**品优购电商系统开发**

**第5章**

**安全框架与商家入驻审核**

传智播客.黑马程序员

# 课程目标

目标1：完成运营商系统登录与安全控制功能

目标2：完成商家申请入驻的功能

目标3：完成商家审核的功能

目标4：完成商家系统登陆与安全控制功能

# 1.Shrio回顾

## 1.1步骤一

依赖配置：

|  |
| --- |
| *<!-- 配置shiro安全框架 -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**>  <**version**>1.4.0</**version**> </**dependency**> |

## 1.2步骤二

web.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>  ***<!-- 配置Spring委派过滤器代理(要求Spring容器中有了个bean的id为shiroFilter) -->* <filter>  <filter-name>shiroFilter</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>shiroFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>**  </**web-app**> |

## 1.3步骤三

applicationContext-shiro.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>  *<!-- 1. 配置shiro作为Spring委派过滤器(shiro过滤器工厂) -->* <**bean id="shiroFilter"**  **class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 配置登录页面访问URL -->* <**property name="loginUrl" value="/login.html"**/>  *<!-- 配置权限管理URL过滤器链(配置顺序为自上而下) -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /logout = logout  /\*\* = user  </**value**>  </**property**>  </**bean**>  *<!-- 2. 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置认证域(来自ini配置文件)-->* <**property name="realm" ref="iniRealm"**/>  </**bean**>  *<!-- 3. 配置IniRealm认证域 -->* <**bean id="iniRealm" class="org.apache.shiro.realm.text.IniRealm"**>  *<!-- 设置ini配置文件 -->* <**constructor-arg name="resourcePath" value="classpath:shiro.ini"**/>  </**bean**> </**beans**> |

配置说明：

<property name="**filterChainDefinitions**"></property>

配置权限管理URL过滤器链(配置顺序为自上而下)

1. **user**: 过滤器只要用户登录

(isRemembered()==true or isAuthenticated()==true)即可访问成功

1. **authc**: 过滤器只要用户登录isAuthenticated()==true即可访问成功
2. **anon** : 匿名访问，不需要登录
3. **logout** : 安全退出

# 2.运营商系统登录与安全控制

## 2.1需求分析

完成运营商登录功能：http://manager.pinyougou.com/login.html



## 2.2登录功能实现

### 2.2.1配置文件

第一步：修改pinyougou-manager-web的pom.xml，添加依赖

|  |
| --- |
| <**dependencies**>  *<!-- 配置shiro安全框架 -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**>  </**dependency**> </**dependencies**> |

第二步：修改pinyougou-manager-web/src/main/webapp/WEB-INF/web.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>   ***<!-- 配置Spring委派过滤器代理(要求Spring容器中有了个bean的id为shiroFilter) -->* <filter>  <filter-name>shiroFilter</filter-name>  <filter-class>org.springframework.web.filter.DelegatingFilterProxy</filter-class>  </filter>  <filter-mapping>  <filter-name>shiroFilter</filter-name>  <url-pattern>/\*</url-pattern>  </filter-mapping>**   *<!-- 配置Spring MVC前端控制器(核心控制器) -->* <**servlet**>  <**servlet-name**>pinyougou</**servlet-name**>  <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>   **<param-value>classpath:springmvc.xml,  classpath:applicationContext-shiro.xml</param-value>**  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>   *<!-- 配置全局编码过滤器 -->* <**filter**>  <**filter-name**>characterEncodingFilter</**filter-name**> <**filter-class**>org.springframework.web.filter.CharacterEncodingFilter</**filter-class**>  <**init-param**>  <**param-name**>encoding</**param-name**>  <**param-value**>UTF-8</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>characterEncodingFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>  *<!-- 欢迎列表 -->* <**welcome-file-list**>  <**welcome-file**>index.html</**welcome-file**>  </**welcome-file-list**> </**web-app**> |

第三步：在pinyougou-manager-web/src/main/resources/目录下创建

applicationContext-shiro.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 配置shiro作为Spring委派过滤器(shiro过滤器工厂) -->* <**bean id="shiroFilter"**  **class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置注入安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!--  配置登录页面访问URL  没有登录的用户请求需要登录的页面时自动跳转到登录页面，不是必须的属性，  不输入地址的话会自动寻找项目web项目的根目录下的”/login.html页面  -->* <**property name="loginUrl" value="/login.html"**/>  *<!--  配置过滤器链(配置顺序为自上而下)  1. user: 过滤器只要用户登录(isRemembered()==true or isAuthenticated()==true)即可访问成功  2. authc: 过滤器只要用户登录isAuthenticated()==true即可访问成功  3. anon : 匿名访问，不需要登录  4. logout : 安全退出  -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /img/\*\* = anon  /js/\*\* = anon  /plugins/\*\* = anon  /user/login = anon  /logout = logout  /\*\* = user  </**value**>  </**property**>  </**bean**>   *<!-- 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置认证域(来自ini配置文件)-->* <**property name="realm" ref="iniRealm"**/>  </**bean**>   *<!-- 配置IniRealm认证域 -->* <**bean id="iniRealm" class="org.apache.shiro.realm.text.IniRealm"**>  *<!-- 设置ini配置文件 -->* <**constructor-arg name="resourcePath" value="classpath:shiro.ini"**/>  </**bean**> </**beans**> |

第四步：pinyougou-manager-web/src/main/resources/目录下创建shrio.ini

|  |
| --- |
| # 配置运营商用后台管理用户 [**users**] admin=123456,admin root=123456,admin test=123456,admin  # 配置所有角色与权限 [**roles**] admin=\*:\* |

### 2.2.2登录页面

修改pinyougou-manager-web/src/main/webapp/login.html页面

|  |
| --- |
| <**form action="/user/login" method="post" class="sui-form"**>  <**div class="input-prepend"**>  <**span class="add-on loginname"**></**span**>  <**input name="username" type="text"  placeholder="邮箱/用户名/手机号" class="span2 input-xfat"**/>  </**div**>  <**div class="input-prepend"**>  <**span class="add-on loginpwd"**></**span**>  <**input name="password" type="password"  placeholder="请输入密码" class="span2 input-xfat"**/>  </**div**>  <**div class="setting"**>  <**div id="slider"**>  <**div id="slider\_bg"**></**div**>  <**span id="label"**>>></**span**>  <**span id="labelTip"**>拖动滑块验证</**span**>  </**div**>  </**div**>  <**div class="logined"**>  <**input type="submit"   class="sui-btn btn-block btn-xlarge btn-danger"   value="登&nbsp;&nbsp;录"**/>  </**div**> </**form**> |

### 2.2.3后端代码

pinyougou-manager-web\src\main\java\com\pinyougou\manager\controller\LoginController.java提供用户登录方法

|  |
| --- |
| **package** com.pinyougou.manager.controller;  **import** org.apache.shiro.SecurityUtils; **import** org.apache.shiro.authc.UsernamePasswordToken; **import** org.apache.shiro.subject.Subject; **import** org.springframework.stereotype.Controller; **import** org.springframework.web.bind.annotation.PostMapping; **import** org.springframework.web.bind.annotation.RequestMapping; */\*\* 登录控制器 \*/* **@Controller @RequestMapping("/user")** **public class** LoginController {   */\*\* 用户登录 \*/* @PostMapping(**"/login"**)  **public** String login(String username, String password){  **try**{  */\*\*########### shiro框架身份认证 ##########\*/  /\*\* 获取认证的主体 \*/* Subject subject = SecurityUtils.*getSubject*();  */\*\* 创建用户名与密码令牌 \*/* UsernamePasswordToken token = **new** UsernamePasswordToken(username, password);  */\*\* 身份认证 \*/* subject.login(token);  */\*\* 判断用户是否通过身份认证 \*/* **if** (subject.isAuthenticated()){  **return "redirect:/admin/index.html"**;  }  }**catch**(Exception ex){  ex.printStackTrace();  }  **return "redirect:/login.html"**;  } } |

## 2.3获取登录用户名

### 2.3.1后端代码

pinyougou-manager-web\src\main\java\com\pinyougou\manager\controller\LoginController.java提供获取登录用户名方法

|  |
| --- |
| */\*\* 获取登录用户名 \*/* @GetMapping(**"/findLoginName"**) **@ResponseBody** **public** Map<String, String> findLoginName(){  *// 获取登录用户名* String loginName = SecurityUtils.*getSubject*().getPrincipal().toString();  Map<String,String> data = **new** HashMap<>();  data.put(**"loginName"**, loginName);  **return** data; } |

### 2.3.2前端代码

在pinyougou-manager-web\src\main\webapp\js\controller目录下创建

indexController.js：

|  |
| --- |
| *// 窗口加载完* ***window***.onload = **function** () {  **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **loginName** : **''** *// 登录用户名* },  **methods** : { *// 操作方法* findLoginName : **function** () { *// 获取登录用户名  // 发送异步请示* **axios**.get(**"/user/findLoginName"**)  .then(**function**(response){  *// 获取响应数据* vue.**loginName** = response.**data**.**loginName**;  });  }  },  created : **function** () { *// 创建生命周期  // 调用获取登录用户名方法* **this**.findLoginName();  }  }); }; |

pinyougou-manager-web/src/main/webapp/admin/index.html引入js

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/indexController.js"**></**script**> |

将页面上的 测试用户 替换成 {{loginName}}

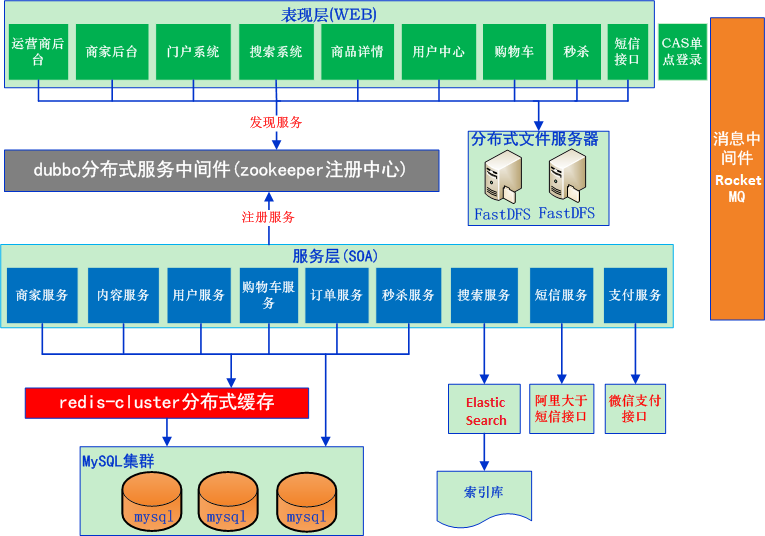
## 2.4退出登录



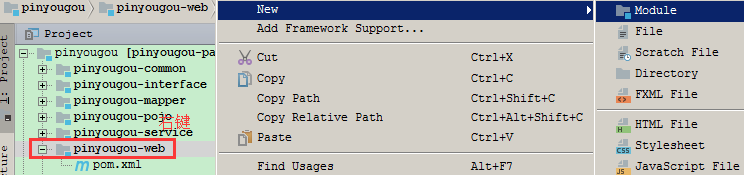
pinyougou-manager-web/src/main/webapp/admin/index.html页面，修改注销的链接：

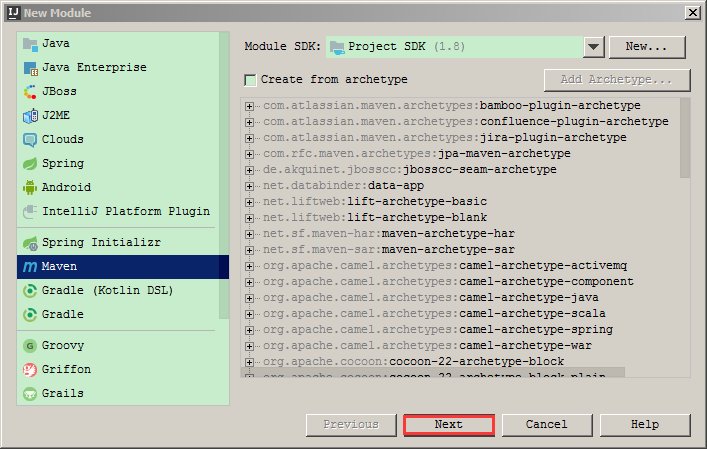
|  |
| --- |
| <**div class="pull-right"**>  <**a href="/logout" class="btn btn-default btn-flat"**>注销</**a**> </**div**> |

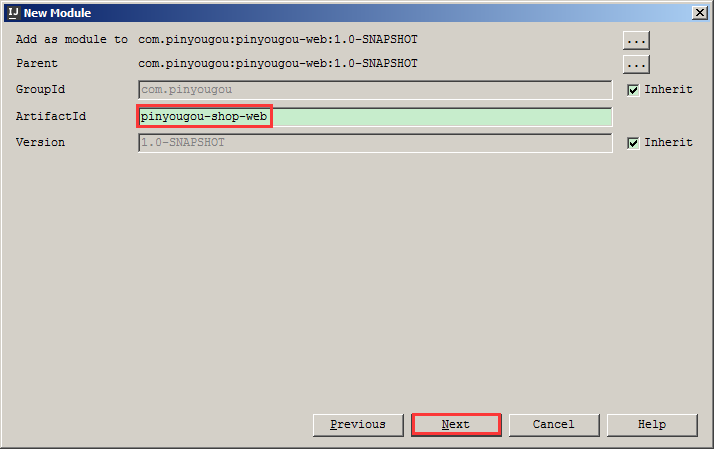
# 3.搭建商家管理后台

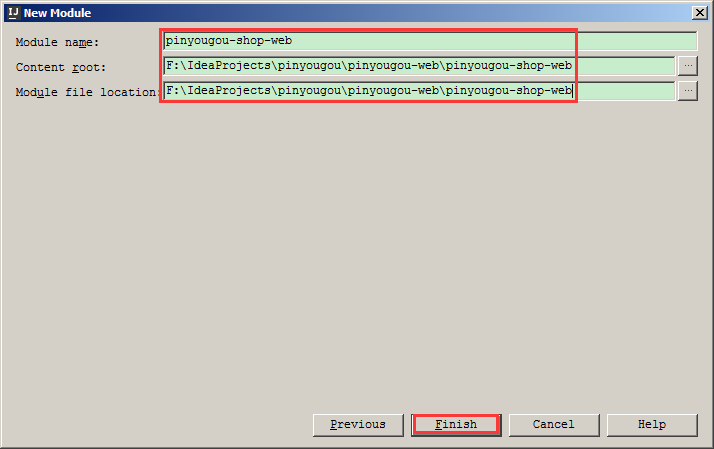


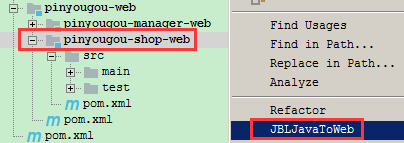
## 3.1 pinyougou-shop-web(war)











与pinyougou-manager-web的构建方式一样，拷贝pinyougou-manager-web模块中的web.xml(去掉shiro配置)、springmvc.xml、log4j.properties

pinyougou-shop-web/pom.xml：

|  |
| --- |
| <**properties**>  *<!-- 覆盖父工程中的port属性 -->* <**port**>**9102**</**port**> </**properties**> |

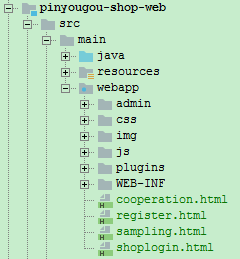
pinyougou-shop-web/src/main/resources/springmvc.xml：

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns:mvc="http://www.springframework.org/schema/mvc"  xmlns:dubbo="http://code.alibabatech.com/schema/dubbo"  xmlns:context="http://www.springframework.org/schema/context"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd  http://www.springframework.org/schema/mvc  http://www.springframework.org/schema/mvc/spring-mvc.xsd  http://code.alibabatech.com/schema/dubbo  http://code.alibabatech.com/schema/dubbo/dubbo.xsd  http://www.springframework.org/schema/context  http://www.springframework.org/schema/context/spring-context.xsd"**>  *<!-- 配置开启组件扫描 -->* **<context:component-scan base-package="com.pinyougou.shop.controller"/>**  *<!-- 开启MVC注解驱动 -->* <**mvc:annotation-driven**>  *<!-- 配置消息转换 -->* <**mvc:message-converters**>  *<!-- 配置用fastjson作为json处理框架 -->* <**bean class="com.alibaba.fastjson.support.spring.FastJsonHttpMessageConverter"**>  *<!-- 配置支持的类型 -->* <**property name="supportedMediaTypes"  value="application/json;charset=UTF-8"**/>  </**bean**>  </**mvc:message-converters**>  </**mvc:annotation-driven**>  *<!-- 配置静态资源用WEB容器默认的servlet来处理 -->* <**mvc:default-servlet-handler**/>  *<!-- ############## 配置dubbo服务消费者 ############## -->  <!-- 配置当前应用的名称 -->* <**dubbo:application name="pinyougou-shop-web"**/>  *<!-- 配置zookeeper作为注册中心，注册服务地址 -->* <**dubbo:registry protocol="zookeeper" address="192.168.12.131:2181"**/>  *<!-- 配置采用包扫描来引用服务，产生服务接口的代理对象 -->* <**dubbo:annotation package="com.pinyougou.shop.controller"**/> </**beans**> |

**创建包名：com.pinyougou.shop.controller**

## 3.2准备工作

将“资料\商家管理后台静态资源”下的所有文件拷贝到pinyougou-shop-web/src/main/webapp/目录下



配置域名访问：shop.pinyougou.com

第一步：修改hosts文件，增加域名

127.0.0.1 shop.pinyougou.com

第二步：修改nginx.conf增加反向代理

|  |
| --- |
| server {  listen 80;  server\_name shop.pinyougou.com;  proxy\_set\_header X-Forwarded-Host $host;  proxy\_set\_header X-Forwarded-Server $host;  location / {  proxy\_pass http://127.0.0.1:9102;  proxy\_connect\_timeout 600;  proxy\_read\_timeout 600;  }  } |

第三步：重启Nginx

nginx -s reload

第四步：访问地址

**http://shop.pinyougou.com/admin/index.html**

# 4.商家申请入驻

## 4.1需求分析

商家申请入驻，需要填写商家相关的信息。待运营商平台审核通过后即可使用。

**访问地址：http://shop.pinyougou.com/register.html**

## 4.3前端代码

修改pinyougou-shop-web/src/main/webapp/register.html页面，引入js

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> |

register.html绑定表单（部分代码）31行-171行：

|  |
| --- |
| <**form class="sui-form form-horizontal"**>  <**div class="control-group"**>  <**label class="control-label"**>登录名（不可修改）：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="登录名"  v-model="entity.sellerId"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>登录密码：</**label**>  <**div class="controls"**>  <**input type="password" placeholder="登录密码"  v-model="entity.password"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>店铺名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="店铺名称"  v-model="entity.nickName"  class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司名称"  v-model="entity.name" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司电话：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司电话"  v-model="entity.telephone" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>公司详细地址：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="公司详细地址"  v-model="entity.addressDetail" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人姓名：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人姓名"  v-model="entity.linkmanName" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人QQ：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人QQ"  v-model="entity.linkmanQq" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人手机：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人手机"  v-model="entity.linkmanMobile" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>联系人EMAIL：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="联系人EMAIL"  v-model="entity.linkmanEmail" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>营业执照号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="营业执照号"  v-model="entity.licenseNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>税务登记证号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="税务登记证号"  v-model="entity.taxNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>组织机构代码证：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="组织机构代码证"  v-model="entity.orgNumber" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>法定代表人：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="法定代表人"  v-model="entity.legalPerson" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>法定代表人身份证号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="法定代表人身份证号"  v-model="entity.legalPersonCardId" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>开户行名称：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="开户行名称"  v-model="entity.bankName" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>开户行支行：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="开户行支行"  v-model="entity.bankUser" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>银行账号：</**label**>  <**div class="controls"**>  <**input type="text" placeholder="银行账号"  v-model="entity.brief" class="input-xfat input-xlarge"**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**>  **&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**</**label**>  <**div class="controls"**>  <**input name="m1" type="checkbox" value="2" checked=""**>  <**span**>同意协议并注册 <**a href="sampling.html"**>《品优购商家入驻协议》</**a**></**span**>  </**div**>  </**div**>  <**div class="control-group"**>  <**label class="control-label"**></**label**>  <**div class="controls btn-reg"**>  <**a class="sui-btn btn-block btn-xlarge btn-danger"  @click="saveOrUpdate()"**>申请入驻</**a**>  </**div**>  </**div**> </**form**> |

pinyougou-shop-web\src\main\webapp\register.html页面提供js代码

|  |
| --- |
| <**script type="text/javascript"**>  *// 窗口加载完* ***window***.onload = **function** () {  *// 创建Vue对象* **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **entity** : {} *// 数据封装对象(表单)* },  **methods** : { *// 定义操作方法* saveOrUpdate: **function** () { *// 添加或修改  // 发送异步请求* **axios**.post(**"/seller/save"**, **this**.**entity**)  .then(**function** (response) {  *// 获取响应数据* **if** (response.**data**) { *// 操作成功  /\*\* 跳转到商家登录页面 \*/* ***location***.href = **"/shoplogin.html"**;  } **else** {  *alert*(**'操作失败！'**);  }  });  }  }  });  }; </**script**> |

## 4.4后端代码

### 4.4.1控制器层

pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller包下创建SellerController.java

|  |
| --- |
| @RestController @RequestMapping(**"/seller"**) **public class** SellerController {  */\*\* 注入商家服务接口代理对象 \*/* @Reference(timeout = 10000)  **private** SellerService **sellerService**;  */\*\* 添加商家 \*/* @PostMapping(**"/save"**)  **public boolean** save(@RequestBody Seller seller){  **try**{  **sellerService**.save(seller);  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**;  } } |

### 4.4.2服务层

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl包下创建SellerServiceImpl.java

|  |
| --- |
| @Service  @Transactional **public class** SellerServiceImpl **implements** SellerService {  @Autowired  **private** SellerMapper **sellerMapper**;  */\*\* 添加商家 \*/* @Override  **public void** save(Seller seller) {  **try**{  seller.setStatus(**"0"**);  seller.setCreateTime(**new** Date());  **sellerMapper**.insertSelective(seller);  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  }  }  @Override  **public void** update(Seller seller) {  }  @Override  **public void** delete(Serializable id) {  }  @Override  **public void** deleteAll(Serializable[] ids) {  }  @Override  **public** Seller findOne(Serializable id) {  **return null**;  }  @Override  **public** List<Seller> findAll() { **return null**;}  @Override  **public** List<Seller> findByPage(Seller seller, **int** page, **int** rows) {  **return null**;  } } |

# 5.商家审核【运营商后台】

## 5.1需求分析

商家申请入驻后，需要网站运营人员在运营商后台pinyougou-manager-web中进行审核，审核后商家才可以登录系统。

**状态值： 0：未审核 1：已审核 2：审核未通过 3：关闭**

## 5.2商家待审核列表

### 5.2.1前端代码

修改pinyougou-manager-web/src/main/webapp/admin/seller\_check.html,引入js：

|  |
| --- |
| <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/pagination.js"**></**script**> <**script src="/js/controller/sellerController.js"**></**script**> |

seller\_check.html加入分页组件，</table>标签后面：

|  |
| --- |
| <**pagination :pages="pages" :current="page" @navpage="search"**/> |

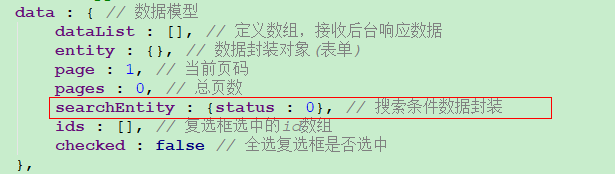
v-for循环迭代：

|  |
| --- |
| <**tr v-for="entity in dataList"**>  <**td**><**input type="checkbox"**></**td**>  <**td**>{{**entity**.sellerId}}</**td**>  <**td**>{{**entity**.**name**}}</**td**>  <**td**>{{**entity**.nickName}}</**td**>  <**td**>{{**entity**.linkmanName}}</**td**>  <**td**>{{**entity**.telephone}}</**td**>  <**td class="text-center"**>  <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#sellerModal"**>详情  </**button**>  </**td**> </**tr**> |

seller\_check.htm查询条件、查询按钮绑定点击事件：

|  |
| --- |
| <**div class="has-feedback"**>  公司名称：<**input v-model="searchEntity.name"**/>  店铺名称： <**input v-model="searchEntity.nickName"**/>  <**button class="btn btn-default" @click="search(1)"**>查询</**button**> </**div**> |

pinyougou-manager-web\src\main\webapp\js\controller\sellerController.js设置默认的查询条件：



### 5.2.2后台代码

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller包下创建SellerController.java：

|  |
| --- |
| @RestController @RequestMapping(**"/seller"**) **public class** SellerController {  @Reference(timeout = 10000)  **private** SellerService **sellerService**;  */\*\* 多条件分页查询商家 \*/* @GetMapping(**"/findByPage"**)  **public** PageResult findByPage(Seller seller, Integer page,  **@RequestParam(defaultValue = "10")Integer rows**){  **try** {  */\*\* GET请求中文转码 \*/* **if** (seller != **null** && StringUtils.*isNoneBlank*(seller.getName())) {  seller.setName(**new** String(seller.getName()  .getBytes(**"ISO8859-1"**), **"UTF-8"**));  }  **if** (seller != **null** && StringUtils.*isNoneBlank*(seller.getNickName())) {  seller.setNickName(**new** String(seller.getNickName()  .getBytes(**"ISO8859-1"**), **"UTF-8"**));  }  }**catch** (Exception ex){  ex.printStackTrace();  }  **return sellerService**.findByPage(seller, page, rows);  } } |

pinyougou-interface/src/main/java/com.pinyougou.service/SellerService.java：

|  |
| --- |
| */\*\* 多条件分页查询 \*/* PageResult findByPage(Seller seller, **int** page, **int** rows); |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java：

|  |
| --- |
| */\*\* 多条件分页查询商家 \*/* @Override **public** PageResult findByPage(Seller seller, **int** page, **int** rows) {  **try**{  */\*\* 开始分页 \*/* PageInfo<Seller> pageInfo = PageHelper.*startPage*(page, rows)  .doSelectPageInfo(**new** ISelect() {  @Override  **public void** doSelect() {  **sellerMapper**.findAll(seller);  }  });  **return new** PageResult(**pageInfo.getPages()**, pageInfo.getList());  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

pinyougou-mapper/src/main/java/com.pinyougou.mapper/SellerMapper.java：

|  |
| --- |
| */\*\* 多条件查询商家 \*/* List<Seller> findAll(Seller seller); |

pinyougou-mapper/src/main/resources/mappers/SellerMapper.xml：

|  |
| --- |
| *<!-- 多条件查询商家 -->* <**select id="findAll" resultType="Seller"**>  select *\** from tb\_seller  <**where**>  <**if test="status != null and status !=''"**>  status = #{status}  </**if**>  <**if test="name != null and name !=''"**>  and name like CONCAT('%',#{name},'%')  </**if**>  <**if test="nickName != null and nickName !=''"**>  and nick\_name like CONCAT('%',#{nickName},'%')  </**if**>  </**where**> </**select**> |

## 5.3商家详情



pinyougou-manager-web/src/main/webapp/admin/seller\_check.html列表的“详情”按钮，绑定点击事件

|  |
| --- |
| <**button type="button" class="btn bg-olive btn-xs"  data-toggle="modal" data-target="#sellerModal"  @click="show(entity);"**>详情</**button**> |

pinyougou-manager-web/src/main/webapp/admin/seller\_check.html绑定页面弹出窗口 (107行-198行)

|  |
| --- |
| *<!-- 选项卡开始 -->* <**div id="myTabContent" class="tab-content"**>  <**div class="tab-pane active in" id="home"**>  <**br**/>  <**table class="table table-bordered table-striped" width="800px"**>  <**tr**>  <**td**>公司名称</**td**>  <**td**>{{entity.**name**}}</**td**>  </**tr**>  <**tr**>  <**td**>公司手机</**td**>  <**td**>{{entity.mobile}}</**td**>  </**tr**>  <**tr**>  <**td**>公司电话</**td**>  <**td**>{{entity.telephone}}</**td**>  </**tr**>  <**tr**>  <**td**>公司详细地址</**td**>  <**td**>{{entity.addressDetail}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="linkman"**>  <**br**/>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>联系人姓名</**td**>  <**td**>{{entity.linkmanName}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人QQ</**td**>  <**td**>{{entity.linkmanQq}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人手机</**td**>  <**td**>{{entity.linkmanMobile}}</**td**>  </**tr**>  <**tr**>  <**td**>联系人E-Mail</**td**>  <**td**>{{entity.linkmanEmail}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="certificate"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>营业执照号</**td**>  <**td**>{{entity.licenseNumber}}</**td**>  </**tr**>  <**tr**>  <**td**>税务登记证号</**td**>  <**td**>{{entity.taxNumber}}</**td**>  </**tr**>  <**tr**>  <**td**>组织机构代码证号</**td**>  <**td**>{{entity.orgNumber}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="ceo"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>法定代表人</**td**>  <**td**>{{entity.legalPerson}}</**td**>  </**tr**>  <**tr**>  <**td**>法定代表人身份证号</**td**>  <**td**>{{entity.legalPersonCardId}}</**td**>  </**tr**>  </**table**>  </**div**>  <**div class="tab-pane fade" id="bank"**>  <**br**>  <**table class="table table-bordered table-striped"**>  <**tr**>  <**td**>开户行名称</**td**>  <**td**>{{entity.bankName}}</**td**>  </**tr**>  <**tr**>  <**td**>开户行支行</**td**>  <**td**>{{entity.bankUser}}</**td**>  </**tr**>  <**tr**>  <**td**>银行账号</**td**>  <**td**>{{entity.brief}}</**td**>  </**tr**>  </**table**>  </**div**> </**div**> *<!-- 选项卡结束 -->* |

## 5.4商家审核

### 5.4.1前端代码

seller\_check.html页面上的按钮，绑定点击事件：

|  |
| --- |
| <**button class="btn btn-success" data-dismiss="modal"  @click="updateStatus(entity.sellerId,'1')"  aria-hidden="true"**>审核通过</**button**> <**button class="btn btn-danger" data-dismiss="modal"  @click="updateStatus(entity.sellerId,'2')"  aria-hidden="true"**>审核未通过</**button**> <**button class="btn btn-danger" data-dismiss="modal"  @click="updateStatus(entity.sellerId,'3')"  aria-hidden="true"**>关闭商家</**button**> <**button class="btn btn-default" data-dismiss="modal"  aria-hidden="true"**>关闭</**button**> |

修改pinyougou-manager-web/src/main/webapp/js/controller的

sellerController.js：

|  |
| --- |
| updateStatus : **function** (sellerId, status) { *// 商家审核* **axios**.get(**"/seller/updateStatus?sellerId="** + sellerId +**'&status='** + status)  .then(**function**(response){  **if**(response.**data**){  *// 重新加载数据* vue.search(1);  }**else**{  *alert*(**"操作失败！"**);  }  }); } |

### 5.4.2后端代码

pinyougou-manager-web/src/main/java/com.pinyougou.manager.controller的SellerController.java

|  |
| --- |
| */\*\* 审核商家(修改商家状态) \*/* @GetMapping(**"/updateStatus"**) **public boolean** updateStatus(String sellerId, String status){  **try** {  **sellerService**.updateStatus(sellerId, status);  **return true**;  } **catch** (Exception e) {  e.printStackTrace();  **return false**;  } } |

pinyougou-interface/src/main/java/com.pinyougou.service/SellerService.java服务接口

|  |
| --- |
| */\*\* 修改商家状态 \*/* **void** updateStatus(String sellerId, String status); |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java服务实现

|  |
| --- |
| */\*\* 修改商家状态 \*/* **public void** updateStatus(String sellerId, String status){  **try**{  Seller seller = **new** Seller();  seller.setSellerId(sellerId);  seller.setStatus(status);  **sellerMapper**.updateByPrimaryKeySelective(seller);  }**catch**(Exception ex){  **throw new** RuntimeException(ex);  } } |

# 6.商家系统登录与安全控制

## 6.1需求分析

完成商家系统登录与安全控制，商家账号来自数据库，并实现密码加密。

## 6.2配置&登录页面

1. pom.xml

|  |
| --- |
| *<!-- 配置shiro安全框架 -->* <**dependency**>  <**groupId**>org.apache.shiro</**groupId**>  <**artifactId**>shiro-spring</**artifactId**> </**dependency**> |

1. web.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="http://java.sun.com/xml/ns/javaee"  xsi:schemaLocation="http://java.sun.com/xml/ns/javaee  http://java.sun.com/xml/ns/javaee/web-app\_3\_0.xsd"  version="3.0"**>   *<!-- 配置Spring委派过滤器代理 -->* <**filter**>  <**filter-name**>shiroFilter</**filter-name**>  <**filter-class**>org.springframework.web.filter.DelegatingFilterProxy</**filter-class**>  </**filter**>  <**filter-mapping**>  <**filter-name**>shiroFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 配置Spring MVC前端控制器(核心控制器) -->* <**servlet**>  <**servlet-name**>pinyougou</**servlet-name**>  <**servlet-class**>org.springframework.web.servlet.DispatcherServlet</**servlet-class**>  <**init-param**>  <**param-name**>contextConfigLocation</**param-name**>  <**param-value**>classpath:springmvc.xml,  classpath:applicationContext-shiro.xml</**param-value**>  </**init-param**>  <**load-on-startup**>1</**load-on-startup**>  </**servlet**>  <**servlet-mapping**>  <**servlet-name**>pinyougou</**servlet-name**>  <**url-pattern**>/</**url-pattern**>  </**servlet-mapping**>   *<!-- 配置全局编码过滤器 post -->* <**filter**>  <**filter-name**>characterEncodingFilter</**filter-name**>  <**filter-class**>org.springframework.web.filter.CharacterEncodingFilter</**filter-class**>  <**init-param**>  <**param-name**>encoding</**param-name**>  <**param-value**>UTF-8</**param-value**>  </**init-param**>  </**filter**>  <**filter-mapping**>  <**filter-name**>characterEncodingFilter</**filter-name**>  <**url-pattern**>/\*</**url-pattern**>  </**filter-mapping**>   *<!-- 欢迎列表 -->* <**welcome-file-list**>  <**welcome-file**>index.html</**welcome-file**>  </**welcome-file-list**> </**web-app**> |

1. pinyougou-shop-web/src/main/resources/applicationContext-shiro.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="utf-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 配置shiro作为Spring委派过滤器(shiro过滤器工厂) -->* <**bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 配置注入安全管理对象 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 配置登录页面访问URL -->* <**property name="loginUrl" value="/shoplogin.html"**/>  *<!--  配置权限管理URL过滤器链(配置顺序为自上而下)  1. user: 过滤器只要用户登录(isRemembered()==true or isAuthenticated()==true)即可访问成功  2. authc: 过滤器只要用户登录isAuthenticated()==true即可访问成功  3. anon : 匿名访问，不需要登录  4. logout : 安全退出  -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /img/\*\* = anon  /js/\*\* = anon  /plugins/\*\* = anon **/login = anon**  **/\*.html = anon**  **/seller/save = anon**  /logout = logout  /\*\* = user  </**value**>  </**property**>  </**bean**>  *<!-- 配置安全管理器 -->* <**bean id="securityManager"  class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 配置自定义认证域-->* <**property name="realm" ref="authorizingRealm"**/>  </**bean**>  *<!-- 配置自定义认证域 -->* <**bean id="authorizingRealm" class="com.pinyougou.shop.realm.SellerAuthorizingRealm"**>  </**bean**> </**beans**> |

1. shoplogin.html

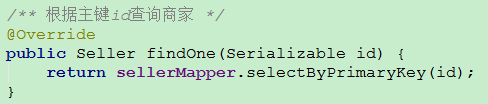
|  |
| --- |
| <**form class="sui-form" action="/login" method="post"**>  <**div class="input-prepend"**>  <**span class="add-on loginname"**></**span**>  <**input type="text" name="username"  placeholder="邮箱/用户名/手机号" class="span2 input-xfat"**/>  </**div**>  <**div class="input-prepend"**>  <**span class="add-on loginpwd"**></**span**>  <**input type="password" name="password"  placeholder="请输入密码" class="span2 input-xfat"**/>  </**div**>  <**div class="setting"**>  <**label class="checkbox inline"**>  <**input name="m1" type="checkbox" value="2" checked=""**/>自动登录  </**label**>  <**span class="forget"**>忘记密码？</**span**>  </**div**>  <**div class="logined"**>  <**input type="submit"   class="sui-btn btn-block btn-xlarge btn-danger"   value="登&nbsp;&nbsp;录"**/>  </**div**> </**form**> |

## 6.3自定义认证域

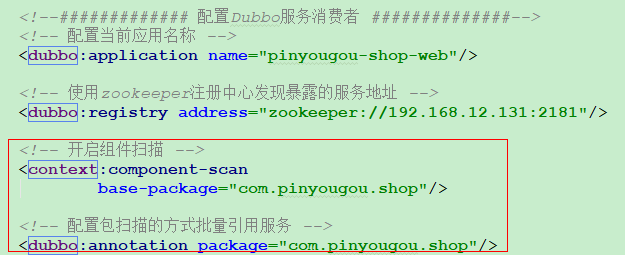
在pinyougou-shop-web/src/main/java/com.pinyougou.shop.realm包下，创建**SellerAuthorizingRealm.java**认证域

|  |
| --- |
| */\*\* 自定义认证域 \*/* **public class** SellerAuthorizingRealm **extends** AuthorizingRealm {   @Reference(timeout = 10000)  **private** SellerService **sellerService**;   */\*\* 授权方法 \*/* @Override  **protected** AuthorizationInfo doGetAuthorizationInfo(  PrincipalCollection principalCollection) {  **return null**;  }   */\*\* 身份认证的方法(subject.login()) \*/* @Override  **protected** AuthenticationInfo doGetAuthenticationInfo(  AuthenticationToken authenticationToken)  **throws** AuthenticationException {  **try** {   *// 获取登录用户名* String username = (String) authenticationToken.getPrincipal();  System.***out***.println(**"username = "** + username);  System.***out***.println(**"sellerService = "** + **sellerService**);   *// 根据用户名查询商家对象* Seller seller = **sellerService**.findOne(username);  *// 判断seller 是否为空，审核状态码是否为：1* **if** (seller != **null** && **"1"**.equals(seller.getStatus())) {  *// 返回身份认证信息封装对象* **return new** SimpleAuthenticationInfo(username,  seller.getPassword(), **this**.getName());  }  }**catch** (Exception ex){  **throw new** AuthenticationException(ex);  }  **return null**;  } } |

pinyougou-sellergoods-service/src/main/java/com.pinyougou.sellergoods.service.impl的SellerServiceImpl.java



pinyougou-shop-web\src\main\resources\springmvc.xml包扫描：



## 6.4登录控制器

pinyougou-shop-web\src\main\java\com\pinyougou\shop\controller\LoginController.java

|  |
| --- |
| **package** com.pinyougou.shop.controller;  **import** org.apache.shiro.SecurityUtils; **import** org.apache.shiro.authc.UsernamePasswordToken; **import** org.apache.shiro.subject.Subject; **import** org.springframework.stereotype.Controller; **import** org.springframework.web.bind.annotation.PostMapping; */\*\* 获取登录用户名控制器 \*/* **@Controller** **public class** LoginController {   */\*\* 用户登录 \*/* **@PostMapping("/login")**  **public** String login(String username, String password){  **try**{  */\*\*########### 让shiro框架帮我们作身份认证 ##########\*/  /\*\* 获取HttpSession相关的主体 \*/* Subject subject = SecurityUtils.*getSubject*();  */\*\* 创建用户名与密码令牌 \*/* UsernamePasswordToken token = **new** UsernamePasswordToken(username, password);  */\*\* 身份认证 \*/* subject.login(token);  */\*\* 判断用户是否通过身份认证 \*/* **if** (subject.isAuthenticated()){  **return "redirect:/admin/index.html"**;  }  }**catch**(Exception ex){  ex.printStackTrace();  }  **return "redirect:/shoplogin.html"**;  } } |

## 6.5密码加密

### 6.5.1商家入驻密码加密

商家申请入驻的密码要使用MD5算法加迭代次数进行加密存储，修改pinyougou-shop-web/src/main/java/com.pinyougou.shop.controller的

SellerController.java的save方法

|  |
| --- |
| */\*\* 申请入驻 \*/* @PostMapping(**"/save"**) **public boolean** save(@RequestBody Seller seller){  **try** {  ***// 密码加密* String password = new SimpleHash("md5",  seller.getPassword(), seller.getSellerId(), 5).toHex();  *// 设置密码* seller.setPassword(password);**  **sellerService**.save(seller);  **return true**;  }**catch** (Exception ex){  ex.printStackTrace();  }  **return false**; } |

### 6.5.2密码凭证匹配器

pinyougou-shop-web/src/main/resources/applicationContext-shiro.xml

|  |
| --- |
| *<?***xml version="1.0" encoding="UTF-8"***?>* <**beans xmlns="http://www.springframework.org/schema/beans"  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xsi:schemaLocation="http://www.springframework.org/schema/beans  http://www.springframework.org/schema/beans/spring-beans.xsd"**>   *<!-- 1. 配置Shiro过滤器工厂(拦截所有的请求URL) -->* <**bean id="shiroFilter" class="org.apache.shiro.spring.web.ShiroFilterFactoryBean"**>  *<!-- 设置安全管理器 -->* <**property name="securityManager" ref="securityManager"**/>  *<!-- 设置登录页面请求url -->* <**property name="loginUrl" value="/shoplogin.html"**/>   *<!-- 配置过滤器链(顺序从上而下) 配置哪些请求url用哪些过滤器  1．user: 过滤器只要用户登录  (isRemembered()==true or isAuthenticated()==true)即可访问成功  2．authc: 过滤器只要用户登录isAuthenticated()==true即可访问成功  3．anon : 匿名访问，不需要登录  4．logout : 安全退出  -->* <**property name="filterChainDefinitions"**>  <**value**>  /css/\*\* = anon  /img/\*\* = anon  /js/\*\* = anon  /plugins/\*\* = anon  /\*.html = anon  /seller/save = anon  /login = anon  /logout = logout  /\*\* = user  </**value**>  </**property**>  </**bean**>   *<!-- 2. 配置安全管理器 (身份认证与授权)-->* <**bean id="securityManager" class="org.apache.shiro.web.mgt.DefaultWebSecurityManager"**>  *<!-- 设置认证域-->* <**property name="realm" ref="authorizingRealm"**/>  </**bean**>   *<!-- 3. 配置自定义认证域Realm (提供身份认证与授权) -->* <**bean id="authorizingRealm" class="com.pinyougou.shop.realm.SellerAuthorizingRealm"**>  *<!-- 设置密码凭证匹配器 -->* <**property name="credentialsMatcher" ref="credentialsMatcher"**/>  </**bean**>   *<!-- 配置密码凭证匹配器 -->* <**bean id="credentialsMatcher" class="org.apache.shiro.authc.credential.HashedCredentialsMatcher"**>  *<!-- 设置加密算法的名称 -->* <**property name="hashAlgorithmName" value="md5"**/>  *<!-- 设置加密迭代次数 -->* <**property name="hashIterations" value="5"**/>  *<!-- 设置存储密码采用16进制编码的字符串 -->* <**property name="storedCredentialsHexEncoded" value="true"**/>  </**bean**> </**beans**> |

### 6.5.3自定义认证域

pinyougou-shop-web\src\main\java\com\pinyougou\shop\realm\SellerAuthorizingRealm.java

|  |
| --- |
| */\*\* 身份认证方法 \*/* @Override **protected** AuthenticationInfo doGetAuthenticationInfo(  AuthenticationToken authenticationToken) **throws** AuthenticationException {  **try** {  *// 获取登录用户名* String username = (String) authenticationToken.getPrincipal();  *// 根据用户名查询商家对象* Seller seller = **sellerService**.findOne(username);  *// 判断seller 是否为空，审核状态码是否为：1* **if** (seller != **null** && **"1"**.equals(seller.getStatus())) {  *// 返回身份认证信息封装对象* **return new** SimpleAuthenticationInfo(username, seller.getPassword(),  **ByteSource.Util.*bytes*(seller.getSellerId())**, **this**.getName());  }  }**catch** (Exception ex){  **throw new** AuthenticationException(ex);  }  **return null**; } |

## 6.6显示登录名

说明：pinyougou-shop-web参照运营商后台(pinyougou-manager-web)

|  |
| --- |
| **LoginController.java：**  */\*\* 获取登录用户名 \*/* @GetMapping(**"/findLoginName"**) @ResponseBody **public** Map<String, String> findLoginName(){  *// 获取登录用户名* String loginName = SecurityUtils.*getSubject*()  .getPrincipal().toString();  Map<String,String> data = **new** HashMap<>();  data.put(**"loginName"**, loginName);  **return** data; }  **js/controller/indexController.js：(已准备)**  *// 窗口加载完* ***window***.onload = **function** () {  **var** vue = **new Vue**({  **el** : **'#app'**, *// 元素绑定* **data** : { *// 数据模型* **loginName** : **''** *// 登录用户名* },  **methods** : { *// 操作方法* findLoginName : **function** () { *// 获取登录用户名  // 发送异步请示* **axios**.get(**"/findLoginName"**)  .then(**function**(response){  *// 获取响应数据* vue.**loginName** = response.**data**.**loginName**;  });  }  },  created : **function** () { *// 创建生命周期  // 调用获取登录用户名方法* **this**.findLoginName();  }  }); };  **admin/index.jsp：**  <**script src="/js/vue.min.js"**></**script**> <**script src="/js/axios.min.js"**></**script**> <**script src="/js/controller/indexController.js"**></**script**>  **把 chendahai 替换成 {{loginName}}** |

## 6.7退出登录

说明：pinyougou-shop-web参照运营商后台(pinyougou-manager-web)

|  |
| --- |
| **admin/index.html页面：**  <**div class="pull-right"**>  <**a href="/logout" class="btn btn-default btn-flat"**>注销</**a**> </**div**> |