# 技术框架

## 后端

Jdk 1.8

核心框架：Spring boot 1.5.2

安全框架：Shiro

持久层框架：Mybatis

数据库连接池：Alibaba Druid

缓存框架：Ehcache

日志框架：SL4J

推送：spring-websocket

前后端交互：json

## 前端

Jquery

Html

样式、弹窗：Layui

样式：Bootstrap

树形：Ztree

分页：Datatables

# 项目简介

传统的java web项目都是以war包运行，需要安装tomcat等运行容器，随着spring boot的出现，完全打破了这一现状，此admin项目就是为此而研发的，前后端代码完全分离，基于ajax进行交互。接口完全采用Restful的风格,可以作为开发项目的脚手架，做为基础项目。

该项目采用jdk1.8 + spring boot1.5.2 + shiro + mysql + layui + json + datatables

想学习spring boot的同学也可以从此项目入手，该代码杜绝过度的封装，方便学习或者二次开发，你可以修改订制任何功能模块。

目前包含，用户管理，在线用户，修改密码，菜单管理，角色管理分配，文件管理，公告管理，代码生成，数据源监控，接口swagger文档，头像修改，个人信息修改。

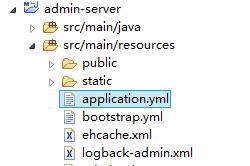
后端有logback日志分包，权限控制提供了基于session和Restful的无状态token两种方式，在线用户使用了spring-websocket推送，全局统一异常处理。

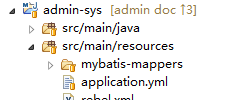
mybatis未进行二次封装，原滋原味，简单sql采用注解，复杂sql采用Mapper.xml配置。

使用了layui的弹出层、菜单、文件上传、富文本编辑、日历、选项卡等。

# 配置文件说明

## 一览配置文件





## 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。

## 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。

## 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

## ehcache.xml

shiro缓存配置，登录token缓存配置。

## 跨域配置



## 全局统一的异常处理

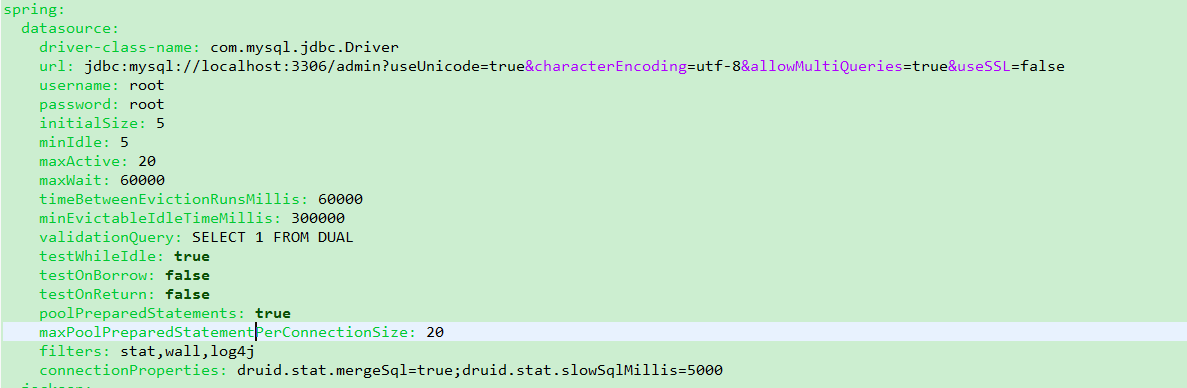
[com](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom).[zw](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw).[admin](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin).[server](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server).[advice](eclipse-javadoc:%E2%98%82=admin-ser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



# 详细配置

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



## 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.zw.admin.service.sys.model, com.zw.admin.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。

## 上传文件



此处配置文件的根目录



此处配置springmvc上传文件大小限制

## 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





此处sql日志只配置了sys项目的dao层，如有别的dao层需要再加配置。

## 端口号配置



## Redis配置



如不使用redis这里可以不用修改

## 异步、线程池配置



# Token介绍

## 登录接口



返回Token对象的json数据

## Token对象

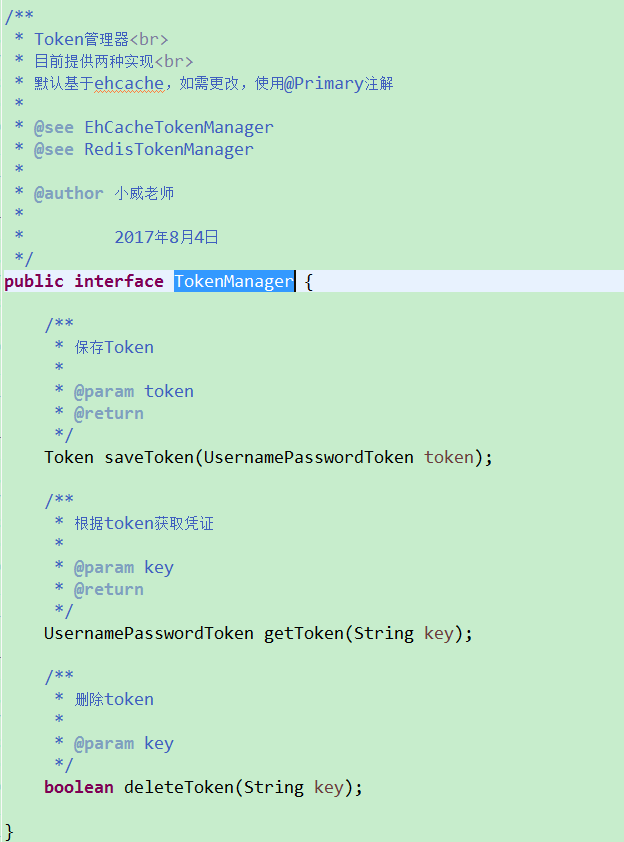
基于网页的session登录和访问不在介绍，以下是Restful token的方式



其中token就是以后用作权限校验的登录凭证，可以加参数的形式或者加header的形式，参数名为login-token，如登陆后访问某一用户信息，

http://localhost:8080/users/1?login-token=xxxxxxxx

## token管理



目前实现类为[com](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom).[zw](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw).[admin](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin).[server](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server).[token](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server.token).EhCacheTokenManager 基于Ehcache实现，[com](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom).[zw](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw).[admin](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin).[server](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server).[token](eclipse-javadoc:%E2%98%82=admin-server/src%5C/main%5C/java%3Ccom.zw.admin.server.token).RedisTokenManager基于redis实现，默认是使用Ehcache，如要使用redis，

1. 请先在application.yml配置redis的host和端口号，并启动redis。
2. 将EhCacheTokenManager类上的注解 @Primary移到RedisTokenManager上即可。