Login.jsp

<%@ **page** contentType="text/html;charset=UTF-8" language="java" %>  
<%@ **taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <title>企业用车租赁系统 | 登录</title>  
 <!-- Tell the browser to be responsive to screen width -->  
 <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">  
 <!-- Bootstrap 3.3.7 -->  
 <link rel="stylesheet" href="../../../static/bower\_components/bootstrap/dist/css/bootstrap.min.css">  
 <!-- Font Awesome -->  
 <link rel="stylesheet" href="../../../static/bower\_components/font-awesome/css/font-awesome.min.css">  
 <!-- Ionicons -->  
 <link rel="stylesheet" href="../../../static/bower\_components/Ionicons/css/ionicons.min.css">  
 <!-- Theme style -->  
 <link rel="stylesheet" href="../../../static/css/AdminLTE.min.css">  
 <!-- iCheck -->  
 <link rel="stylesheet" href="../../../static/plugins/iCheck/square/blue.css">  
  
 <style type="text/css">  
 .bg {  
 margin: 0;  
 padding: 0;  
 font-family: sans-serif;  
 background-size: cover;  
  
  
 background: url('../../../static/bg1.jpg') no-repeat fixed center 0;  
 -webkit-background-size: cover;  
 -o-background-size: cover;  
 -moz-background-size: cover;  
 -ms-background-size: cover;  
 overflow: hidden;  
 }  
  
 .rBox{  
 background:#111111;  
 opacity:0.8;  
 border-radius:30px;  
 }  
  
 .inp{  
 border-radius: 3px;  
 }  
 </style>  
</head>  
<body class="hold-transition login-page bg" >  
<div >  
 <div class="login-box">  
  
 <div class="login-logo">  
 <a href="#"><b>企业公务车辆租赁系统</b></a>  
 </div>  
  
 <div class="login-box-body rBox">  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
 <p class="login-box-msg">欢迎登录车辆租赁管理系统</p>  
 <form action="**${**pageContext.request.contextPath**}**/auth/login" method="post">

({pageContext.request.contextPath}是JSP取得绝对路径的方法，等价于<%=request.getContextPath()%>)也就是取出部署的应用程序名或者是当前的项目名称  
 <div class="form-group has-feedback">  
 <input type="text" class="form-control inp" placeholder="请输入用户名" name="username">  
 <span class="glyphicon glyphicon-envelope form-control-feedback"></span>  
 </div>  
 <div class="form-group has-feedback">  
 <input type="password" class="form-control inp" placeholder="请输入密码" name="password">  
 <span class="glyphicon glyphicon-lock form-control-feedback"></span>  
 </div>  
 <div class="row">  
 <!-- /.col -->  
 <div class="col-xs-3"></div>  
 <div class="col-xs-6">  
 <button type="submit" class="btn btn-primary btn-block btn-flat">登录</button>  
 </div>  
 <div class="col-xs-3"></div>  
 <!-- /.col -->  
 </div>  
 </form>  
 <a href="/register" class="text-center" style="float: right">没有账号?</a>  
 </div>  
 </div>  
</div>  
 <!-- jQuery 3 -->前台Jquery是样式  
 <script src="../../../static/bower\_components/jquery/dist/jquery.min.js"></script>  
 <!-- Bootstrap 3.3.7 -->  
 </body>  
</html>

AuthController

package xyz.jfshare.car.system.controller;  
import org.apache.shiro.SecurityUtils;  
import org.apache.shiro.authc.AuthenticationException;  
import org.apache.shiro.authc.UsernamePasswordToken;  
import org.apache.shiro.subject.Subject;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.ui.Model;  
import org.springframework.util.DigestUtils;  
import org.springframework.web.bind.annotation.ModelAttribute;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import xyz.jfshare.car.system.entity.User;  
import xyz.jfshare.car.system.service.UserService;

import xyz.jfshare.car.system.util.MD5Utils;  
import xyz.jfshare.car.system.util.RandomUtil;  
import java.util.Map;  
@Controller  
@RequestMapping("/auth/")  
public class AuthController {  
 @Autowired  
 private UserService userService;  
 @PostMapping("login")  
 public String login(String username, String password, Model model){  
 User user = userService.getByUsername(username);  
 if (user == null){  
 model.addAttribute("msg","用户名不存在");  
 return "redirect:/login";  
 }

3、Spring的DigestUtils实现md5码加密：

这是一个非常好用的使用MD5+salt加密的工具类。使用这个工具类，非常简单，

从前台拿到密码passwd，直接MD5Util.getEncryptedPwd(passwd)就可以返回一个长度为56的字符串，

可以用来保存到数据库中，相反，登录的时候，因为MD5加密是不可逆的运算，只能拿用户输入的密码走一遍MD5+salt加密之后，

跟数据库中的passwd比较，看是否一致，一致时密码相同，登录成功.

password = DigestUtils.*md5DigestAsHex*((password+user.getSalt()).getBytes());采用GBK方式编码的

是最常见的用户名/密码的认证机制;同时,由于它实现了RememberMeAuthenticationToken接口，我们可以通过令牌设置“记住我”的功能。 但是，“已记住”和“已认证”是有区别的:已记住的用户仅仅是非匿名用户，你可以通过subject.getPrincipals()获取用户信息。但是它并非是认证通过的用户，当你访问需要认证用户的功能时，你仍然需要重新提交认证信息。这一区别可以参考淘宝网站，网站会默认记住登录的用户，再次访问网站时，对于非敏感的页面功能，页面上会显示记住的用户信息，但是当你访问网站账户信息时仍然需要再次进行登录认证。  
 UsernamePasswordToken token = new UsernamePasswordToken(username,password); // 把用户名和密码封装为 UsernamePasswordToken 对象  
 Subject subject = SecurityUtils.*getSubject*();获取当前的 Subject.

// 测试当前的用户是否已经被认证. 即是否已经登录. // 调动 Subject 的 isAuthenticated()  
 try {  
 subject.login(token); // 执行登录  
 return "redirect:/main";  
 }catch (AuthenticationException ae){ // 所有认证时异常的父类.  
 model.addAttribute("msg","密码错误");  
 return "redirect:/login";  
 }  
 }  
  
 @RequestMapping("main")  
 public String main(@ModelAttribute("data") Map<String,Object> data,Model model){  
 model.addAllAttributes(data);  
 return "main/index";  
 }  
  
 @PostMapping("register")  
 public String register(String nickname,String username, String password,String rePassword,Integer roleId,Model model){  
 User one = userService.getByUsername(username);  
 if (one != null){  
 model.addAttribute("msg","该用户名已经注册");  
 return "redirect:/register";  
 }  
 if (!password.equals(rePassword)){  
 model.addAttribute("msg","两次输入的密码不一致");  
 return "redirect:/register";  
 }  
 if (password.length() < 6){  
 model.addAttribute("msg","密码不能少于6位");  
 return "redirect:/register";  
 }  
 String salt = RandomUtil.*getRandomString*(6);  
 User user = new User();  
 user.setNickname(nickname);  
 user.setUsername(username);  
 user.setRoleId(roleId);  
 user.setSalt(salt);  
 user.setPassword(MD5Utils.*getMD5*(password+salt));

RandomUtil.getRandomString salt获取随机字符串  
  
 boolean res = userService.save(user);  
 if (!res){  
 model.addAttribute("msg","注册失败,请联系管理员");  
 return "redirect:/register";  
 }  
 model.addAttribute("msg","注册成功");  
 return "redirect:/login";  
 }  
  
 @RequestMapping("logout")  
 public String logout(){  
 Subject subject = SecurityUtils.*getSubject*();  
 subject.logout();  
 return "redirect:/login";  
 }  
}

UserServiceImpl

package xyz.jfshare.car.system.service.impl;  
  
import com.baomidou.mybatisplus.extension.service.impl.ServiceImpl;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import xyz.jfshare.car.system.dao.UserMapper;  
import xyz.jfshare.car.system.entity.User;  
import xyz.jfshare.car.system.service.UserService;  
import java.util.List;  
*/\*\*  
 \** ***@author*** *\* Description: 用户  
 \*/*@Service  
public class UserServiceImpl extends ServiceImpl<UserMapper, User> implements UserService { mybatics的Iservice

@Autowired  
 private UserMapper userMapper;  
  
 @Override  
 public User getByUsername(String username) {  
 return userMapper.getByUsername(username);  
 }  
  
 @Override  
 public User getByUserId(Integer userId) {  
 return userMapper.getByUserId(userId);  
 }  
  
 @Override  
 public List<User> listUser(User user) {  
 return userMapper.listUser(user);  
 }  
  
 @Override  
 public Integer countUser(User user) {  
 return userMapper.countUser(user);  
 }  
}

user.xml

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
 PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="xyz.jfshare.car.system.dao.UserMapper">  
 <select id="getByUsername" resultType="xyz.jfshare.car.system.entity.User">  
 select \* from sys\_user where username = #{username}  
 </select>  
 <select id="getByUserId" resultType="xyz.jfshare.car.system.entity.User">  
 select  
 su.id,  
 su.nickname,  
 su.username,  
 su.status,  
 su.icon,  
 su.role\_id,  
 su.update\_time,  
 sr.description as roleName  
 from sys\_user su  
 left join sys\_role sr on su.role\_id = sr.id  
 where su.id = #{userId}  
 and su.status = 0  
 </select>  
  
 <select id="listUser" resultType="xyz.jfshare.car.system.entity.User">  
 select  
 su.id,  
 su.nickname,  
 su.username,  
 su.status,  
 su.icon,  
 su.role\_id,  
 su.create\_time,  
 su.update\_time,  
 sr.description as roleName  
 from sys\_user su  
 left join sys\_role sr on su.role\_id = sr.id  
 where su.status = 0  
 <if test="nickname != null">  
 and su.nickname like CONCAT('%',#{nickname},'%')  
 </if>  
  
  
 order by su.id desc  
 </select>  
  
 <select id="countUser" resultType="java.lang.Integer">  
 SELECT COUNT(su.id)  
 from sys\_user su  
 left join sys\_role sr on su.role\_id = sr.id  
 where su.status = 0  
 <if test="nickname != null">  
 and su.nickname like CONCAT('%',#{nickname},'%')  
 </if>  
 </select>  
</mapper>

Register.jsp

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<%@ **taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <title>公务车辆租赁系统 | 注册</title>  
 <!-- Tell the browser to be responsive to screen width -->  
 <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">  
 <!-- Bootstrap 3.3.7 -->  
 <link rel="stylesheet" href="../../../static/bower\_components/bootstrap/dist/css/bootstrap.min.css">  
 <!-- Font Awesome -->  
 <link rel="stylesheet" href="../../../static/bower\_components/font-awesome/css/font-awesome.min.css">  
 <!-- Ionicons -->  
 <link rel="stylesheet" href="../../../static/bower\_components/Ionicons/css/ionicons.min.css">  
 <!-- Theme style -->  
 <link rel="stylesheet" href="../../../static/css/AdminLTE.min.css">  
 <style type="text/css">  
 .bg {  
 margin: 0;  
 padding: 0;  
 font-family: sans-serif;  
 background-size: cover;  
  
  
 background: url('../../../static/bg2.jpg') no-repeat fixed center 0;  
 -webkit-background-size: cover;  
 -o-background-size: cover;  
 -moz-background-size: cover;  
 -ms-background-size: cover;  
 overflow: hidden;  
 }  
  
 .rBox{  
 background:#111111;  
 opacity:0.8;  
 border-radius:30px;  
 }  
  
 .inp{  
 border-radius: 3px;  
 }  
 </style>  
</head>  
<body class="hold-transition register-page bg">  
  
<div class="register-box">  
 <div class="register-logo">  
 <a href="#" style="color: #3c8dbc"><b>企业公务用车系统</b>注册</a>  
 </div>  
  
 <div class="register-box-body rBox">  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
 <p class="login-box-msg">欢迎注册车辆租赁管理系统</p>  
  
 <form action="/auth/register" method="post">  
 <div class="form-group has-feedback">  
 <input name="nickname" type="text" class="form-control inp" placeholder="请输入昵称">  
 <span class="glyphicon glyphicon-text-color form-control-feedback"></span>  
 </div>  
 <div class="form-group has-feedback">  
 <select name="roleId" class="form-control" id="inputGender">  
 <option value="2">用车公司</option>  
 <option value="3">司机</option>  
 </select>  
 <span class="glyphicon glyphicon-user form-control-feedback"></span>  
 </div>  
 <div class="form-group has-feedback">  
 <input name="username" type="text" class="form-control inp" placeholder="请输入用户名">  
 <span class="glyphicon glyphicon-user form-control-feedback"></span>  
 </div>  
 <div class="form-group has-feedback">  
 <input name="password" type="password" class="form-control inp" placeholder="请输入密码">  
 <span class="glyphicon glyphicon-lock form-control-feedback"></span>  
 </div>  
 <div class="form-group has-feedback">  
 <input name="rePassword" type="password" class="form-control inp" placeholder="请再次输入密码">  
 <span class="glyphicon glyphicon-log-in form-control-feedback"></span>  
 </div>  
 <div class="row">  
  
 <div class="col-xs-3"></div>  
 <div class="col-xs-6">  
 <button type="submit" class="btn btn-primary btn-block btn-flat inp" style="float: right">注册</button>  
  
 </div>  
 <div class="col-xs-3"></div>  
 </div>  
 </form>  
  
 <a href="login.jsp" class="text-center">已有账号?</a>  
 </div>  
 <!-- /.form-box -->  
</div>  
  
<!-- jQuery 3 -->  
<script src="../../../static/bower\_components/jquery/dist/jquery.min.js"></script>  
<!-- Bootstrap 3.3.7 -->  
<script src="../../../static/bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>  
</body>  
</html>

RandomUtil.java

package xyz.jfshare.car.system.util;  
import java.util.Random;  
public class RandomUtil {  
 */\*\*  
 \* 获取指定位数的随机字符串  
 \** ***@param*** *length  
 \** ***@return*** *\*/* public static String getRandomString(int length) {  
 String randomString = "qwertyuiopasdfghjklzxcvbnm1234567890";  
 int lengths = randomString.length();  
 StringBuffer stringBuffer = new StringBuffer(length);  
 Random random = new Random();  
 for (int i = 0; i < length; i++) {  
 stringBuffer.append(randomString.charAt(random.nextInt(lengths)));获取随机数追加到字符后面  
 }  
 return stringBuffer.toString();  
 }  
}

MD5Utils.java

package xyz.jfshare.car.system.util;  
  
import java.io.UnsupportedEncodingException;  
import java.math.BigInteger;  
import java.security.MessageDigest;  
import java.security.NoSuchAlgorithmException;  
  
*/\*\*  
 \* Created by L.key on 2017/6/2.  
 \*/*public class MD5Utils {  
  
 public static String getMD5(String str) {  
 try {  
 MessageDigest messageDigest = MessageDigest.*getInstance*("MD5");  
  
 // 使用指定的 byte 数组对摘要进行最后更新，然后完成摘要计算。  
 // 也就是说，此方法首先调用 update(input)，向 update 方法传递 input 数组，然后调用 digest()。  
 byte[] bytes = messageDigest.digest(str.getBytes("UTF-8"));  
  
 // digest()最后确定返回md5 hash值，返回值为8为字符串。因为md5 hash值是16位的hex值，实际上就是8位的字符  
 // BigInteger函数则将8位的字符串转换成16位hex值，用字符串来表示；得到字符串形式的hash值  
 return new BigInteger(1, bytes).toString(16);  
 } catch (NoSuchAlgorithmException e) {  
 e.printStackTrace();  
 } catch (UnsupportedEncodingException e) {  
 e.printStackTrace();  
 }  
 return null;

}  
}

MessageDigest 类为应用程序提供信息摘要算法的功能，如 MD5 或 SHA 算法。信息摘要是安全的单向哈希函数，它接收任意大小的数据，输出固定长度的哈希值。

MessageDigest 对象开始被初始化。该对象通过使用 [update](http://jszx-jxpt.cuit.edu.cn/JavaAPI/java/security/MessageDigest.html#update(byte)) 方法处理数据。任何时候都可以调用 [reset](http://jszx-jxpt.cuit.edu.cn/JavaAPI/java/security/MessageDigest.html#reset()) 方法重置摘要。一旦所有需要更新的数据都已经被更新了，应该调用 [digest](http://jszx-jxpt.cuit.edu.cn/JavaAPI/java/security/MessageDigest.html#digest()) 方法之一完成哈希计算。

对于给定数量的更新数据，digest 方法只能被调用一次。digest 被调用后，MessageDigest 对象被重新设置成其初始状态。

package xyz.jfshare.car.system.util;  
import org.springframework.util.CollectionUtils;  
import org.springframework.util.StringUtils;  
import java.util.HashSet;  
import java.util.Set;  
  
public class StringUtil {  
 */\*\*  
 \* 分割字符串进SET  
 \*/* @SuppressWarnings("unchecked")  
 public static Set<String> split(String str) {  
 Set<String> set = new HashSet<>();  
 if (StringUtils.*isEmpty*(str))  
 return set;  
 set.addAll(CollectionUtils.*arrayToList*(str.split(",")));  
 return set;  
 }  
}

Index.jsp

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<%@**taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<%@**include** file="../common/head.jsp"%>  
<%@**include** file="../common/topbar.jsp"%>  
<%@**include** file="../common/sidebar.jsp"%>  
<div class="content-wrapper">  
 <section class="content">  
 <div style="text-align: center;line-height: 200px;">  
 <h1 style="margin-top: 20%">欢迎来到公务车辆租赁管理系统<img src="../../../static/favicon.ico"></h1>  
 </div>  
 </section>  
  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
</div>  
  
<%@**include** file="../common/bottom.jsp"%>

Head.jsp

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <meta http-equiv="X-UA-Compatible" content="IE=edge">  
 <title>车辆租赁管理系统</title>  
 <!-- Tell the browser to be responsive to screen width -->  
 <meta content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no" name="viewport">  
 <!-- Bootstrap 3.3.7 -->  
 <link rel="stylesheet" href="../../../static/bower\_components/bootstrap/dist/css/bootstrap.min.css">  
 <!-- Font Awesome -->  
 <link rel="stylesheet" href="../../../static/bower\_components/font-awesome/css/font-awesome.min.css">  
 <!-- Ionicons -->  
 <link rel="stylesheet" href="../../../static/bower\_components/Ionicons/css/ionicons.min.css">  
 <!-- Theme style -->  
 <link rel="stylesheet" href="../../../static/css/AdminLTE.min.css">  
 <!-- AdminLTE Skins. Choose a skin from the css/skins  
 folder instead of downloading all of them to reduce the load. -->  
 <link rel="stylesheet" href="../../../static/css/skins/\_all-skins.min.css">  
 <!-- Morris chart -->  
 <link rel="stylesheet" href="../../../static/bower\_components/morris.js/morris.css">  
 <!-- jvectormap -->  
 <link rel="stylesheet" href="../../../static/bower\_components/jvectormap/jquery-jvectormap.css">  
 <!-- Date Picker -->  
 <link rel="stylesheet" href="../../../static/bower\_components/bootstrap-datepicker/dist/css/bootstrap-datepicker.min.css">  
 <!-- Daterange picker -->  
 <link rel="stylesheet" href="../../../static/bower\_components/bootstrap-daterangepicker/daterangepicker.css">  
 <!-- bootstrap wysihtml5 - text editor -->  
 <link rel="stylesheet" href="../../../static/plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.min.css">  
  
 <link rel="stylesheet" href="../../../static/plugins/bootstrap-slider/slider.css">  
 <!-- HTML5 Shim and Respond.js IE8 support of HTML5 elements and media queries -->  
 <!-- WARNING: Respond.js doesn't work if you view the page via file:// -->  
 <!--[if lt IE 9]>  
 <script src="https://oss.maxcdn.com/html5shiv/3.7.3/html5shiv.min.js"></script>  
 <script src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>  
 <![endif]-->  
  
 <style>  
 .example-modal .modal {  
 position: relative;  
 top: auto;  
 bottom: auto;  
 right: auto;  
 left: auto;  
 display: block;  
 z-index: 1;  
 }  
  
 .example-modal .modal {  
 background: transparent **!important**; background:transparent。意思就是背景透明。  
 }  
 </style>  
  
</head>  
<body class="hold-transition skin-blue sidebar-mini">  
<div class="wrapper">

Topbar.jsp

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<%@**taglib** uri="http://shiro.apache.org/tags" prefix="shiro" %>  
<%@**page** import="java.util.\*, java.text.SimpleDateFormat"%>  
<header class="main-header">  
 <!-- Logo -->  
 <a href="#" class="logo">  
 <!-- mini logo for sidebar mini 50x50 pixels -->  
 <span class="logo-mini"><i class="fa fa-fw fa-genderless"></i></span>  
 <!-- logo for regular state and mobile devices -->  
 <span class="logo-lg"><b>车辆</b>管理系统</span>  
 </a>  
 <!-- Header Navbar: style can be found in header.less -->  
 <nav class="navbar navbar-static-top">  
 <!-- Sidebar toggle button-->  
 <a href="#" class="sidebar-toggle" data-toggle="push-menu" role="button">  
 <span class="sr-only">Toggle navigation</span>  
 </a>  
  
 <span style="font-size: 26px;line-height: 50px;margin: auto 30%">欢迎<**shiro:principal** property="nickname"/>登录 </span>  
  
 <div class="navbar-custom-menu">  
 <ul class="nav navbar-nav">  
 <li class="dropdown user user-menu">  
 <a href="#" class="dropdown-toggle" data-toggle="dropdown">  
 <img src="<**shiro:principal** property="icon"/>" class="user-image" alt="User Image">是要把一个带有icon属性的对象转换为Prinipal后保存在session中，才能在页面上正确显示结果的。(icon是数据库里存路径的地方) 在登录的方法中，调用了subject.login(token)后，还要手动利用principal和realmName构造SimpleAuthenticationInfo对象，其实这里的pricipal是一个Object，就是我们的带有username属性的实体对象，然后将SimpleAuthenticationInfo对象存放在session中。

1. <shiro:principal property="icon "/> 相当于((User)Subject.getPrincipals()).getIcon()

<span class="hidden-xs"><**shiro:principal** property="nickname"/></span>  
 </a>  
 <ul class="dropdown-menu">  
 <!-- User image -->  
 <li class="user-header">  
 <img src="<**shiro:principal** property="icon"/>" class="img-circle" alt="User Image">  
  
 <p>  
 <**shiro:principal** property="nickname"/>  
 <small>  
 **<%** Date dat = new Date();  
 SimpleDateFormat sdf = new SimpleDateFormat("yyyy年 MM月 dd日 HH:mm:ss");  
 String date = sdf.format(dat);  
 **%>  
 <%=**date**%>** </p>  
 </li>  
 <!-- Menu Footer-->  
 <li class="user-footer">  
<%-- <div class="pull-left">--%>  
<%-- <a href="/user/info" class="btn btn-default btn-flat">个人信息</a>--%>  
<%-- </div>--%>  
<%--&lt;%&ndash; <div>&ndash;%&gt;--%>  
<%--&lt;%&ndash; </div>&ndash;%&gt;--%>  
<%-- <div class="pull-right">--%>  
<%-- <button type="button" class="btn btn-default btn-flat" data-toggle="modal" data-target="#update-password" style="width: 100px; float:left">--%>  
<%-- 修改密码--%>  
<%-- </button>--%>  
<%-- <a href="/auth/logout" class="btn btn-default btn-flat">退出</a>--%>  
<%-- </div>--%>  
 <div class="btn-group" style="width: 100%">  
 <a href="/user/info" class="btn btn-default" style="width: 40%">  
 个人信息  
 </a>  
 <a class="btn btn-default" style="width: 40%" data-toggle="modal" data-target="#update-password">  
 修改密码  
 </a>  
 <a href="/auth/logout" class="btn btn-default" style="width: 20%">  
 退出  
 </a>  
 </div>  
 </li>  
 </ul>  
 </li>  
 </ul>  
 </div>  
 </nav>  
 </header>  
  
<div class="modal modal-info fade" id="update-password">  
 <form class="form-horizontal" method="post" action="/user/updatePassword">  
  
 <div class="modal-dialog">  
 <div class="modal-content">  
 <div class="modal-header">  
 <button type="button" class="close" data-dismiss="modal" aria-label="Close">  
 <span aria-hidden="true">&times;</span></button>  
 <h4 class="modal-title">修改密码</h4>  
 </div>  
 <div class="modal-body">  
  
 <div class="box-body">  
 <div class="form-group">  
 <label for="inputOld" class="col-sm-2 control-label">旧密码</label>  
  
 <div class="col-sm-10">  
 <input name="oldPwd" type="password" class="form-control" id="inputOld" placeholder="请输入旧密码">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputNew" class="col-sm-2 control-label">新密码</label>  
  
 <div class="col-sm-10">  
 <input name="newPwd" type="password" class="form-control" id="inputNew" placeholder="请输入新密码">  
 </div>  
 </div>  
 </div>  
 <!-- /.box-footer -->  
 </div>  
 <div class="modal-footer">  
 <button type="button" class="btn btn-outline pull-left" data-dismiss="modal">关闭</button>  
 <button type="submit" class="btn btn-outline">提交</button>  
 </div>  
  
 </div>  
 <!-- /.modal-content -->  
 </div>  
 </form>  
  
 <!-- /.modal-dialog -->  
</div>

Sidebar.jsp

<%@ **page** import="xyz.jfshare.car.system.entity.Permission" %>  
<%@ **page** import="org.apache.shiro.SecurityUtils" %>  
<%@ **page** import="xyz.jfshare.car.system.entity.User" %>  
<%@**page** contentType="text/html;charset=UTF-8" pageEncoding="UTF-8" %>  
<%@**taglib** uri="http://shiro.apache.org/tags" prefix="shiro" %>  
<%@**taglib** uri="http://com.web.output.tag" prefix="mytag"%>  
  
**<%** String activeUri = request.getParameter("activeUri");  
**%>**往左边放<aside class="main-sidebar">  
 <!-- sidebar: style can be found in sidebar.less -->  
 定义章节<section class="sidebar">  
 <!-- Sidebar user panel -->  
 <div class="user-panel">  
 <div class="pull-left image">  
 <img src="<**shiro:principal** property="icon"/>" class="img-circle" alt="User Image">  
 </div>  
 <div class="pull-left info">  
 <p><**shiro:principal** property="nickname"/></p>  
 <a href="#"><i class="fa fa-circle text-success"></i> Online</a>  
 </div>  
 </div>  
 <!-- sidebar menu: : style can be found in sidebar.less -->  
 <ul class="sidebar-menu" data-widget="tree">  
 <**mytag:menu**/>  
 </ul>  
 </section>  
 <!-- /.sidebar -->  
</aside>  
  
<script type="text/javascript">  
 **var** js = document.getElementById('/**<%=**activeUri**%>**');  
 js.className='active';  
</script>

Bottom.jsp

<%@**page** contentType="text/html;charset=UTF-8" pageEncoding="UTF-8" %>  
 <footer class="main-footer">  
 <div class="pull-right hidden-xs">  
 <b>Version</b> 1.0.0  
 </div>  
 <strong>Copyright &copy; 2020-Now.</strong> 公务车辆管理系统  
 </footer>  
</div>  
  
  
  
<!-- jQuery 3 -->  
<script src="../../../static/bower\_components/jquery/dist/jquery.min.js"></script>  
<!-- jQuery UI 1.11.4 -->  
<script src="../../../static/bower\_components/jquery-ui/jquery-ui.min.js"></script>  
<!-- Resolve conflict in jQuery UI tooltip with Bootstrap tooltip -->  
<script>  
 $.widget.bridge('uibutton', $.ui.button);  
</script>  
<!-- Bootstrap 3.3.7 -->  
<script src="../../../static/bower\_components/bootstrap/dist/js/bootstrap.min.js"></script>  
<!-- Morris.js charts -->  
<script src="../../../static/bower\_components/raphael/raphael.min.js"></script>  
<script src="../../../static/bower\_components/morris.js/morris.min.js"></script>  
<!-- Sparkline -->  
<script src="../../../static/bower\_components/jquery-sparkline/dist/jquery.sparkline.min.js"></script>  
<!-- jvectormap -->  
<script src="../../../static/plugins/jvectormap/jquery-jvectormap-1.2.2.min.js"></script>  
<script src="../../../static/plugins/jvectormap/jquery-jvectormap-world-mill-en.js"></script>  
<!-- jQuery Knob Chart -->  
<script src="../../../static/bower\_components/jquery-knob/dist/jquery.knob.min.js"></script>  
<!-- daterangepicker -->  
<script src="../../../static/bower\_components/moment/min/moment.min.js"></script>  
<script src="../../../static/bower\_components/bootstrap-daterangepicker/daterangepicker.js"></script>  
<!-- datepicker -->  
<script src="../../../static/bower\_components/bootstrap-datepicker/dist/js/bootstrap-datepicker.min.js"></script>  
<!-- Bootstrap WYSIHTML5 -->  
<script src="../../../static/plugins/bootstrap-wysihtml5/bootstrap3-wysihtml5.all.min.js"></script>  
<!-- Slimscroll -->  
<script src="../../../static/bower\_components/jquery-slimscroll/jquery.slimscroll.min.js"></script>  
<!-- FastClick -->  
<script src="../../../static/bower\_components/fastclick/lib/fastclick.js"></script>  
<!-- AdminLTE App -->  
<script src="../../../static/js/adminlte.min.js"></script>  
<!-- AdminLTE dashboard demo (This is only for demo purposes) -->  
<script src="../../../static/js/pages/dashboard.js"></script>  
<!-- AdminLTE for demo purposes -->  
<script src="../../../static/js/demo.js"></script>  
</body>  
</html>

shiro/filter/FilterManager.java

package xyz.jfshare.car.system.shiro.filter;  
import lombok.extern.slf4j.Slf4j;  
import org.springframework.stereotype.Component;  
import java.util.LinkedHashMap;  
import java.util.Map;  
@Component（把普通pojo实例化到spring容器中，相当于配置文件中的<bean id=""class=""/>）  
@Slf4j 日志  
public class FilterManager {  
 */\*\*  
 \* 初始化过滤器链 路径为key, 可用角色为value  
 \** ***@return*** *\*/* public Map<String, String> initFilterChain() {  
 Map<String, String> map = new LinkedHashMap<>();  
 map.put("/\*\*.js", "anon");  
 map.put("/druid/\*\*", "anon");  
 map.put("/auth/\*\*", "anon");  
 map.put("/403", "anon");  
 map.put("/templates/\*\*","authc"); anon:无需认证即可访问 authc:需要认证才可访问  
 map.put("/web/\*\*","authc");  
 return map;  
 }  
}

Shiro/rules/jwtRules.java

package xyz.jfshare.car.system.shiro.rules;  
import lombok.extern.slf4j.Slf4j;  
import org.springframework.util.StringUtils;  
import xyz.jfshare.car.system.util.StringUtil;  
import java.io.Serializable;  
import java.util.Set;  
@Slf4j  
public class JwtRule implements Serializable {  
 private static final long *serialVersionUID* = 1L;  
 private String url;  
 private String rules;  
 public String getUrl() {  
 return url; }  
 public String getRules() {  
 return rules;  
 }  
 public StringBuilder toFilterChain() {  
 StringBuilder stringBuilder = new StringBuilder();  
 if (StringUtils.*isEmpty*(this.url)) {  
 return null;  
 }  
 Set<String> rules = StringUtil.*split*(this.rules);  
 if (!StringUtils.*isEmpty*(rules) && rules.contains("anon")) {  
 stringBuilder.append("anon");  
 }  
 if (!StringUtils.*isEmpty*(rules) && !rules.contains("anon")) {  
 stringBuilder.append("jwt" + "[" + this.rules + "]");  
 }  
 return stringBuilder.length() > 0 ? stringBuilder : null;  
 }  
 @Override  
 public String toString() {  
 return "JwtRule{" + "url='" + url + '\'' + ", rules='" + rules + '\'' + '}';  
 }  
}

Shiro/realm/Myrealm

package xyz.jfshare.car.system.shiro.realm;  
import com.baomidou.mybatisplus.core.conditions.query.QueryWrapper;  
import org.apache.shiro.authc.\*;  
import org.apache.shiro.authz.AuthorizationInfo;  
import org.apache.shiro.authz.SimpleAuthorizationInfo;  
import org.apache.shiro.realm.AuthorizingRealm;  
import org.apache.shiro.subject.PrincipalCollection;  
import org.springframework.beans.factory.annotation.Autowired;  
import xyz.jfshare.car.system.entity.Permission;  
import xyz.jfshare.car.system.entity.Role;  
import xyz.jfshare.car.system.entity.User;  
import xyz.jfshare.car.system.service.PermissionService;  
import xyz.jfshare.car.system.service.RoleService;  
import xyz.jfshare.car.system.service.UserService;  
import java.util.HashSet;  
import java.util.List;  
import java.util.Set;  
public class MyRealm extends AuthorizingRealm {  
 @Autowired  
 private UserService userService;  
 @Autowired  
 private PermissionService permissionService;  
 @Autowired  
 private RoleService roleService;  
  
 @Override  
 public Class getAuthenticationTokenClass() {  
 return UsernamePasswordToken.class; }  
 @Override  
 public boolean supports(AuthenticationToken token) {  
 return token instanceof UsernamePasswordToken;  
 }  
 */\*\*  
 \* 执行授权逻辑  
 \* 只有当需要检测用户权限的时候才会调用此方法，例如checkRole,checkPermission之类的  
 \** ***@param*** *principalCollection  
 \** ***@return*** *\*/* @Override  
 protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principalCollection) {  
  
 User user = (User) principalCollection.getPrimaryPrincipal();  
 List<Permission> allPerm = user.getAllPerm();  
 String role = roleService.getById(user.getRoleId()).getName();  
 SimpleAuthorizationInfo simpleAuthorizationInfo = new SimpleAuthorizationInfo();  
 Set<String> permissions = new HashSet<>();  
 Set<String> roles = new HashSet<>();  
  
 roles.add(role);  
 if (!roles.isEmpty()) {  
 simpleAuthorizationInfo.setRoles(roles);  
 }  
  
  
 for (Permission permission : allPerm) {  
 permissions.add(permission.getUri());  
 }  
 if (!permissions.isEmpty()){  
 simpleAuthorizationInfo.setStringPermissions(permissions);  
 }  
 return simpleAuthorizationInfo;  
 }  
  
 */\*\*  
 \* 执行认证逻辑  
 \* 默认使用此方法进行用户名正确与否验证，错误抛出异常即可。  
 \** ***@param*** *authenticationToken  
 \** ***@return*** *\** ***@throws*** *AuthenticationException  
 \*/* @Override  
 protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authenticationToken) throws AuthenticationException {  
 UsernamePasswordToken token = (UsernamePasswordToken) authenticationToken;  
 String username = token.getUsername();  
 String password = String.*valueOf*(token.getPassword());  
 User user = userService.getOne(  
 new QueryWrapper<User>()  
 .eq("username",username)  
 .eq("password",password)  
 );  
  
 Role role = roleService.getById(user.getRoleId());  
 user.setRoleName(role.getDescription());  
  
 List<Permission> menuPerm = permissionService.getMenu(role.getId());  
 List<Permission> allPerm = permissionService.getAll(role.getId());  
  
 user.setAllPerm(allPerm);  
 user.setMenuPerm(menuPerm);  
 //principal是认证的实体信息，也可以是username，也可以是的用户的实体类对象  
 //credentials是密码  
 //realmName是当前realm名字，调用getName()即可  
  
// ByteSource salt = ByteSource.Util.bytes(user.getSalt());  
// System.out.println("salt" + user.getSalt());  
 return new SimpleAuthenticationInfo(user,user.getPassword(),getName());  
 }  
}

Shiro/Simpletag/HasAnyPermissionTag.java

package xyz.jfshare.car.system.simpleTag;  
import org.apache.shiro.subject.Subject;  
import org.apache.shiro.web.tags.PermissionTag;  
*/\*\*  
 \* 扩展标签，具有列出权限中的任意一个  
 \** ***@author*** *\*  
 \*/*public class HasAnyPermissionTag extends PermissionTag {private static final long *serialVersionUID* = 1L;  
 public HasAnyPermissionTag() {  
 }  
 protected boolean showTagBody(String permissions) {  
 boolean hasAnyPermissions = false;  
 Subject subject = getSubject();  
 if (subject != null) {  
 for (String role : permissions.split(",")) {  
 if (subject.isPermitted(role.trim())) {  
 hasAnyPermissions = true;  
 break;} }}  
 return hasAnyPermissions; }}

Shiro/Simpletag/MenuPerm.java

package xyz.jfshare.car.system.simpleTag;  
  
import org.apache.shiro.SecurityUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import xyz.jfshare.car.system.entity.Permission;  
import xyz.jfshare.car.system.entity.User;  
import xyz.jfshare.car.system.service.PermissionService;  
  
import javax.servlet.jsp.tagext.SimpleTagSupport;  
import java.io.IOException;  
import java.util.List;  
  
public class MenuPerm extends SimpleTagSupport //自定义标签库{

通常一个自定义标签会包含3部分内容：标签处理类，tld标签描述文件，jsp文件（使用）。  
  
 @Override  
 public void doTag() throws IOException {  
 User user = (User) SecurityUtils.*getSubject*().getPrincipal();

返回一个java.security.Principal 对象，该对象包含当前授权用户的名称，使用request.getUserPrincipal().getName()得到用户名没有通过认证就没有返回值，是用户登录后才有值的，通过了JAAS认证，也就是登录。

List<Permission> menuPerm = user.getMenuPerm();  
 for (Permission permission : menuPerm) {  
 this.getJspContext().getOut().write(

getJspContext().getOut();**等价于**Response().getWriter()；  
 "<li id="+permission.getUri()+" >\n" +  
 " <a href=\""+permission.getUri()+"/list\">\n" +  
 " <i class=\"fa fa-th\"></i>\n" +  
 " <span>"+permission.getName()+"</span>\n" +  
 " <span class=\"pull-right-container\"></span>\n" +  
 " </a>\n" +  
 "</li>");  
 }  
 }  
}

ApplicationConfig.java

package xyz.jfshare.car.system.config;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.EnableAspectJAutoProxy;  
import org.springframework.context.annotation.FilterType;  
import org.springframework.stereotype.Controller;  
@Configuration  
@ComponentScan自动扫描组件(basePackages = {"xyz.jfshare.car.system"},  
 excludeFilters = {  
 @ComponentScan.Filter(type = FilterType.*ANNOTATION*,value = Controller.class)  
})  
@EnableAspectJAutoProxy  
*/\*\*  
 \* 核心配置文件  
 \*/*public class ApplicationConfig {  
 public ApplicationConfig() {  
 //System.out.println("ApplicationConfig");  
 }  
}

Config.java

package xyz.jfshare.car.system.config;  
  
public class Config {  
  
 public static final Integer *DRIVER\_NOT\_REGISTER* = 0;  
 public static final Integer *DRIVER\_NOT\_CHOOSE* = 1;  
 public static final Integer *DRIVER\_NOT\_REVIEW* = 2;  
 public static final Integer *DRIVER\_NORMAL* = 3;  
 public static final Integer *DRIVER\_BUSY* = 4;  
 public static final Integer *DRIVER\_BANNED* = 5;  
  
  
 public static final Integer *CAR\_NOT\_USE* = 0;  
 public static final Integer *CAR\_USING* = 1;  
  
 public static final Integer *COMPLAIN\_NOT\_REVIEW* = 0;  
 public static final Integer *COMPLAIN\_REVIEWED* = 1;  
  
 public static final Integer *COMPANY\_DELETE* = 3;  
 public static final Integer *COMPANY\_CHOSEN* = 2;  
 public static final Integer *COMPANY\_FREE* = 1;  
  
  
 public static final Integer *ORDER\_NOT\_PAY* = 1;  
 public static final Integer *ORDER\_NOT\_REVIEW* = 2;  
 public static final Integer *ORDER\_REVIEW\_SUCCESS* = 3;  
 public static final Integer *ORDER\_REVIEW\_FAILED* = 4;  
 public static final Integer *ORDER\_GOING* = 5;  
 public static final Integer *ORDER\_DONE* = 6;  
 public static final Integer *ORDER\_COMPLAINED* = 7;  
  
  
}

MybaticsplusConfig

package xyz.jfshare.car.system.config;  
  
import com.alibaba.druid.pool.DruidDataSource;  
import com.baomidou.mybatisplus.extension.plugins.PaginationInterceptor;  
import com.baomidou.mybatisplus.extension.spring.MybatisSqlSessionFactoryBean;  
import org.mybatis.spring.annotation.MapperScan;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.core.io.support.PathMatchingResourcePatternResolver;  
  
import javax.sql.DataSource;  
import java.io.IOException;  
  
@Configuration  
@MapperScan(value = "xyz.jfshare.car.system.dao")  
public class MybatisPlusConfig {  
  
 */\*\*mapper.xml文件路径，必须与其他SqlSessionFactory-mapper路径区分.\*/* public final static String *MAPPER\_XML\_PATH* = "classpath:mapper/\*.xml";  
  
 */\*\*  
 \* 配置数据源  
 \*/* @Bean  
 public DataSource dataSource(PropertyConfig propertyConfig){  
 //System.out.println("dataSource");  
 DruidDataSource dataSource = new DruidDataSource();  
 dataSource.setUsername(propertyConfig.getUser());  
 dataSource.setPassword(propertyConfig.getPassword());  
 dataSource.setUrl(propertyConfig.getUrl());  
 dataSource.setDriverClassName(propertyConfig.getDriver());  
 dataSource.setMaxActive(propertyConfig.getMaxActive());  
 dataSource.setInitialSize(propertyConfig.getInitialSize());  
 dataSource.setMaxWait(propertyConfig.getMaxWait());  
 dataSource.setMinIdle(propertyConfig.getMinIdle());  
 dataSource.setTimeBetweenEvictionRunsMillis(propertyConfig.getTimeBetweenEvictionRunsMillis());  
 dataSource.setMinEvictableIdleTimeMillis(propertyConfig.getMinEvictableIdleTimeMillis());  
 dataSource.setValidationQuery(propertyConfig.getValidationQuery());  
 dataSource.setTestWhileIdle(propertyConfig.isTestWhileIdle());  
 dataSource.setTestOnBorrow(propertyConfig.isTestOnBorrow());  
 dataSource.setTestOnReturn(propertyConfig.isTestOnReturn());  
 dataSource.setMaxOpenPreparedStatements(propertyConfig.getMaxOpenPreparedStatements());  
 dataSource.setRemoveAbandoned(propertyConfig.isRemoveAbandoned());  
 dataSource.setRemoveAbandonedTimeout(propertyConfig.getRemoveAbandonedTimeout());  
 dataSource.setLogAbandoned(propertyConfig.isLogAbandoned());  
 return dataSource;  
 }  
  
 */\*\*  
 \* 获得getMybatisSqlSessionFactoryBean  
 \** ***@return*** *\*/* @Bean(name = "sqlSessionFactory")  
 public MybatisSqlSessionFactoryBean sqlSessionFactory(DataSource dataSource) throws IOException {  
 //System.out.println("sqlSessionFactory");  
 MybatisSqlSessionFactoryBean mybatisPlus = new MybatisSqlSessionFactoryBean();  
 mybatisPlus.setDataSource(dataSource);  
 mybatisPlus.setTypeAliasesPackage("xyz.jfshare.car.system.entity");  
 mybatisPlus.setMapperLocations(new PathMatchingResourcePatternResolver().getResources(*MAPPER\_XML\_PATH*));  
 return mybatisPlus;  
 }  
  
  
  
  
 */\*\*  
 \* 设置mapper文件的扫描路径  
 \** ***@return*** *\*/*// @Bean  
// public MapperScannerConfigurer getMapperScannerConfigurer(){  
// MapperScannerConfigurer mapperScannerConfigurer = new MapperScannerConfigurer();  
// mapperScannerConfigurer.setBasePackage("com.tust.task.student.system.dao");  
// mapperScannerConfigurer.setSqlSessionFactoryBeanName("sqlSessionFactory");  
// return mapperScannerConfigurer;  
// }  
  
 */\*\*  
 \* 分页插件  
 \** ***@return*** *\*/* @Bean  
 public PaginationInterceptor getPaginationInterceptor(){  
 //System.out.println("getPaginationInterceptor");  
 PaginationInterceptor paginationInterceptor = new PaginationInterceptor();  
 //自定义方言 可以没有  
 paginationInterceptor.setDialectType("mysql");  
 return paginationInterceptor;  
 }  
  
}

PropertyConfig

package xyz.jfshare.car.system.config;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.ConfigurableApplicationContext;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.PropertySource;  
@PropertySource("classpath:jdbc.properties")  
@Configuration  
public class PropertyConfig {  
 private ConfigurableApplicationContext applicationContext;  
 @Autowired  
 public PropertyConfig(ConfigurableApplicationContext applicationContext) {  
 this.applicationContext = applicationContext;  
 this.url = applicationContext.getEnvironment().getProperty("spring.datasource.url");  
 this.driver = applicationContext.getEnvironment().getProperty("spring.datasource.driverClassName");  
 this.user = applicationContext.getEnvironment().getProperty("spring.datasource.username");  
 this.password = applicationContext.getEnvironment().getProperty("spring.datasource.password");  
 this.maxActive = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.maxActive"));  
 this.initialSize = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.initialSize"));  
 this.maxWait = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.maxWait"));  
 this.minIdle = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.minIdle"));  
 this.timeBetweenEvictionRunsMillis = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.timeBetweenEvictionRunsMillis"));  
 this.minEvictableIdleTimeMillis = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.minEvictableIdleTimeMillis"));  
 this.validationQuery = applicationContext.getEnvironment().getProperty("spring.datasource.validationQuery");  
 this.testWhileIdle = Boolean.*parseBoolean*(applicationContext.getEnvironment().getProperty("spring.datasource.testWhileIdle"));  
 this.testOnBorrow = Boolean.*parseBoolean*(applicationContext.getEnvironment().getProperty("spring.datasource.testOnBorrow"));  
 this.testOnReturn = Boolean.*parseBoolean*(applicationContext.getEnvironment().getProperty("spring.datasource.testOnReturn"));  
 this.maxOpenPreparedStatements = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.maxOpenPreparedStatements"));  
 this.removeAbandoned = Boolean.*parseBoolean*(applicationContext.getEnvironment().getProperty("spring.datasource.removeAbandoned"));  
 this.removeAbandonedTimeout = Integer.*parseInt*(applicationContext.getEnvironment().getProperty("spring.datasource.removeAbandonedTimeout"));  
 this.logAbandoned = Boolean.*parseBoolean*(applicationContext.getEnvironment().getProperty("spring.datasource.logAbandoned"));  
 }  
  
 private String url;  
 private String driver;  
 private String user;  
 private String password;  
 private int maxActive;  
 private int initialSize;  
 private int maxWait;  
 private int minIdle;  
 private int timeBetweenEvictionRunsMillis;  
 private int minEvictableIdleTimeMillis;  
 private String validationQuery;  
 private boolean testWhileIdle;  
 private boolean testOnBorrow;  
 private boolean testOnReturn;  
 private int maxOpenPreparedStatements;  
 private boolean removeAbandoned;  
 private int removeAbandonedTimeout;  
 private boolean logAbandoned;  
  
 public ConfigurableApplicationContext getApplicationContext() {  
 return applicationContext;  
 }  
  
 public void setApplicationContext(ConfigurableApplicationContext applicationContext) {  
 this.applicationContext = applicationContext;  
 }  
  
 public String getUrl() {  
 return url;  
 }  
  
 public void setUrl(String url) {  
 this.url = url;  
 }  
  
 public String getDriver() {  
 return driver;  
 }  
  
 public void setDriver(String driver) {  
 this.driver = driver;  
 }  
  
 public String getUser() {  
 return user;  
 }  
  
 public void setUser(String user) {  
 this.user = user;  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password;  
 }  
  
 public int getMaxActive() {  
 return maxActive;  
 }  
  
 public void setMaxActive(int maxActive) {  
 this.maxActive = maxActive;  
 }  
  
 public int getInitialSize() {  
 return initialSize;  
 }  
  
 public void setInitialSize(int initialSize) {  
 this.initialSize = initialSize;  
 }  
  
 public int getMaxWait() {  
 return maxWait;  
 }  
  
 public void setMaxWait(int maxWait) {  
 this.maxWait = maxWait;  
 }  
  
 public int getMinIdle() {  
 return minIdle;  
 }  
  
 public void setMinIdle(int minIdle) {  
 this.minIdle = minIdle;  
 }  
  
 public int getTimeBetweenEvictionRunsMillis() {  
 return timeBetweenEvictionRunsMillis;  
 }  
  
 public void setTimeBetweenEvictionRunsMillis(int timeBetweenEvictionRunsMillis) {  
 this.timeBetweenEvictionRunsMillis = timeBetweenEvictionRunsMillis;  
 }  
  
 public int getMinEvictableIdleTimeMillis() {  
 return minEvictableIdleTimeMillis;  
 }  
  
 public void setMinEvictableIdleTimeMillis(int minEvictableIdleTimeMillis) {  
 this.minEvictableIdleTimeMillis = minEvictableIdleTimeMillis;  
 }  
  
 public String getValidationQuery() {  
 return validationQuery;  
 }  
  
 public void setValidationQuery(String validationQuery) {  
 this.validationQuery = validationQuery;  
 }  
  
 public boolean isTestWhileIdle() {  
 return testWhileIdle;  
 }  
  
 public void setTestWhileIdle(boolean testWhileIdle) {  
 this.testWhileIdle = testWhileIdle;  
 }  
  
 public boolean isTestOnBorrow() {  
 return testOnBorrow;  
 }  
  
 public void setTestOnBorrow(boolean testOnBorrow) {  
 this.testOnBorrow = testOnBorrow;  
 }  
  
 public boolean isTestOnReturn() {  
 return testOnReturn;  
 }  
  
 public void setTestOnReturn(boolean testOnReturn) {  
 this.testOnReturn = testOnReturn;  
 }  
  
 public int getMaxOpenPreparedStatements() {  
 return maxOpenPreparedStatements;  
 }  
  
 public void setMaxOpenPreparedStatements(int maxOpenPreparedStatements) {  
 this.maxOpenPreparedStatements = maxOpenPreparedStatements;  
 }  
  
 public boolean isRemoveAbandoned() {  
 return removeAbandoned;  
 }  
  
 public void setRemoveAbandoned(boolean removeAbandoned) {  
 this.removeAbandoned = removeAbandoned;  
 }  
  
 public int getRemoveAbandonedTimeout() {  
 return removeAbandonedTimeout;  
 }  
  
 public void setRemoveAbandonedTimeout(int removeAbandonedTimeout) {  
 this.removeAbandonedTimeout = removeAbandonedTimeout;  
 }  
  
 public boolean isLogAbandoned() {  
 return logAbandoned;  
 }  
  
 public void setLogAbandoned(boolean logAbandoned) {  
 this.logAbandoned = logAbandoned;  
 }  
  
 @Override  
 public String toString() {  
 return "PropertyConfig{" +  
 "applicationContext=" + applicationContext +  
 ", url='" + url + '\'' +  
 ", driver='" + driver + '\'' +  
 ", user='" + user + '\'' +  
 ", password='" + password + '\'' +  
 ", maxActive=" + maxActive +  
 ", initialSize=" + initialSize +  
 ", maxWait=" + maxWait +  
 ", minIdle=" + minIdle +  
 ", timeBetweenEvictionRunsMillis=" + timeBetweenEvictionRunsMillis +  
 ", minEvictableIdleTimeMillis=" + minEvictableIdleTimeMillis +  
 ", validationQuery='" + validationQuery + '\'' +  
 ", testWhileIdle=" + testWhileIdle +  
 ", testOnBorrow=" + testOnBorrow +  
 ", testOnReturn=" + testOnReturn +  
 ", maxOpenPreparedStatements=" + maxOpenPreparedStatements +  
 ", removeAbandoned=" + removeAbandoned +  
 ", removeAbandonedTimeout=" + removeAbandonedTimeout +  
 ", logAbandoned=" + logAbandoned +  
 '}';  
 }  
}

ShiroConfig

package xyz.jfshare.car.system.config;  
import lombok.extern.slf4j.Slf4j;  
import org.apache.shiro.authc.credential.HashedCredentialsMatcher;  
import org.apache.shiro.spring.LifecycleBeanPostProcessor;  
import org.apache.shiro.spring.security.interceptor.AuthorizationAttributeSourceAdvisor;  
import org.apache.shiro.spring.web.ShiroFilterFactoryBean;  
import org.apache.shiro.web.mgt.DefaultWebSecurityManager;  
import org.springframework.aop.framework.autoproxy.DefaultAdvisorAutoProxyCreator;  
import org.springframework.beans.factory.annotation.Qualifier;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.DependsOn;  
import xyz.jfshare.car.system.shiro.realm.MyRealm;  
  
import java.util.HashMap;  
import java.util.Map;  
  
@Configuration  
@Slf4j  
public class ShiroConfig {  
  
 @Bean(name = "lifecycleBeanPostProcessor")  
 public static LifecycleBeanPostProcessor getLifecycleBeanPostProcessor() {  
 //System.out.println("lifecycleBeanPostProcessor");  
 return new LifecycleBeanPostProcessor();  
 }  
  
 @Bean  
 @DependsOn({"lifecycleBeanPostProcessor"})  
 public DefaultAdvisorAutoProxyCreator advisorAutoProxyCreator(){  
 //System.out.println("lifecycleBeanPostProcessor");  
 DefaultAdvisorAutoProxyCreator advisorAutoProxyCreator = new DefaultAdvisorAutoProxyCreator();  
 advisorAutoProxyCreator.setProxyTargetClass(true);  
 return advisorAutoProxyCreator;  
 }  
 @Bean  
 public AuthorizationAttributeSourceAdvisor authorizationAttributeSourceAdvisor(@Qualifier("theSecurityManager") DefaultWebSecurityManager securityManager){  
 //System.out.println("authorizationAttributeSourceAdvisor");  
 AuthorizationAttributeSourceAdvisor authorizationAttributeSourceAdvisor = new AuthorizationAttributeSourceAdvisor();  
 authorizationAttributeSourceAdvisor.setSecurityManager(securityManager);  
 return authorizationAttributeSourceAdvisor;  
 }  
  
 @Bean(name = "shiroFilter")  
 public ShiroFilterFactoryBean getShiroFilterFactoryBean(@Qualifier("theSecurityManager") DefaultWebSecurityManager securityManager){  
 //System.out.println("shiroFilter");  
 ShiroFilterFactoryBean shiroFilterFactoryBean = new ShiroFilterFactoryBean();  
 //设置安全管理器  
 shiroFilterFactoryBean.setSecurityManager(securityManager);  
  
 //未授权界面;  
// shiroFilterFactoryBean.setUnauthorizedUrl("/unauthorized");  
  
 Map<String, String> filterChain = new HashMap<>();  
 filterChain.put("/\*\*.js", "anon");  
 filterChain.put("/druid/\*\*", "anon");  
 filterChain.put("/403", "anon");  
 filterChain.put("/auth/\*\*","anon");  
 filterChain.put("/login","anon");  
 filterChain.put("/static/\*\*","anon");  
 filterChain.put("/register","anon");  
 filterChain.put("/templates/auth/\*\*","anon");  
 filterChain.put("/\*\*","authc");  
 shiroFilterFactoryBean.setFilterChainDefinitionMap(filterChain);  
 shiroFilterFactoryBean.setLoginUrl("/login");  
 return shiroFilterFactoryBean;  
 }  
  
 @Bean(name = "theSecurityManager")  
 public DefaultWebSecurityManager getSecurityManager(@Qualifier("realm") MyRealm realm){  
 //System.out.println("theSecurityManager");  
 DefaultWebSecurityManager securityManager = new DefaultWebSecurityManager();  
 //关联realm  
 securityManager.setRealm(realm);  
 return securityManager;  
 }  
  
 @Bean("realm")  
 public MyRealm getRealm(){  
 //System.out.println("realm");  
 MyRealm realm = new MyRealm();  
  
// realm.setCredentialsMatcher(getHashedCredentialsMatcher());  
 return realm;  
 }  
  
 @Bean  
 public HashedCredentialsMatcher getHashedCredentialsMatcher(){  
 HashedCredentialsMatcher credentialsMatcher = new HashedCredentialsMatcher();  
 credentialsMatcher.setHashAlgorithmName("MD5");  
 credentialsMatcher.setHashIterations(1);  
 return credentialsMatcher;  
 }  
}

WebConfig

package xyz.jfshare.car.system.config;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.ComponentScan;  
import org.springframework.context.annotation.FilterType;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.servlet.ViewResolver;  
import org.springframework.web.servlet.config.annotation.EnableWebMvc;  
import org.springframework.web.servlet.config.annotation.ResourceHandlerRegistry;  
import org.springframework.web.servlet.config.annotation.WebMvcConfigurer;  
import org.springframework.web.servlet.handler.SimpleMappingExceptionResolver;  
import org.springframework.web.servlet.view.InternalResourceViewResolver;  
import xyz.jfshare.car.system.resolver.GlobalExceptionResolver;  
import java.util.Properties;  
  
//@Configuration  
@ComponentScan(basePackages = "xyz.jfshare.car.system",  
 includeFilters = {@ComponentScan.Filter(type = FilterType.*ANNOTATION*,value = Controller.class)},  
 useDefaultFilters = false  
// excludeFilters = {@ComponentScan.Filter(type = FilterType.ANNOTATION,value = Configuration.class)}  
)  
@EnableWebMvc  
*/\*\*  
 \* springmvc配置文件  
 \*/*public class WebConfig implements WebMvcConfigurer {  
 @Bean  
 public SimpleMappingExceptionResolver exceptionResolver() {  
 //System.out.println("exceptionResolver");  
 SimpleMappingExceptionResolver resolver = new SimpleMappingExceptionResolver();  
 Properties mappings = new Properties();  
 mappings.setProperty("org.apache.shiro.authz.UnauthorizedException", "/unauthorized");  
 mappings.setProperty("org.apache.shiro.authz.UnauthenticatedException", "/unauthenticated");  
 resolver.setExceptionMappings(mappings);  
  
 return resolver;  
 }  
  
 @Bean  
 public ViewResolver viewResolver() {  
 //System.out.println("viewResolver");  
 InternalResourceViewResolver resolver = new InternalResourceViewResolver();  
 resolver.setPrefix("/WEB-INF/templates/");  
 resolver.setSuffix(".jsp");  
 resolver.setExposeContextBeansAsAttributes(true);  
 return resolver;  
 }  
  
 @Override  
 public void addResourceHandlers(ResourceHandlerRegistry registry) {  
 //System.out.println("addResourceHandlers");  
 registry.addResourceHandler("/static/\*\*")//过滤静态资源  
 .addResourceLocations("/static/","/favicon.ico");  
 }  
  
 @Bean("handlerExceptionResolver")  
 public GlobalExceptionResolver handlerExceptionResolver(){  
 //System.out.println("handlerExceptionResolver");  
 return new GlobalExceptionResolver();  
 }  
}

WebInit

package xyz.jfshare.car.system.config;  
import org.springframework.web.context.WebApplicationContext;  
import org.springframework.web.filter.CharacterEncodingFilter;  
import org.springframework.web.filter.DelegatingFilterProxy;  
import org.springframework.web.servlet.DispatcherServlet;  
import org.springframework.web.servlet.FrameworkServlet;  
import org.springframework.web.servlet.HandlerExecutionChain;  
import org.springframework.web.servlet.HandlerMapping;  
import org.springframework.web.servlet.resource.DefaultServletHttpRequestHandler;  
import org.springframework.web.servlet.support.AbstractAnnotationConfigDispatcherServletInitializer;  
  
import javax.servlet.DispatcherType;  
import javax.servlet.FilterRegistration.Dynamic;  
import javax.servlet.ServletContext;  
import javax.servlet.ServletException;  
import javax.servlet.ServletRegistration;  
import javax.servlet.http.HttpServletRequest;  
import java.util.EnumSet;  
  
  
*/\*\*  
 \* web.xml  
 \* 继承AbstractAnnotationConfigDispatcherServletInitializer，会同时创建DispatcherServlet和ContextLoaderListener  
 \*/*public class WebInit extends AbstractAnnotationConfigDispatcherServletInitializer {  
 //将bean加入容器  
 @Override  
 protected Class<?>[] getRootConfigClasses() {  
 //这里可以配置需要加入容器的Bean,同样可以声明配置类,然后加Bean  
 return new Class[]{ApplicationConfig.class};  
 }  
  
 //这个也是相当于将Bean加入容器  
 //相当于springmvc容器  
 // url映射配置,返回spring的配置文件  
 //这里WebConfig主要是配置DispatcherSerlvet,视频解析器,JSON等  
 @Override  
 protected Class<?>[] getServletConfigClasses() {  
 return new Class[]{WebConfig.class};  
 }  
 //拦截请求匹配,只拦截/  
 @Override  
 protected String[] getServletMappings() {  
 return new String[]{"/"};  
 }  
  
 @Override  
 public void onStartup(ServletContext servletContext) throws ServletException {  
 super.onStartup(servletContext);  
  
 Dynamic characterEncoding = servletContext.addFilter("characterEncoding", CharacterEncodingFilter.class);  
 characterEncoding.setInitParameter("forceEncoding", "true");  
 characterEncoding.setInitParameter("encoding", "UTF-8");  
 characterEncoding.addMappingForUrlPatterns(null, true, "/\*");  
  
 EnumSet<DispatcherType> enumSet = EnumSet.*of*(DispatcherType.*INCLUDE*,DispatcherType.*REQUEST*,DispatcherType.*FORWARD*,DispatcherType.*ERROR*);  
 Dynamic shiroFilter = servletContext.addFilter("shiroFilter", DelegatingFilterProxy.class);  
 shiroFilter.setInitParameter("targetFilterLifecycle", "true"); // 设置true由servlet容器控制filter的生命周期  
 shiroFilter.addMappingForUrlPatterns(enumSet, true, "/\*");  
 // 静态资源处理  
// servletContext.addFilter("resourceUrlEncoding", ResourceUrlEncodingFilter.class).addMappingForUrlPatterns(null, true, "/\*");  
 }  
  
 @Override  
 protected void customizeRegistration(ServletRegistration.Dynamic registration) {  
 registration.setInitParameter("throwExceptionIfNoHandlerFo‌​und","true");  
 }  
  
  
 @Override  
 protected FrameworkServlet createDispatcherServlet(WebApplicationContext servletAppContext) {  
 final DispatcherServlet dispatcherServlet = new DispatcherServlet(){  
 @Override  
 protected HandlerExecutionChain getHandler(HttpServletRequest request) throws Exception {  
 if (this.getHandlerMappings() != null) {  
 for (HandlerMapping hm : this.getHandlerMappings()) {  
 if (logger.isTraceEnabled()) {  
 logger.trace(  
 "Testing handler map [" + hm + "] in DispatcherServlet with name '" + getServletName() + "'");  
 }  
 HandlerExecutionChain handler = hm.getHandler(request);  
 if (handler != null&& !(handler.getHandler() instanceof DefaultServletHttpRequestHandler)) {  
 return handler;  
 }  
 }  
 }  
 return null;  
 }  
 {  
 setApplicationContext(servletAppContext);  
 }  
 };  
 dispatcherServlet.setThrowExceptionIfNoHandlerFound(true);  
 return dispatcherServlet;  
 }  
}

审核车辆！！！！！！！

driverConcroller

@RequestMapping("/chooseCar/list")!!!一开始到这里！！！！  
public String chooseCar(Model model){  
 User user = (User) SecurityUtils.*getSubject*().getPrincipal();  
 BusDriver driver = driverService.getOne(  
 new QueryWrapper<BusDriver>()  
 .eq("user\_id",user.getId())  
 );  
 if (driver == null){  
 model.addAttribute("msg","请先完善信息");  
 return "redirect:/user/info";  
 }else {  
 if (driver.getCarId() == 0){  
 return "redirect:/car/list";  
 }else {  
 return "redirect:/car/info";  
 }  
 }  
}

UserController

@RequestMapping("info")  
public String userInfo(Model model,HttpSession session){  
 User user = (User) SecurityUtils.*getSubject*().getPrincipal();  
 if(user.getRoleId() == 2) {//使用者为用车公司时  
 BusCompany company = companyService.getByUserId(user.getId());  
 if (company != null){  
 List<BusOrder> orderList = orderService.getByCompanyId(company.getId());  
 List<BusComplain> complainList = complainService.listByCompanyId(company.getId());  
 session.setAttribute("company", company);  
 session.setAttribute("complainList", complainList);  
 session.setAttribute("orderList", orderList);  
 return "redirect:/companyInfo";  
 }else {  
 List<BusCompany> companyList = companyService.getAll();  
 model.addAttribute("companyList",companyList);  
 return "company/companyChoose";  
 }  
 } else if(user.getRoleId() == 3) {//使用者为司机时  
 BusDriver driver = driverService.getOne(  
 new QueryWrapper<BusDriver>() 条件构造器Entity 对象封装操作类，  
 .eq("user\_id",user.getId())eq等于 user\_id=user.getId()

根据 entity 条件，获取QueryWrapper实体对象封装操作类（可以为 null）下方获取到queryWrapper后查询条件为user\_id字段为int的ID );  
 if (driver != null){  
 BusCar car = carService.getById(driver.getCarId());  
 List<BusOrder> orderList = orderService.getByDriverId(driver.getId());  
 List<BusComplain> complainList = complainService.listByDriverId(driver.getId());  
 session.setAttribute("car", car);  
 session.setAttribute("driver", driver);  
 session.setAttribute("complainList", complainList);  
 session.setAttribute("orderList", orderList);  
  
 return "redirect:/driverInfo";  
 }else {  
 List<BusDriver> driverList = driverService.getAll();  
 model.addAttribute("driverList",driverList);  
 return "driver/driverChoose";  
 }  
 }else {  
 return "redirect:/main";  
 }  
}

driverList.jsp

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<%@**taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<%@**include** file="../common/head.jsp"%>  
<%@**include** file="../common/topbar.jsp"%>  
<%@**include** file="../common/sidebar.jsp"%>  
<div class="content-wrapper">  
 <section class="content-header">  
 <h1>  
 个人主页(请先完善信息)  
 </h1>  
 </section>  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
  
 <!-- Main content -->  
 <section class="content">  
  
 <div class="row">  
 <div class="col-md-3">  
  
 <!-- Profile Image -->  
 <div class="box box-primary">  
 <div class="box-body box-profile">  
 <img class="profile-user-img img-responsive img-circle" src="<**shiro:principal** property="icon"/>" alt="User profile picture">  
  
 <h3 class="profile-username text-center"><**shiro:principal** property="nickname"/></h3>  
  
 <p class="text-muted text-center"><**shiro:principal** property="roleName"/></p>  
  
 <ul class="list-group list-group-unbordered">  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">用户名:&nbsp;&nbsp;</b> <div class="pull-left"><**shiro:principal** property="username"/></div>  
 </li>  
 </ul>  
 </div>  
 <!-- /.box-body -->  
 </div>  
 <!-- /.box -->  
  
 </div>  
 <!-- /.col -->  
 <div class="col-md-9">  
 <div class="nav-tabs-custom">  
 <ul class="nav nav-tabs">  
 <li class="active"><a href="#choose" data-toggle="tab">选择已有司机信息</a></li> #是链接到本页面  
 <li><a href="#add" data-toggle="tab">新建个人信息</a></li>  
 </ul>  
 <div class="tab-content">  
 <div class="active tab-pane" id="choose">  
 <table class="table table-bordered">  
 <tr>  
 <th style="width: 80px;text-align:center">序号</th>  
 <th style="text-align:center">姓名</th>  
 <th style="text-align:center">性别</th>  
 <th style="text-align:center">年龄</th>  
 <th style="text-align:center">电话</th>  
 <th style="text-align:center">邮箱</th>  
 <th style="text-align:center">状态</th>  
 <th style="text-align:center">操作</th>  
 </tr>  
  
 <**c:forEach** items="**${**driverList**}**" var="driver" varStatus="itemStatus">  
 <tr style="text-align:center">  
 <td><**c:out** value="**${**itemStatus.count**}**"/> </td>  
 <td><**c:out** value="**${**driver.name**}**"/> </td>  
 <td><**c:out** value="**${**driver.gender==0?'男':'女'**}**"/> </td>  
 <td><**c:out** value="**${**driver.age**}**"/> </td>  
 <td><**c:out** value="**${**driver.phone**}**"/> </td>  
 <td><**c:out** value="**${**driver.email**}**"/> </td>  
 <td><**c:out** value="**${**driver.status == 0?'可选':'已选'**}**"/> </td>  
 <**c:if** test="**${**driver.status == 0**}**">  
 <td><a href="/driver/addUser?id=**${**driver.id**}**&userId=<**shiro:principal** property="id"/>">选择</a></td>  
 </**c:if**>  
 </tr>  
 </**c:forEach**>  
 </table>  
 </div>  
 <!-- /.tab-pane -->  
 <div class="tab-pane" id="add">  
 <div class="col-sm-6">  
 <img class="img-responsive" src="<**shiro:principal** property="icon"/>" alt="Photo">  
 </div>  
 <form class="form-horizontal" action="/driver/selfAdd" method="post">  
 <div class="row">  
 <div class="col-md-6">  
 <div class="box box-primary">  
 <div class="box-body">  
 <div class="form-group">  
 <div style="display: none">  
 <input name="userId" value="<**shiro:principal** property="id"/>" type="text" id="inputUserId">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputName" class="col-sm-2 control-label">姓名:</label>  
 <div class="col-sm-10">  
 <input name="name" type="text" class="form-control" id="inputName" placeholder="输入姓名">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputName" class="col-sm-2 control-label">性别:</label>  
 <div class="col-sm-10">  
 <select name="gender" class="form-control" id="inputGender">  
 <option value="0">男</option>  
 <option value="1">女</option>  
 </select>  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputAge" class="col-sm-2 control-label">年龄:</label>  
 <div class="col-sm-10">  
 <input name="age" type="text" class="form-control" id="inputAge" placeholder="输入年龄">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputPhone" class="col-sm-2 control-label">电话:</label>  
 <div class="col-sm-10">  
 <input name="phone" type="text" class="form-control" id="inputPhone" placeholder="输入电话">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputEmail" class="col-sm-2 control-label">邮箱:</label>  
 <div class="col-sm-10">  
 <input name="email" type="text" class="form-control" id="inputEmail" placeholder="输入邮箱">  
 </div>  
 </div>  
 <button type="reset" class="btn btn-default">重置</button>  
 <button type="submit" class="btn btn-info pull-right">提交</button>  
 </div>  
 </div>  
 </div>  
 </div>  
 <!-- /.box-body -->  
 </form>  
 </div>  
 <!-- /.tab-pane -->  
 </div>  
 <!-- /.tab-content -->  
 </div>  
 <!-- /.nav-tabs-custom -->  
 </div>  
 <!-- /.col -->  
 </div>  
 <!-- /.row -->  
  
 </section>  
 <!-- /.content -->  
  
</div>  
<script>  
  
 **function** sendGender(gender){  
 document.getElementById("inputGender").value=gender;  
 }  
</script>  
<%@**include** file="../common/bottom.jsp"%>

DriverInfo.jsp

<style>  
 .text-muted{  
 margin: 10px 10px 5px;  
 }  
</style>  
<div class="content-wrapper">  
 <section class="content-header">  
 <h1>  
 个人主页  
 </h1>  
 </section>  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
  
 <!-- Main content -->  
 <section class="content">  
  
 <div class="row">  
 <div class="col-md-3">  
  
 <!-- Profile Image -->  
 <div class="box box-primary">  
 <div class="box-body box-profile">  
 <img class="profile-user-img img-responsive img-circle" src="<**shiro:principal** property="icon"/>" alt="User profile picture">  
  
 <h3 class="profile-username text-center"><**shiro:principal** property="nickname"/></h3>  
  
 <p class="text-muted text-center"><**shiro:principal** property="roleName"/></p>  
  
 <ul class="list-group list-group-unbordered">  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">用户名:&nbsp;&nbsp;</b> <div class="pull-left"><**shiro:principal** property="username"/></div>  
 </li>  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">姓名:&nbsp;&nbsp;</b> <div class="pull-left">**${**sessionScope.get("driver").name**}**</div>  
 </li>  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">性别:&nbsp;&nbsp;</b> <div class="pull-left">**${**sessionScope.driver.gender==0?'男':'女'**}**</div>  
 </li>  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">年龄:&nbsp;&nbsp;</b> <div class="pull-left">**${**sessionScope.driver.age**}**</div>  
 </li>  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">电话:&nbsp;&nbsp;</b> <div class="pull-left">**${**sessionScope.driver.phone**}**</div>  
 </li>  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">邮箱:&nbsp;&nbsp;</b> <div class="pull-left">**${**sessionScope.driver.email**}**</div>  
 </li>  
 </ul>  
 </div>  
 <!-- /.box-body -->  
 </div>  
 <!-- /.box -->  
  
 <!-- About Me Box -->  
 <div class="box box-primary">  
 <div class="box-header with-border">  
 <**c:if** test="**${**sessionScope.driver.status == 2**}**">  
 <h3 class="box-title">车辆信息(待审核)</h3>  
 </**c:if**>  
 <**c:if** test="**${**sessionScope.driver.status == 3**}**">  
 <h3 class="box-title">车辆信息</h3>  
 </**c:if**>  
 </div>  
 <!-- /.box-header -->  
 <div class="box-body">  
 <**c:if** test="**${**sessionScope.car == **null}**">  
 暂无车辆  
 </**c:if**>  
 <**c:if** test="**${**sessionScope.car != **null}**">  
 <strong><i class="fa fa-book margin-r-5"></i> 车牌</strong>  
  
 <p class="text-muted">**${**sessionScope.car.number**}**</p>  
  
 <hr>  
  
 <strong><i class="fa fa-map-marker margin-r-5"></i> 颜色</strong>  
  
 <p class="text-muted">**${**sessionScope.car.color**}**</p>  
  
 <hr>  
  
 <strong><i class="fa fa-pencil margin-r-5"></i> 品牌</strong>  
  
 <p class="text-muted">**${**sessionScope.car.bands**}**</p>  
  
 <hr>  
  
 <strong><i class="fa fa-file-text-o margin-r-5"></i> 车型</strong>  
 <p class="text-muted">  
 <**c:if** test="**${**sessionScope.car.type == 1**}**">  
 四座  
 </**c:if**>  
 <**c:if** test="**${**sessionScope.car.type == 2**}**">  
 七座  
 </**c:if**>  
 <**c:if** test="**${**sessionScope.car.type == 3**}**">  
 其他  
 </**c:if**>  
 </p>  
 </**c:if**>  
 </div>  
 <!-- /.box-body -->  
 </div>

DriverChoose.jap

<%@**page** contentType="text/html;charset=UTF-8" language="java" pageEncoding="UTF-8" %>  
<%@**taglib** uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>  
<%@**include** file="../common/head.jsp"%>  
<%@**include** file="../common/topbar.jsp"%>  
<%@**include** file="../common/sidebar.jsp"%>  
<div class="content-wrapper">  
 <section class="content-header">  
 <h1>  
 个人主页(请先完善信息)  
 </h1>  
 </section>  
  
 <**c:if** test="**${**param.msg != **null}**">  
 <**c:choose**>  
 <**c:when** test="**${**param.msg.contains(\"成功\")**}**">  
 <div class="alert alert-success alert-dismissible" style="overflow: auto">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button>  
 <h4><i class="icon fa fa-check"></i> 成功</h4>  
 **${**param.msg**}** </div>  
 </**c:when**>  
 <**c:otherwise**>  
 <div class="alert alert-warning alert-dismissible">  
 <button type="button" class="close" data-dismiss="alert" aria-hidden="true">×</button>  
 <h4><i class="icon fa fa-warning"></i> 失败</h4>  
 **${**param.msg**}** </div>  
 </**c:otherwise**>  
 </**c:choose**>  
 </**c:if**>  
  
 <!-- Main content -->  
 <section class="content">  
  
 <div class="row">  
 <div class="col-md-3">  
  
 <!-- Profile Image -->  
 <div class="box box-primary">  
 <div class="box-body box-profile">  
 <img class="profile-user-img img-responsive img-circle" src="<**shiro:principal** property="icon"/>" alt="User profile picture">  
  
 <h3 class="profile-username text-center"><**shiro:principal** property="nickname"/></h3>  
  
 <p class="text-muted text-center"><**shiro:principal** property="roleName"/></p>  
  
 <ul class="list-group list-group-unbordered">  
 <li class="list-group-item" style="height: 38px">  
 <b class="pull-left" style="width: 100px;text-align: right">用户名:&nbsp;&nbsp;</b> <div class="pull-left"><**shiro:principal** property="username"/></div>  
 </li>  
 </ul>  
 </div>  
 <!-- /.box-body -->  
 </div>  
 <!-- /.box -->  
  
 </div>  
 <!-- /.col -->  
 <div class="col-md-9">  
 <div class="nav-tabs-custom">  
 <ul class="nav nav-tabs">  
 <li class="active"><a href="#choose" data-toggle="tab">选择已有司机信息</a></li>  
 <li><a href="#add" data-toggle="tab">新建个人信息</a></li>  
 </ul>  
 <div class="tab-content">  
 <div class="active tab-pane" id="choose">  
 <table class="table table-bordered">  
 <tr>  
 <th style="width: 80px;text-align:center">序号</th>  
 <th style="text-align:center">姓名</th>  
 <th style="text-align:center">性别</th>  
 <th style="text-align:center">年龄</th>  
 <th style="text-align:center">电话</th>  
 <th style="text-align:center">邮箱</th>  
 <th style="text-align:center">状态</th>  
 <th style="text-align:center">操作</th>  
 </tr>  
  
 <**c:forEach** items="**${**driverList**}**" var="driver" varStatus="itemStatus">  
 <tr style="text-align:center">  
 <td><**c:out** value="**${**itemStatus.count**}**"/> </td>  
 <td><**c:out** value="**${**driver.name**}**"/> </td>  
 <td><**c:out** value="**${**driver.gender==0?'男':'女'**}**"/> </td>  
 <td><**c:out** value="**${**driver.age**}**"/> </td>  
 <td><**c:out** value="**${**driver.phone**}**"/> </td>  
 <td><**c:out** value="**${**driver.email**}**"/> </td>  
 <td><**c:out** value="**${**driver.status == 0?'可选':'已选'**}**"/> </td>  
 <**c:if** test="**${**driver.status == 0**}**">  
 <td><a href="/driver/addUser?id=**${**driver.id**}**&userId=<**shiro:principal** property="id"/>">选择</a></td>  
 </**c:if**>  
 </tr>  
 </**c:forEach**>  
 </table>  
 </div>  
 <!-- /.tab-pane -->  
 <div class="tab-pane" id="add">  
 <div class="col-sm-6">  
 <img class="img-responsive" src="<**shiro:principal** property="icon"/>" alt="Photo">  
 </div>  
 <form class="form-horizontal" action="/driver/selfAdd" method="post">  
 <div class="row">  
 <div class="col-md-6">  
 <div class="box box-primary">  
 <div class="box-body">  
 <div class="form-group">  
 <div style="display: none">  
 <input name="userId" value="<**shiro:principal** property="id"/>" type="text" id="inputUserId">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputName" class="col-sm-2 control-label">姓名:</label>  
 <div class="col-sm-10">  
 <input name="name" type="text" class="form-control" id="inputName" placeholder="输入姓名">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputName" class="col-sm-2 control-label">性别:</label>  
 <div class="col-sm-10">  
 <select name="gender" class="form-control" id="inputGender">  
 <option value="0">男</option>  
 <option value="1">女</option>  
 </select>  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputAge" class="col-sm-2 control-label">年龄:</label>  
 <div class="col-sm-10">  
 <input name="age" type="text" class="form-control" id="inputAge" placeholder="输入年龄">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputPhone" class="col-sm-2 control-label">电话:</label>  
 <div class="col-sm-10">  
 <input name="phone" type="text" class="form-control" id="inputPhone" placeholder="输入电话">  
 </div>  
 </div>  
 <div class="form-group">  
 <label for="inputEmail" class="col-sm-2 control-label">邮箱:</label>  
 <div class="col-sm-10">  
 <input name="email" type="text" class="form-control" id="inputEmail" placeholder="输入邮箱">  
 </div>  
 </div>  
 <button type="reset" class="btn btn-default">重置</button>  
 <button type="submit" class="btn btn-info pull-right">提交</button>  
 </div>  
 </div>  
 </div>  
 </div>  
 <!-- /.box-body -->  
 </form>  
 </div>  
 <!-- /.tab-pane -->  
 </div>  
 <!-- /.tab-content -->  
 </div>  
 <!-- /.nav-tabs-custom -->  
 </div>  
 <!-- /.col -->  
 </div>  
 <!-- /.row -->  
  
 </section>  
 <!-- /.content -->  
  
</div>  
<script>  
  
 **function** sendGender(gender){  
 document.getElementById("inputGender").value=gender;  
 }  
</script>  
<%@**include** file="../common/bottom.jsp"%>

用车公司的！！！

BusCompanyController.java

用车公司：package xyz.jfshare.car.system.controller;  
  
import javax.servlet.http.HttpSession;  
import java.util.ArrayList;  
import java.util.List;  
import java.util.Objects;  
  
@Controller  
@RequestMapping("/company/")  
public class BusCompanyController {  
  
 @Autowired  
 private BusCompanyService companyService;  
 @Autowired  
 private BusOrderService orderService;  
  
 @RequestMapping("list")  
 public String list(@RequestParam(required = false,defaultValue = "")String name,  
 @RequestParam(required = false,defaultValue = "")String leader,  
 @RequestParam(required = false,defaultValue = "1") Integer page,  
 @RequestParam(required = false,defaultValue = "10")Integer pageSize,  
 Model model){  
 SecurityUtils.*getSubject*().checkPermission("/company/list");  
 SecurityUtils.*getSubject*().checkRole("admin");  
 // 筛选条件  
 BusCompany company = new BusCompany();  
 if (StringUtils.isNotBlank(name)){  
 company.setName(name);  
 }  
 if (StringUtils.isNotBlank(leader)){  
 company.setLeader(leader);  
 }  
 if(!Objects.equals(0,pageSize)){  
 company.setSkip((page-1)\*pageSize);  
 company.setSize(pageSize);  
 }  
  
 List<BusCompany> list = companyService.listCompany(company);  
 Integer total = companyService.countCompany(company);  
  
 int pages = total/pageSize + (total % pageSize==0?0:1);  
  
 model.addAttribute("dataList",list);  
 model.addAttribute("total",total);  
 model.addAttribute("pages",pages);  
  
 return "company/companyList";  
 }  
  
 @RequestMapping("add")  
 public String add(String name, Integer userId, String leader, String email, RedirectAttributes attributes){  
 SecurityUtils.getSubject().checkPermission("/company/add");  
 BusCompany company = new BusCompany();  
 company.setName(name);  
 if(userId!=null){  
 company.setUserId(userId);  
 }  
 company.setLeader(leader);  
 company.setEmail(email);  
 boolean res = companyService.save(company);  
 if (!res){  
 attributes.addAttribute("msg","添加失败,请联系管理员");  
 return "redirect:list";  
 }  
 attributes.addAttribute("msg","添加成功");  
 return "redirect:list";  
 }  
  
 @RequestMapping("selfAdd")  
 public String stuAdd(String name, String leader, String email, RedirectAttributes attributes, HttpSession session){  
 SecurityUtils.getSubject().checkPermission("/company/selfAdd");  
 SecurityUtils.getSubject().checkRole("company");  
 Integer userId = ((User) SecurityUtils.getSubject().getPrincipal()).getId();  
 BusCompany company = new BusCompany();  
 company.setName(name);  
 if(userId!=null){  
 company.setUserId(userId);  
 }  
 company.setLeader(leader);  
 company.setStatus(Config.COMPANY\_CHOSEN);  
 company.setEmail(email);  
 boolean res = companyService.save(company);  
 if (!res){  
 attributes.addAttribute("msg","添加失败,请联系管理员");  
 return "redirect:/user/info";  
 }  
 attributes.addAttribute("msg","添加成功");  
 return "redirect:/user/info";  
 }  
  
 @RequestMapping("delete")  
 @Transactional(rollbackFor = RuntimeException.class)  
 public String delete(Integer id,RedirectAttributes attributes){  
 SecurityUtils.getSubject().checkPermission("/company/delete");  
 SecurityUtils.getSubject().checkRole("admin");  
 boolean res = companyService.removeById(id);  
 if (!res){  
 attributes.addAttribute("msg","删除失败,请联系管理员");  
 return "redirect:list";  
 }  
 attributes.addAttribute("msg","删除成功");  
 return "redirect:list";  
 }  
  
 @RequestMapping("update")  
 public String update(Integer id,String name, String email,String leader, RedirectAttributes attributes){  
 SecurityUtils.getSubject().checkPermission("/company/update");  
 SecurityUtils.getSubject().checkRole("admin");  
 BusCompany company = companyService.getById(id);  
 company.setName(name);  
 company.setEmail(email);  
 company.setLeader(leader);  
 boolean res = companyService.updateById(company);  
 if (!res){  
 attributes.addAttribute("msg","修改失败,请联系管理员");  
 return "redirect:list";  
 }  
 attributes.addAttribute("msg","修改成功");  
 return "redirect:list";  
 }  
  
 @RequestMapping("addUser")  
 public String addUser(Integer id, Integer userId, HttpSession session, RedirectAttributes attributes){  
 SecurityUtils.getSubject().checkPermission("/company/addUser");  
 SecurityUtils.getSubject().checkRole("company");  
 BusCompany company = companyService.getById(id);  
 if(userId!=null){  
 company.setUserId(userId);  
 }  
 boolean res = companyService.updateById(company);  
 if(!res){  
 throw new RuntimeException("添加失败");  
 }  
 session.setAttribute("company", company);  
 attributes.addAttribute("msg","添加成功");  
 return "redirect:/companyInfo";  
 }  
  
 @RequestMapping("pay")  
 public String pay(Integer orderId, RedirectAttributes attributes){  
 SecurityUtils.*getSubject*().checkPermission("/company/pay");  
 SecurityUtils.*getSubject*().checkRole("company");  
  
 BusOrder order = orderService.getById(orderId);  
 order.setStatus(Config.*ORDER\_DONE*);  
 boolean res = orderService.updateById(order);  
 if (!res){  
 attributes.addAttribute("msg","更新失败，请联系管理员");  
 }  
 attributes.addAttribute("msg","支付成功");  
 return "redirect:/user/info";  
 }  
  
 @RequestMapping("payPage")  
 public String payPage(){  
 return "redirect:/pay.html";  
 }  
  
 @RequestMapping("removeOrder")  
 public String removeOrder(Integer orderId, RedirectAttributes attributes){  
 SecurityUtils.getSubject().checkPermission("/company/removeOrder");  
 SecurityUtils.getSubject().checkRole("company");  
 boolean res = orderService.removeById(orderId);  
 if (!res){  
 attributes.addAttribute("msg","删除失败，请联系管理员");  
 }  
 attributes.addAttribute("msg","删除成功");  
 return "redirect:/user/info";  
 }  
}