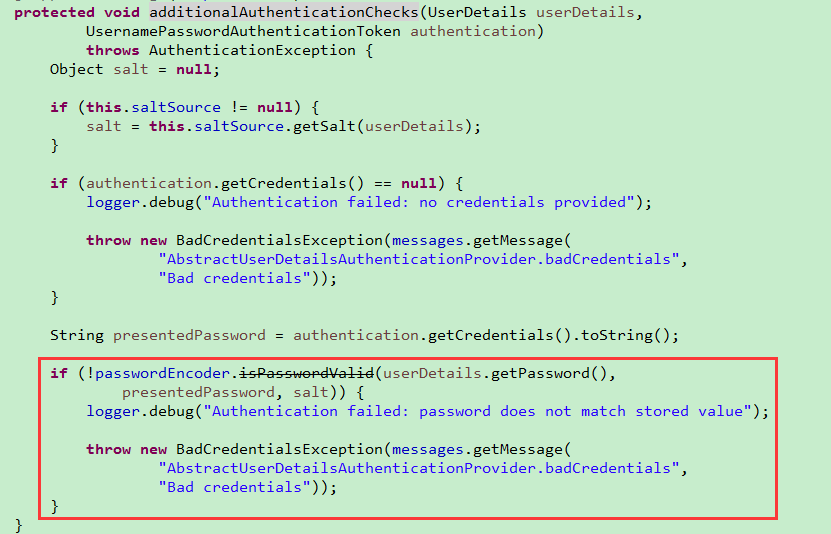
可以沿着源码查看类

[org](eclipse-javadoc:%E2%98%82=0-center/C:%5C/Users%5C/wei.zhang1%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/4.2.3.RELEASE%5C/spring-security-core-4.2.3.RELEASE.jar%3Corg).[springframework](eclipse-javadoc:%E2%98%82=0-center/C:%5C/Users%5C/wei.zhang1%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/4.2.3.RELEASE%5C/spring-security-core-4.2.3.RELEASE.jar%3Corg.springframework).[security](eclipse-javadoc:%E2%98%82=0-center/C:%5C/Users%5C/wei.zhang1%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/4.2.3.RELEASE%5C/spring-security-core-4.2.3.RELEASE.jar%3Corg.springframework.security).[authentication](eclipse-javadoc:%E2%98%82=0-center/C:%5C/Users%5C/wei.zhang1%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/4.2.3.RELEASE%5C/spring-security-core-4.2.3.RELEASE.jar%3Corg.springframework.security.authentication).[dao](eclipse-javadoc:%E2%98%82=0-center/C:%5C/Users%5C/wei.zhang1%5C/.m2%5C/repository%5C/org%5C/springframework%5C/security%5C/spring-security-core%5C/4.2.3.RELEASE%5C/spring-security-core-4.2.3.RELEASE.jar%3Corg.springframework.security.authentication.dao).AbstractUserDetailsAuthenticationProvider里的方法



这个方法的实现





1. **Token实现类**

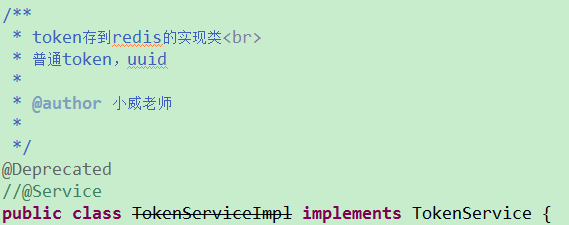
接口[com](eclipse-javadoc:%E2%98%82=boot-security-jpa/src%5C/main%5C/java%3Ccom).[boot](eclipse-javadoc:%E2%98%82=boot-security-jpa/src%5C/main%5C/java%3Ccom.boot).[security](eclipse-javadoc:%E2%98%82=boot-security-jpa/src%5C/main%5C/java%3Ccom.boot.security).[server](eclipse-javadoc:%E2%98%82=boot-security-jpa/src%5C/main%5C/java%3Ccom.boot.security.server).[service](eclipse-javadoc:%E2%98%82=boot-security-jpa/src%5C/main%5C/java%3Ccom.boot.security.server.service).TokenService

1. redis实现

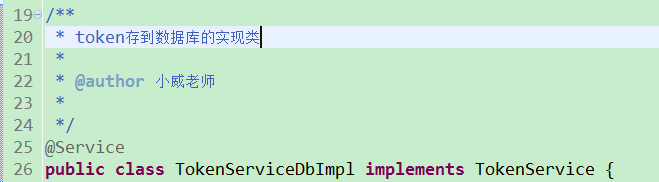
JWT



简单uuid，已作废



1. 数据库实现



1. 切换到数据库存储

根据上图发现，redis实现比数据库实现的service多了一个注解@Primary，那么只要将com.boot.security.server.service.impl.TokenServiceJWTImpl类上的注解@Primary挪到com.boot.security.server.service.impl.TokenServiceDbImpl即可。或者直接干掉TokenServiceJWTImpl类上的注释



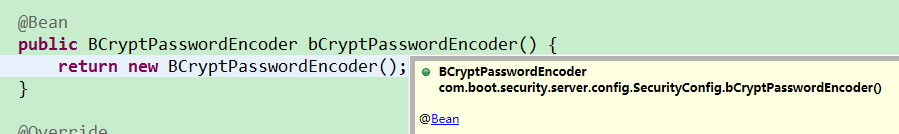
其实接口有多个实现类，并且都@Service了的话

@Autowired

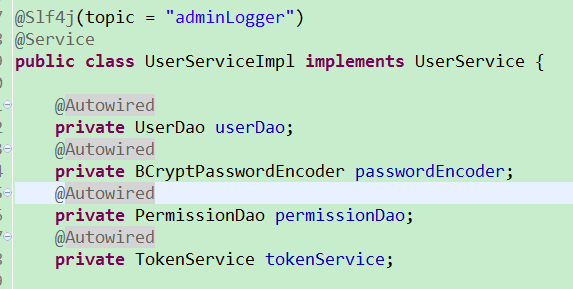
**private** TokenService tokenService;

**自动注入起作用的就是带@Primary的实现类。**

1. **密码加密**

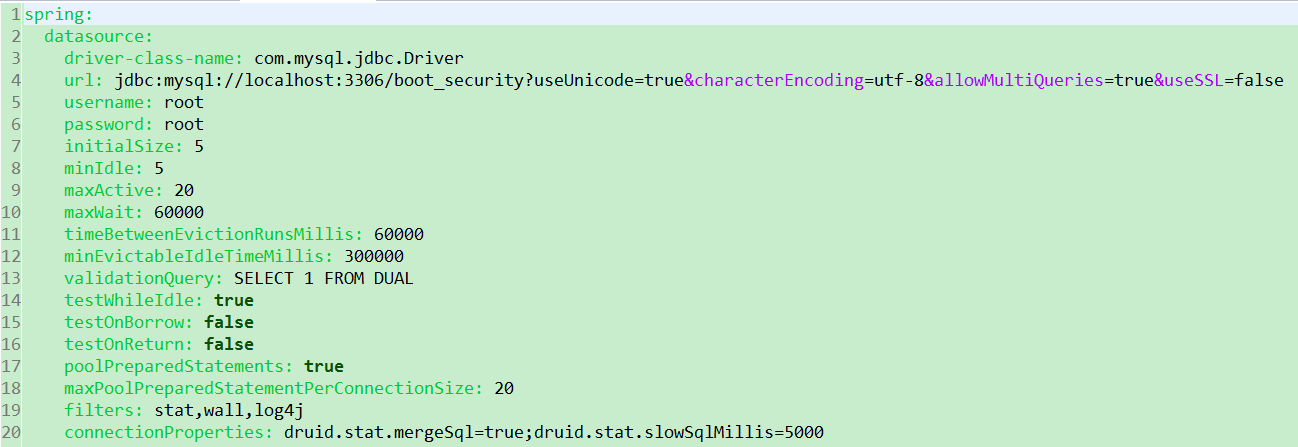


具体参考添加用户和修改密码的UserServiceImp.java



1. **详细配置**
2. 数据源配置

在application.yml中用的是阿里巴巴的druid，主要会修改端口号，数据库名，用户名和密码。



对应java类

[com](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom).[boot](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw).security.boot.[config](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server.config).DruidConfig



1. Mybatis配置

在application.yml中



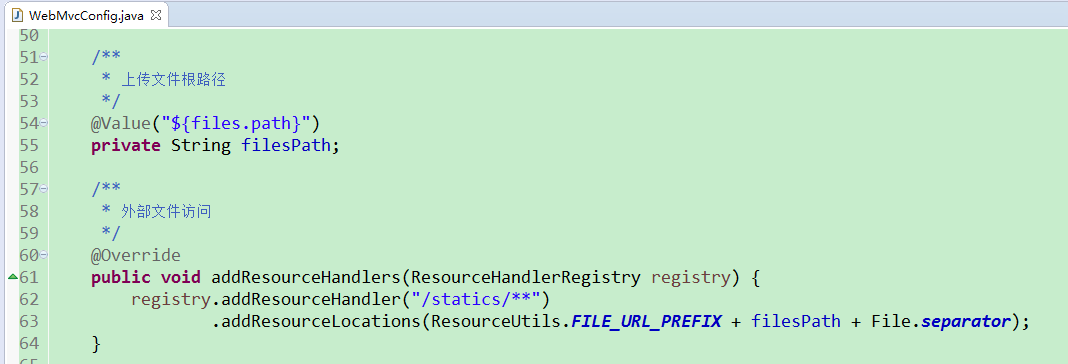
type-aliases-package，配置类别名的包，如有多个包，用逗号分开继续写，如com.boot.security.service.model, com.boot.security.service.test.model

mapper-locations，配置复杂sql的Mapper文件，简单sql直接用注解写在dao文件里面了。

1. 上传文件



上传文件会存储在该路径下，对应



如d:/files/a.png,访问路径就是<http://localhost:8080/statics/a.png>



此处配置springmvc上传文件大小限制

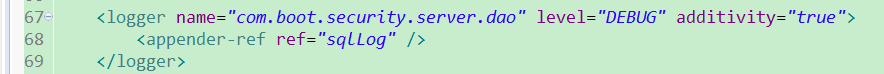
1. 日志配置



日志文件配置位置、root日志级别，自定义业务日志级别，生成日志文件位置，日志文件分包大小

对应logback-admin.xml中的





1. 端口号配置



1. Redis配置



1. 异步、线程池配置



1. 邮件配置

不发邮件的话，请忽略即可，如要发邮件，

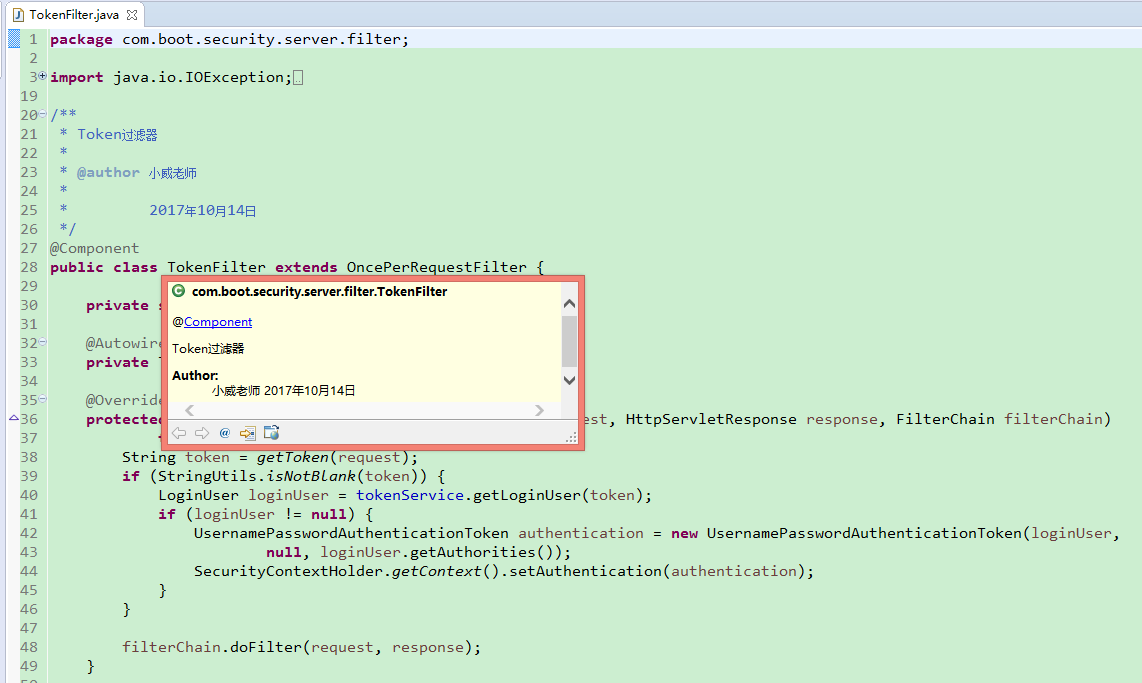
如要使用邮件模块，请写上正确的username和password，并且将最后两行的注释配置打开，否则发邮件可能会失败，如下图



1. **接口开发访问说明**

如可以<http://localhost:8080/users/current?token=049ba87d-6d> 这样以参数的形式带上token，也可以把token放入header中。

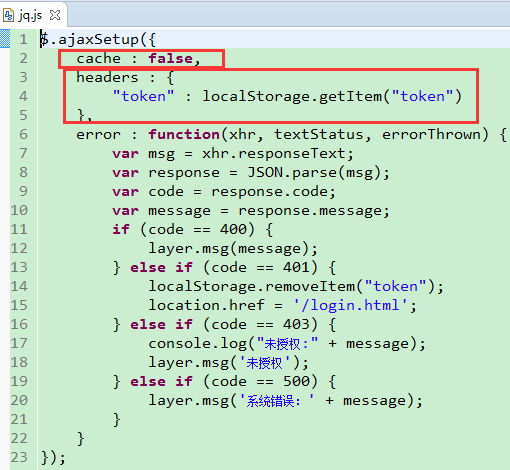
[com](eclipse-javadoc:%E2%98%82=boot-security/src%5C/main%5C/java%3Ccom).[boot](eclipse-javadoc:%E2%98%82=boot-security/src%5C/main%5C/java%3Ccom.boot).[security](eclipse-javadoc:%E2%98%82=boot-security/src%5C/main%5C/java%3Ccom.boot.security).[server](eclipse-javadoc:%E2%98%82=boot-security/src%5C/main%5C/java%3Ccom.boot.security.server).[filter](eclipse-javadoc:%E2%98%82=boot-security/src%5C/main%5C/java%3Ccom.boot.security.server.filter).TokenFilter



从请求参数或者header中解析token，

在页面开发中，只需要引入jq.js即可，具体根据你的页面文件位置进行引入

如<script type="text/javascript" src="../../js/jq.js"></script>



该jq.js文件，对ajax进行了全局处理，禁用缓存，将token放入了header

1. **代码生成介绍**
2. 输入表名

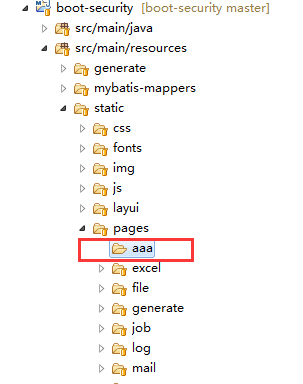


1. 预览



保存，将根路径下java文件copy到项目里，

页面文件copy到static下的pages里，单独建一个文件夹放入页面文件如aaa



Mapper.xml放入mybatis-mappers文件夹下即可，重启项目，配置菜单和权限即可。

1. **定时任务管理介绍**
2. 添加任务



仅支持无参数方法，选定springBeanName会自动联动方法名。

Job名请使用唯一值，cron表达式自己输入，可点击检查来校验。