# 说明

**本工程仅供学习使用，请勿用于商用。**

本工程分为3个项目：

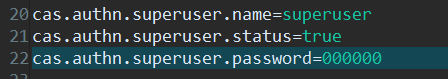
1. cas-------------------------单点登录
2. security------------------主要项目，用来管理各个子项目，子项目都基于此项目运行
3. security-sdk-------------封装了shiro代码，可打包成jar包，用于子项目快速接入security项目，快速搭建用户认证模块和权限模块

# 项目描述

## 1．cas

cas4.2.7版本，此项目部署完成后一般情况下是不需要再进行维护了，对源代码稍微做了一些改动，改动内容如下：

1. 原本用户信息是放在配置文件里，改为从数据库读取，密码进行了MD5盐值加密验证。
2. 增加了一个超级用户功能，超级用户是不需要分配权限信息，默认是拥有所有权限，可以在系统刚刚部署数据库还没有任何数据，和应急时使用，超级用户配置方法如下：
   1. WEB-INF/ cas.properties

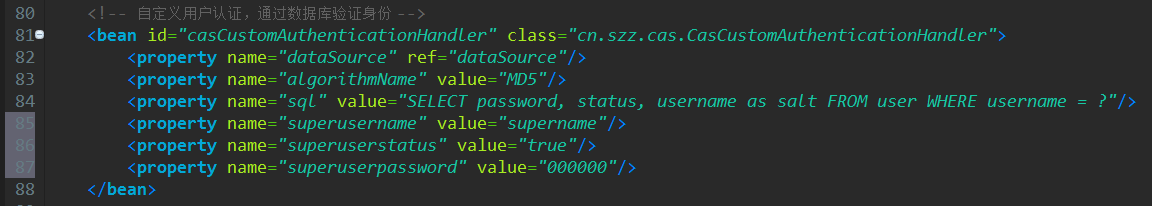


cas.authn.superuser.name 用户名

cas.authn.superuser.status 状态:true 有效 false 无效

cas.authn.superuser.status 密码，采用明文密码

* 1. WEB-INF/ deployerConfigContext.xml



superusername 用户名

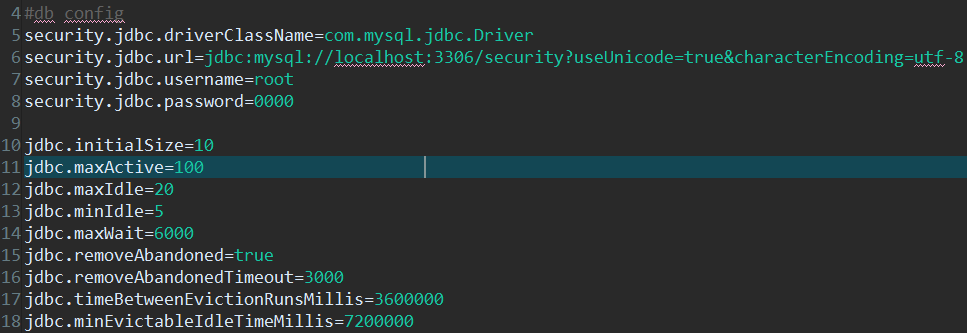
superuserstatus 状态:true 有效 false 无效

superuserpassword 密码

以上两种方法优先选择第二种。如果数据库存在用户名与超级用户名一样的，会优先验证超级用户信息，如果验证不通过，才会进行数据库用户信息验证，建议普通用户不要与超级用户存在一样的用户名。

cas在部署的时候需要指定部署地址，请将WEB-INF/ cas.properties下的server.name修改为你的项目地址。

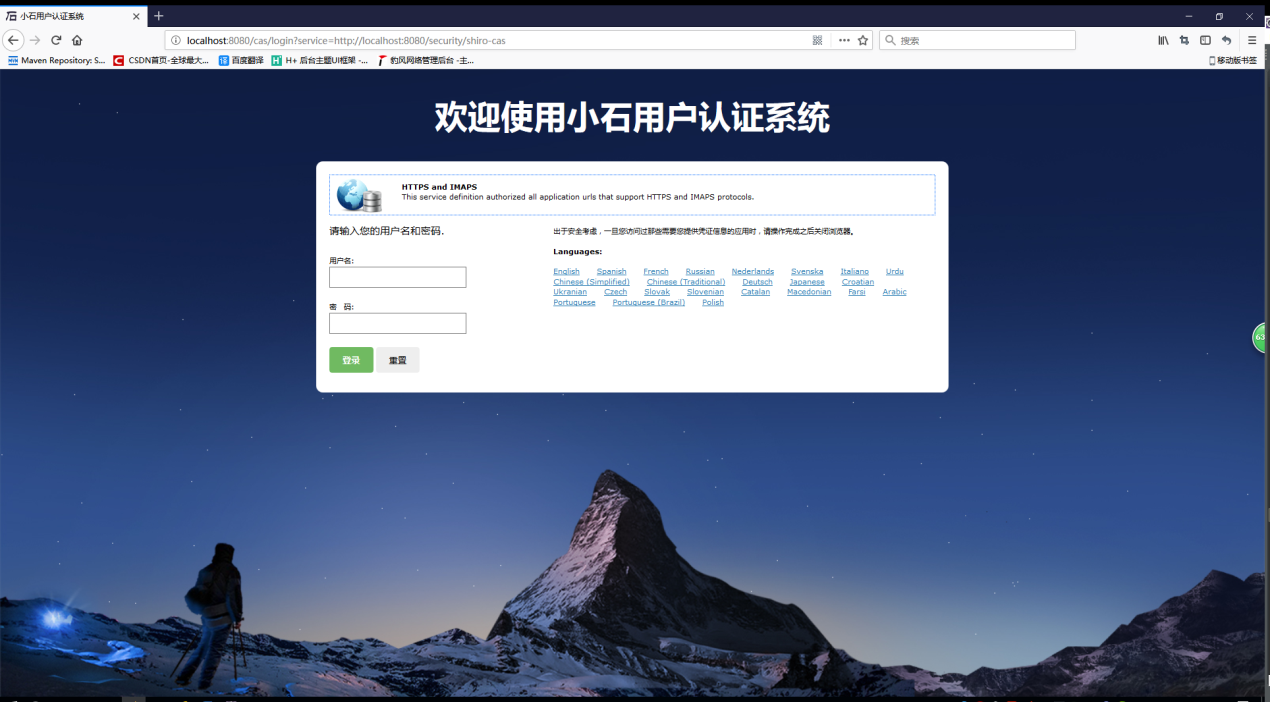
数据库信息配置WEB-INF/ cas.properties



系统默认语言配置WEB-INF/ cas.properties



主要页面：



## 2．security

开发工具：

1、JDK8

2、ecplise4.5.2

3、tomcat8.0

4、mysql5.5

5、maven3.3.9

后端框架：

1、spring4.3.14

2、springMVC4.3.14

3、mybatis1.3.1

4、shiro1.2.6

5、hessian4.0.33

6、ehcache2.6.11

前端框架：

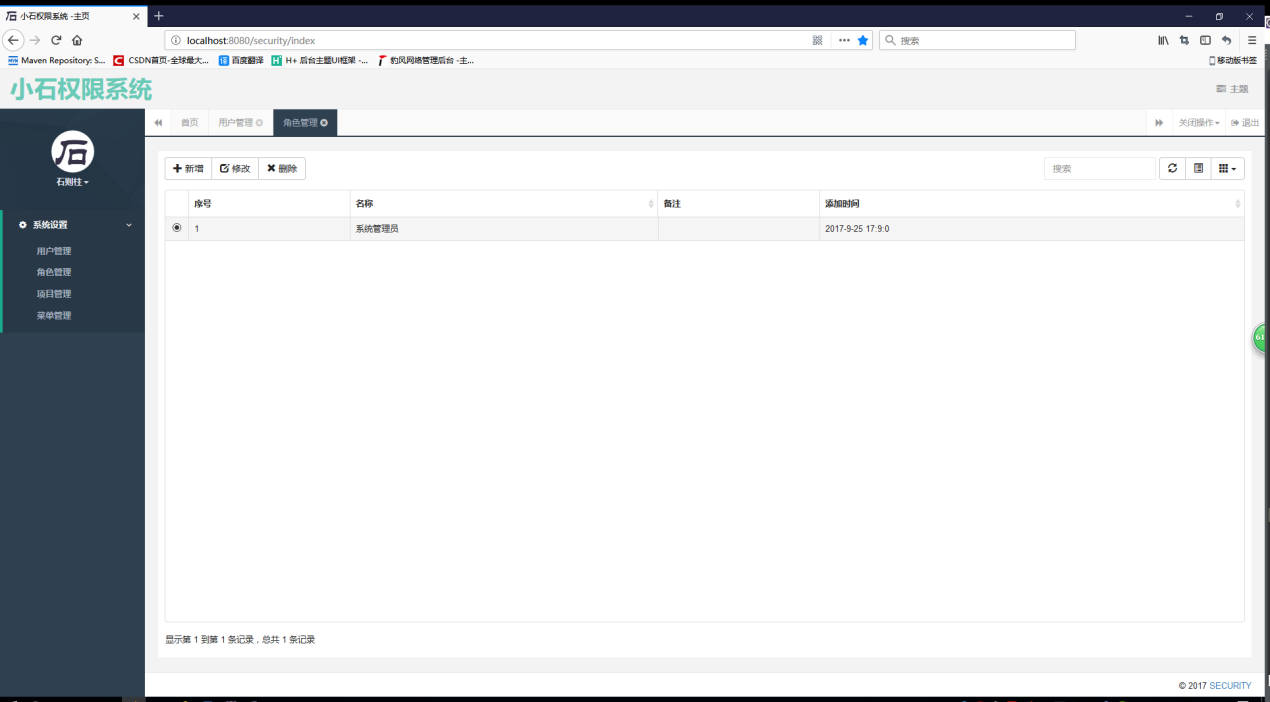
1、H+

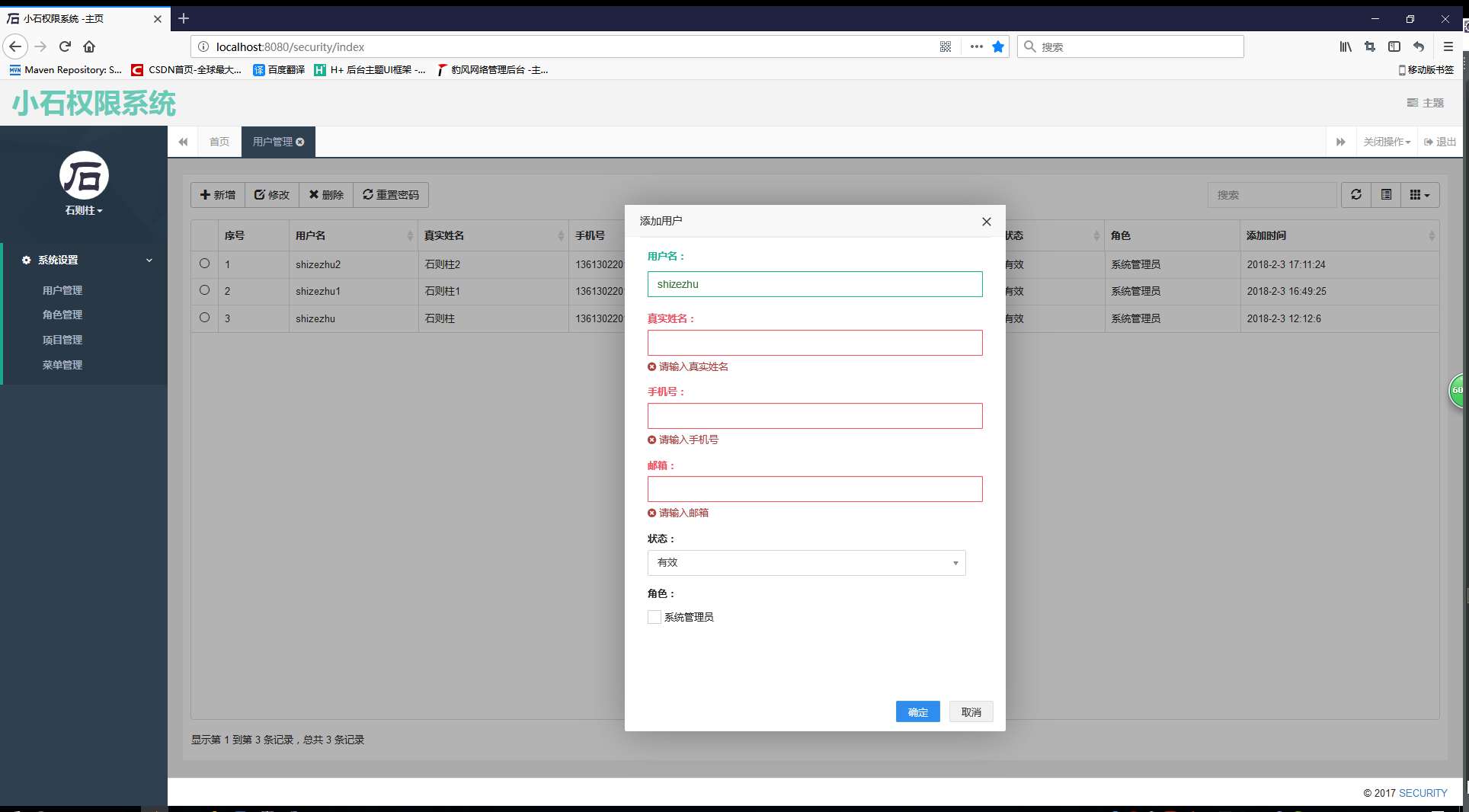
主要模块：

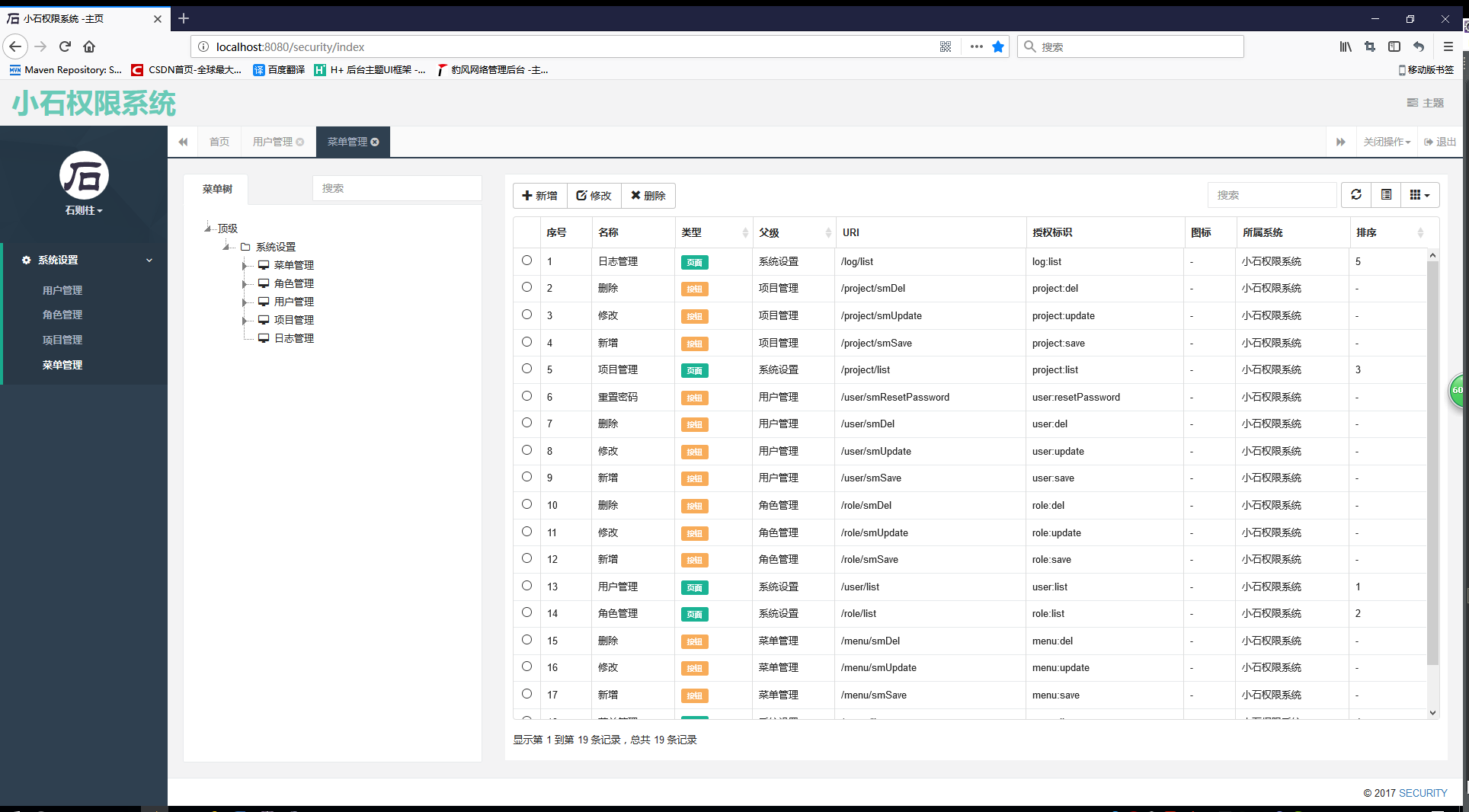
1. 项目管理
   1. 新增
   2. 修改
   3. 删除
2. 角色管理
   1. 新增
   2. 修改
   3. 删除
3. 用户管理
   1. 新增
   2. 修改
   3. 删除
   4. 重置密码
4. 菜单管理
   1. 新增
   2. 修改
   3. 删除
5. 日志管理

主要页面：

1. 主页面







## 3．security-sdk

开发工具：

1、JDK7

2、ecplise4.5.2

3、maven3.3.9

后端框架：

1、spring4.3.14

4、shiro1.2.6

5、hessian4.0.33

6、ehcache2.6.11

可打包成jar文件导入到其它系统中使用，主要提供以下服务：

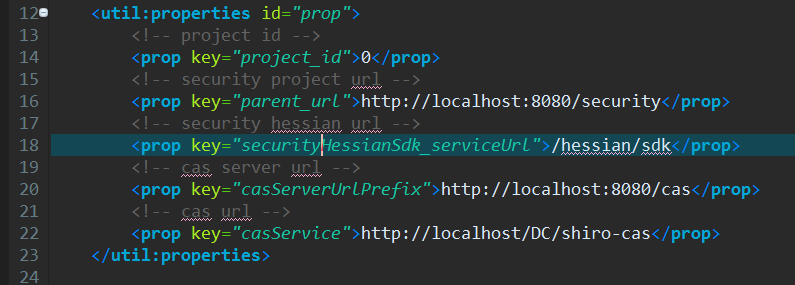
1. 用户信息认证
2. 用户权限认证
3. 远程获取用户信息
4. 远程获取权限信息
5. 操作日志保存接口
6. 远程刷新项目权限信息
7. 远程刷新用户授权信息

使用方法：

1. 在security中的项目管理新增一个项目并得到项目ID
2. 将security-sdk.jar和所需的依赖包导入新项目
3. 将配置文件
   1. ehcache.xml---------------------------------------缓存配置
   2. security-sdk-remote.xml-----------------------远程接口配置
   3. security-sdk.xml----------------------------------主要配置

拷贝到新项目

1. 修改security-sdk.xml



project\_id：项目ID

parent\_url：security项目地址

securityHessianSdk\_serviceUrl：security远程接口地址

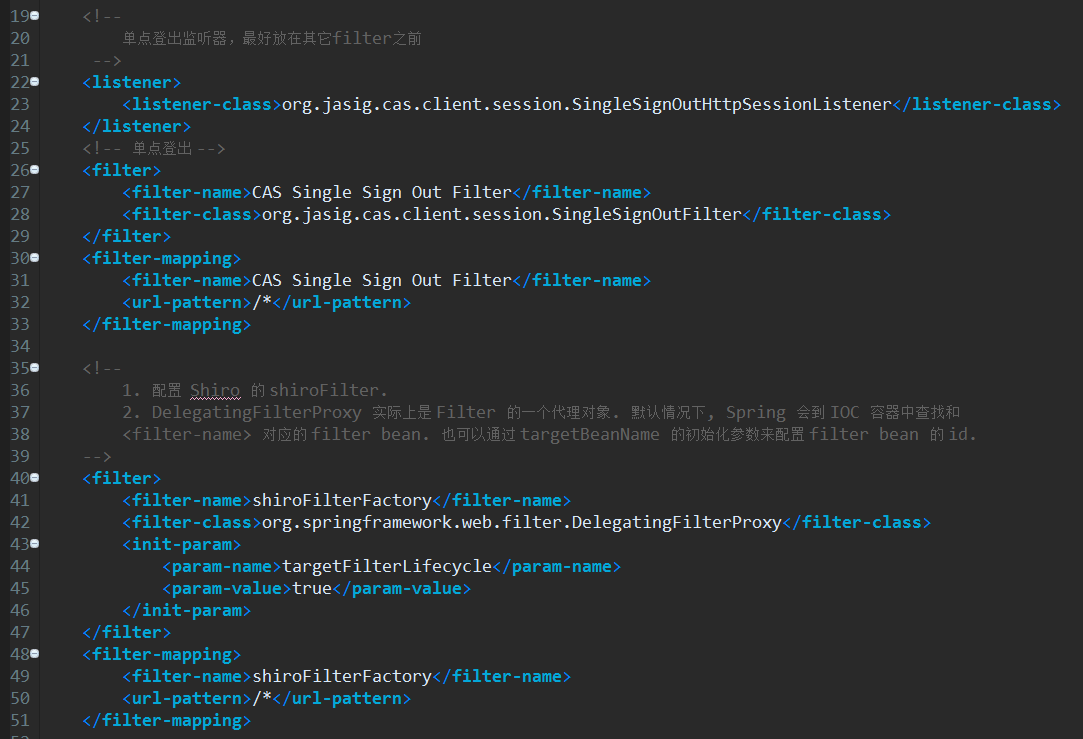
casServerUrlPrefix：cas项目地址

casService：本项目cas服务地址，默认为http://ip:port/项目/shiro-cas

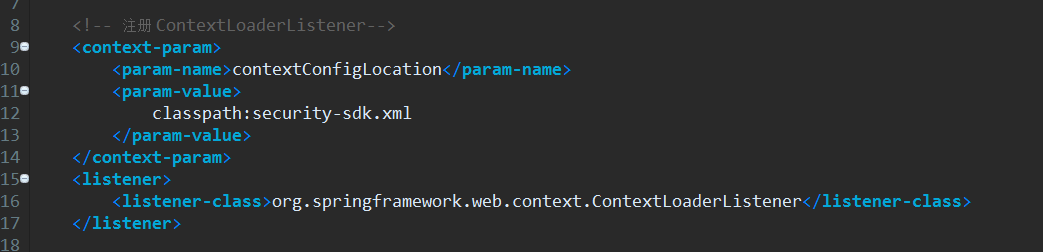


此处为静态资源权限配置，按需修改

1. 修改web.xml



最好是放在其它filter之前。





完成上面的步骤后就已经成功接入到security系统。然后将子系统的菜单通过security的菜单管理录入到系统，就可以通过security访问各个子系统了。

sdk包提供一个服务类SecuritySdkService，提供api如下:

1、SecuritySdkUser getCurrentUser();//获取当前用户2、SecuritySdkUser getUserById(long userId);//根据ID获取用户3、List<SecuritySdkUser> listUserAll();//获取所有用户4、void saveLog(String userIp, String name, String descr, LogType type, String content);//保存日志5、void saveLog(String name, String descr, LogType type, String content);//保存日志6、void saveLog(String userIp, String name, String descr, LogType type);//保存日志7、void saveLog(String name, String descr, LogType type);//保存日志8、SecurityHessianClient getSecurityHessianClient();//获取security项目远程服务9、long getProjectId();//获取本项目ID

在进行权限和用户认证的时候会有未认证通过的时候，默认返回的状态信息如下：

1. 用户未登陆

{status : ” NOLOGIN”, message : “no logon”}

1. 用户无权限

{status : ” NOAUTH”, message : “no auth”}

可以通过security-sdk.xml进行修改



status\_key\_name：状态码key值

message\_key\_name：状态消息key值

no\_login\_status：用户未登录状态码

no\_login\_message：用户未登录消息

no\_auth\_status：用户无权限状态码

no\_auth\_message：用户无权限消息