项目名称：

通用后台权限管理系统

项目目标：

1. 掌握软件开发的基本流程
2. 掌握springboot框架
3. 掌握mybatis框架各知识点
4. 掌握shiro安全框架的认证和授权
5. 掌握html和javascript各知识点
6. 掌握mysql各知识点

项目大纲：

1. 背景知识
2. 需求分析
3. 功能分析
4. 架构设计
5. 代码开发

目录

[1、背景知识 2](#_Toc41922574)

[2、需求分析 2](#_Toc41922575)

[3、功能分析 3](#_Toc41922576)

[4、原型设计 10](#_Toc41922577)

[5、架构设计 10](#_Toc41922578)

[5.1、整体架构设计 11](#_Toc41922579)

[5.7、数据模型设计 11](#_Toc41922580)

[6、 代码开发 13](#_Toc41922581)

[src/main/java/com/example/Util/ProjectConstants.java 13](#_Toc41922582)

[src/main/java/com/example/Util/TreeUtils.java 13](#_Toc41922583)

[src/main/java/com/example/shiro/ShiroConfig.java 16](#_Toc41922584)

[src/main/java/com/example/shiro/ShiroRealm.java 18](#_Toc41922585)

[src/main/java/com/example/shiro/ShiroUtil.java 21](#_Toc41922586)

[src/main/java/com/example/entity/Dept.java 22](#_Toc41922587)

[src/main/java/com/example/entity/Menu.java 23](#_Toc41922588)

[src/main/java/com/example/entity/MsJson.java 25](#_Toc41922589)

[src/main/java/com/example/entity/MsTree.java 26](#_Toc41922590)

[src/main/java/com/example/entity/Role.java 26](#_Toc41922591)

[src/main/java/com/example/entity/RoleMenu.java 28](#_Toc41922592)

[src/main/java/com/example/entity/User.java 28](#_Toc41922593)

[src/main/java/com/example/entity/UserRole.java 31](#_Toc41922594)

[src/main/java/com/example/dao/DeptMapper.java 32](#_Toc41922595)

[src/main/java/com/example/dao/MenuMapper.java 32](#_Toc41922596)

[src/main/java/com/example/dao/RoleMapper.java 33](#_Toc41922597)

[src/main/java/com/example/dao/RoleMenuMapper.java 34](#_Toc41922598)

[src/main/java/com/example/dao/UserMapper.java 35](#_Toc41922599)

[src/main/java/com/example/dao/UserRoleMapper.java 35](#_Toc41922600)

[src/main/java/com/example/controller/DeptController.java 36](#_Toc41922601)

[src/main/java/com/example/controller/LoginController.java 38](#_Toc41922602)

[src/main/java/com/example/controller/MenuController.java 39](#_Toc41922603)

[src/main/java/com/example/controller/PageController.java 41](#_Toc41922604)

[src/main/java/com/example/controller/RoleController.java 42](#_Toc41922605)

[src/main/java/com/example/controller/UserController.java 45](#_Toc41922606)

[src/main/java/com/example/biz/DeptBiz.java 49](#_Toc41922607)

[src/main/java/com/example/biz/MenuBiz.java 49](#_Toc41922608)

[src/main/java/com/example/biz/RoleBiz.java 50](#_Toc41922609)

[src/main/java/com/example/biz/RoleMenuBiz.java 51](#_Toc41922610)

[src/main/java/com/example/biz/UserBiz.java 51](#_Toc41922611)

[src/main/java/com/example/biz/UserRoleBiz.java 52](#_Toc41922612)

[src/main/java/com/example/biz/impl/UserRoleBizImpl.java 52](#_Toc41922613)

[src/main/java/com/example/biz/impl/UserBizImpl.java 53](#_Toc41922614)

[src/main/java/com/example/biz/impl/RoleMenuBizImpl.java 54](#_Toc41922615)

[src/main/java/com/example/biz/impl/RoleBizImpl.java 55](#_Toc41922616)

[src/main/java/com/example/biz/impl/MenuBizImpl.java 57](#_Toc41922617)

[src/main/java/com/example/biz/impl/DeptBizImpl.java 59](#_Toc41922618)

## 1、背景知识

在当前风起云涌的电商领域和广大企业信息化管理领域，软件系统的后台管理都是不可或缺的组成部分，而后台管理中的权限管理模块是衡量一个后台管理是否安全高效的重要指标。

## 2、需求分析

由于不同的后台管理系统需求多样化，本次开发的是通用型，对于大多数的后台管理系统逻辑都已足够使用，主要应用于WEB应用程序，如：网站管理后台、CMS、CRM、OA、商城后台等等。

目前常见的权限控制分为四大类：

1. **用户管理**

对于后台产品而言，是需要公司内部人员去创建账号的。而这个账号就是一把钥匙，我们通过控制账号所具备的权限，进而控制这个员工的所操作范围。

1. **角色管理**

角色往往是基于业务管理需求而预先在系统中设定好的固定标签，每个角色对应明确的系统权限，他是一个集合的概念，是众多最小权限颗粒的组成。我们通过把权限给这个角色，再把角色给账号，从而实现账号的权限，因此它承担了一个桥梁的作用。引入角色这个概念，可以帮助我们灵活的扩展，使一个账号可以具备多种角色。

1. **功能权限控制**

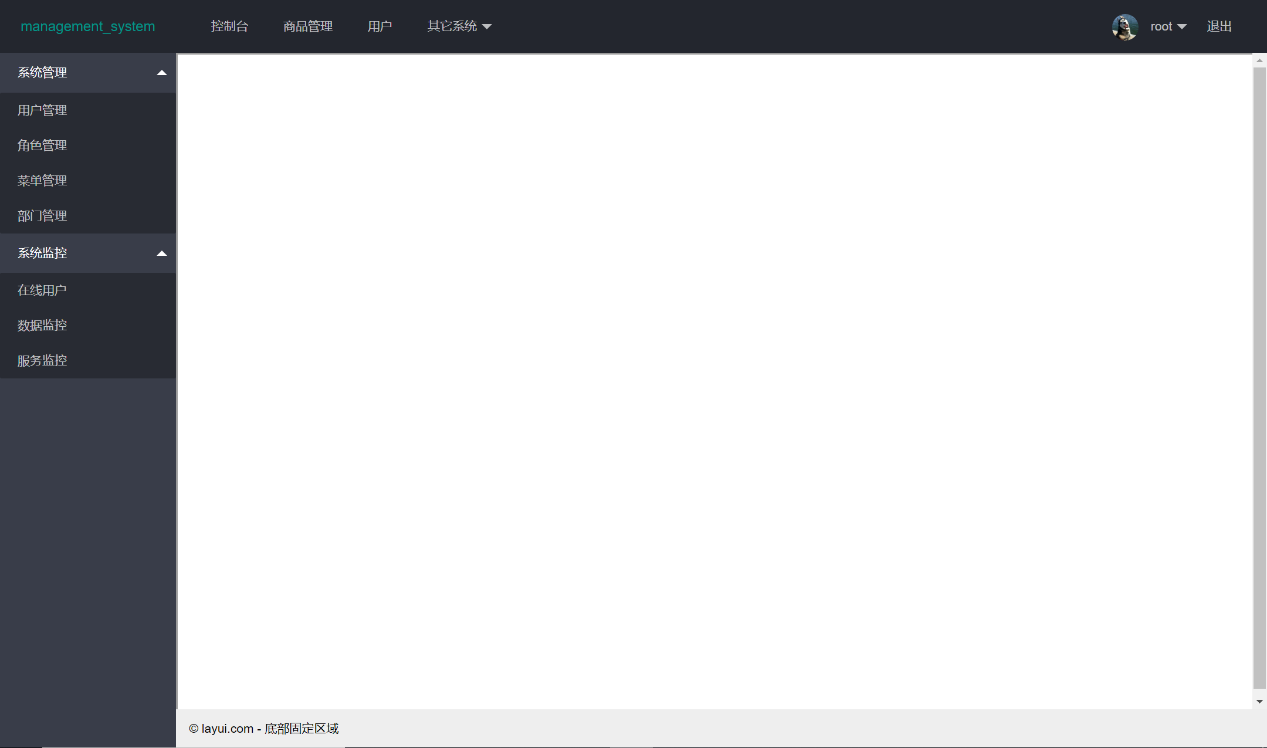
功能权限定义：为可见、可以操作的功能范围。在目录、菜单上加权限控制，有权限的就可以访问对应模块，没有的连菜单名都看不到。在业务模块的功能按钮上加权限控制，最小粒度的控制用户行为。

1. **数据权限控制**

数据权限控制主要控制某条数据记录对用户是否可见，结合功能权限可以更灵活的配置业务过程中每一位员工的功能操作权限及数据可见范围，全面保障企业数据的安全性。

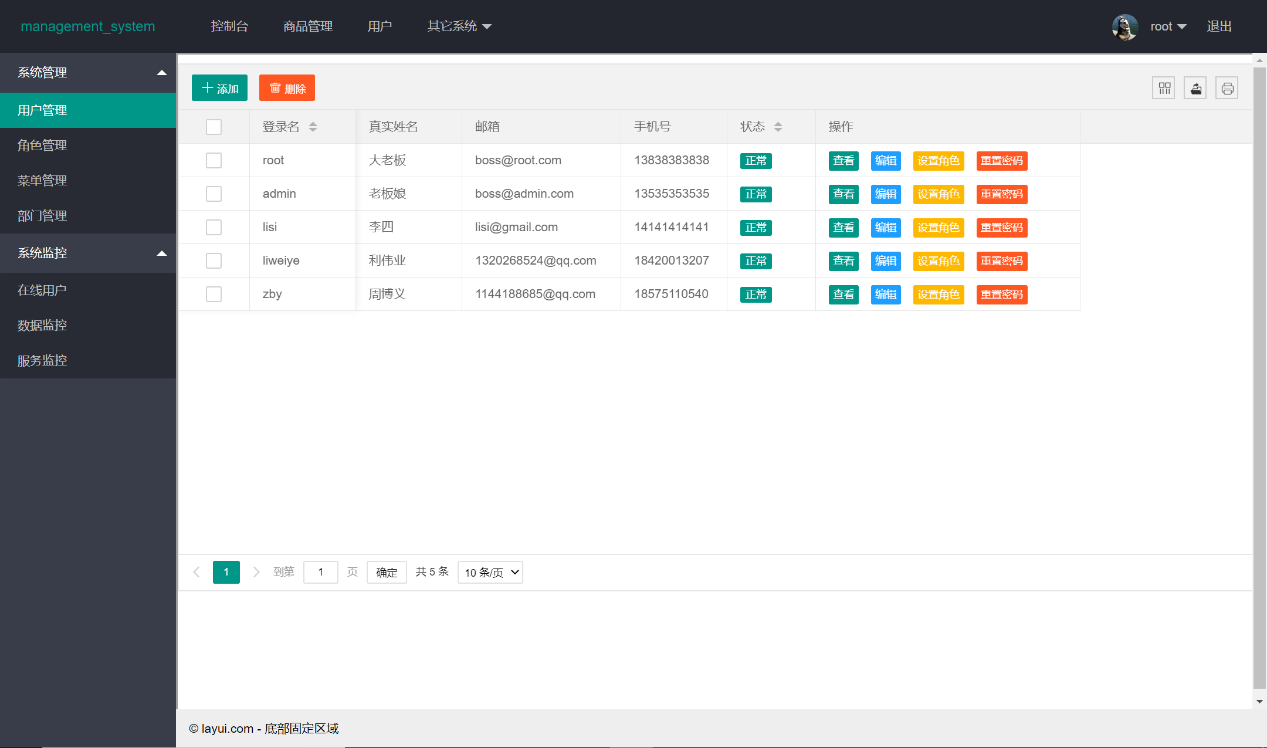
## 3、功能分析

1. 登录功能
   1. 登录页面

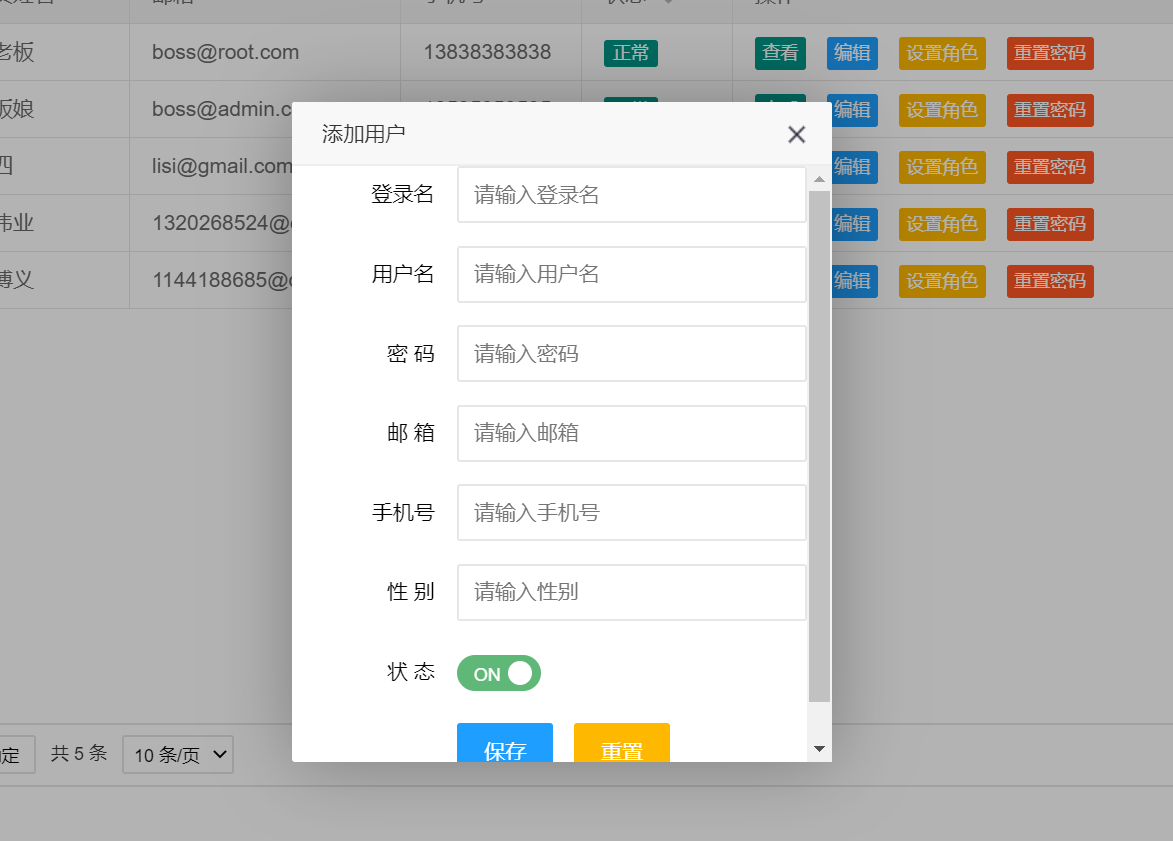


* 1. 异常登录提示错误信息

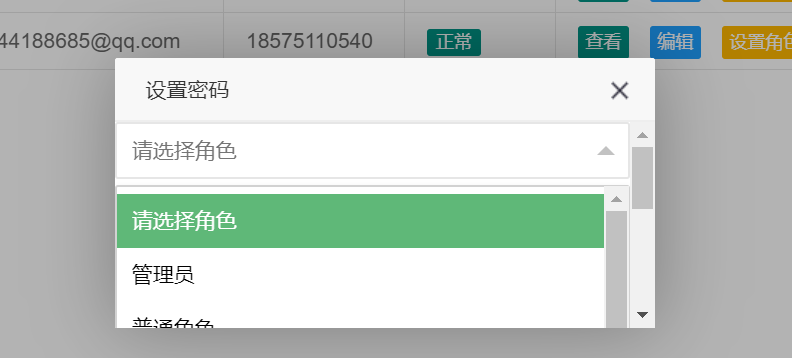
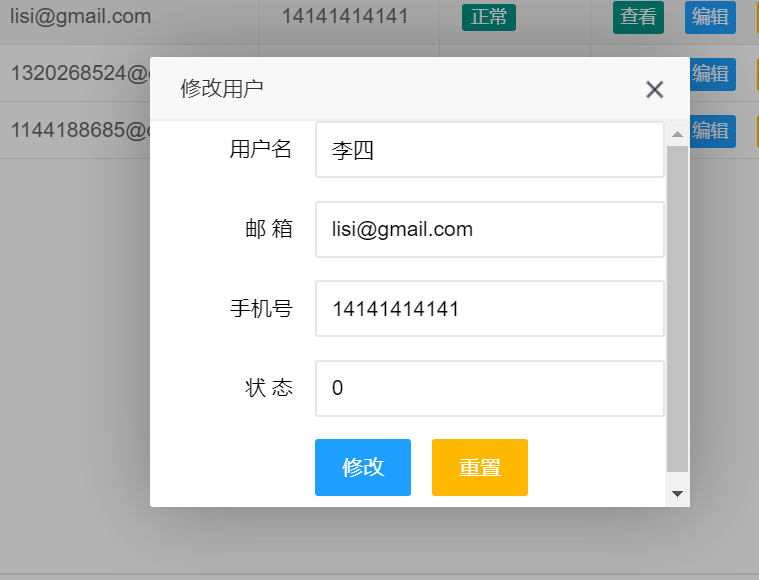
1. 用户管理
   1. 查询用户



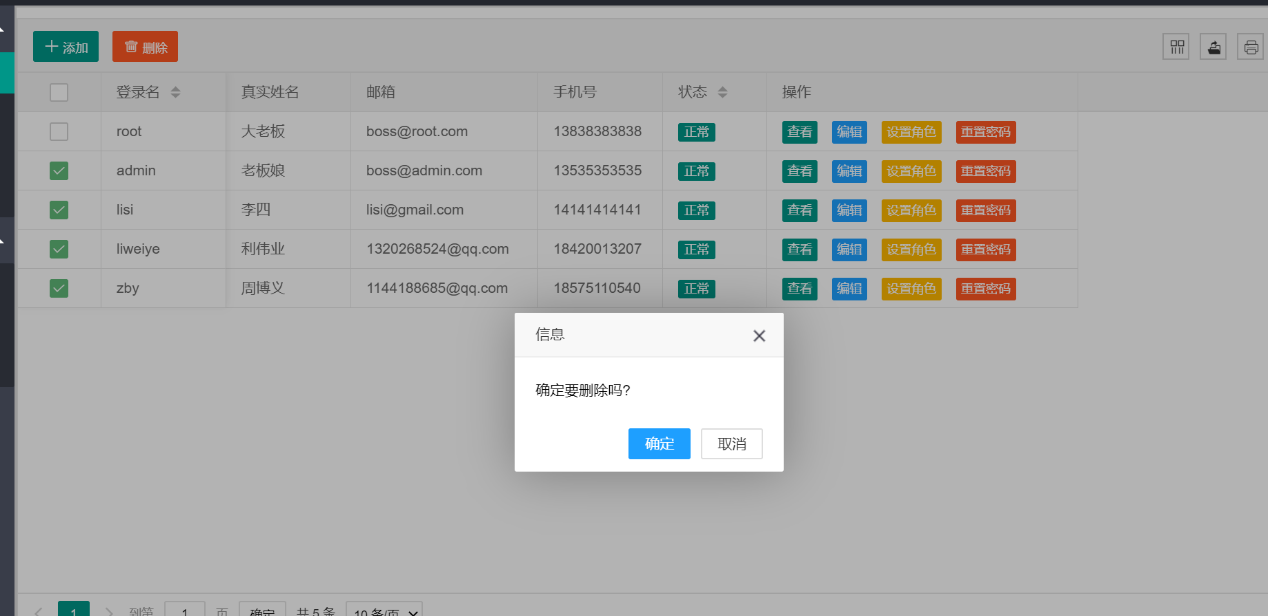
* 1. 添加用户，设置角色



* 1. 修改用户，禁用用户



* 1. 删除用户



* 1. 修改密码



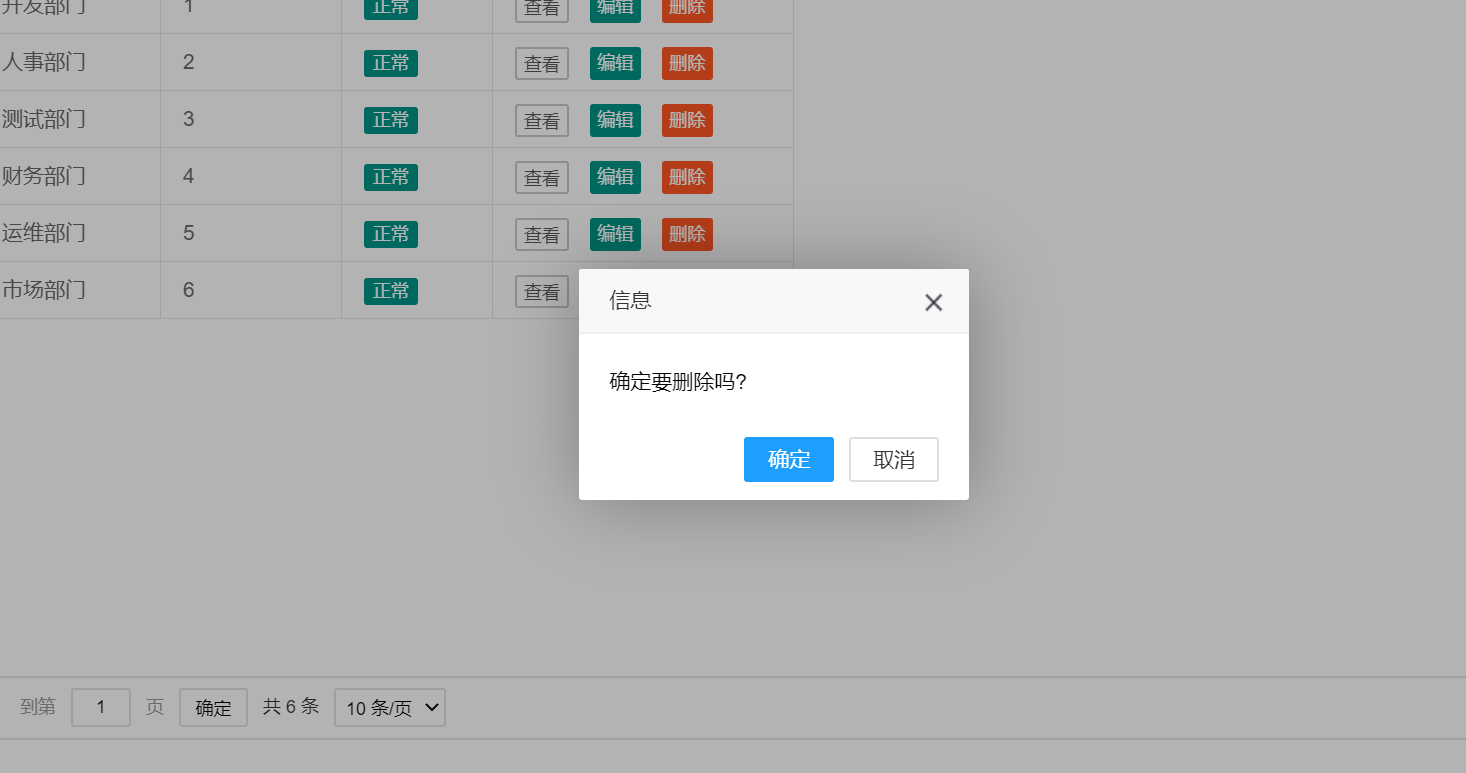
1. 部门管理
   1. 查询部门
   2. 添加部门



* 1. 修改部门，禁用部门



* 1. 删除部门



1. 角色管理
   1. 查询角色



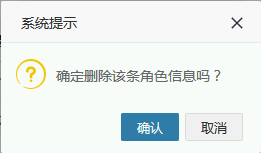
* 1. 添加角色



* 1. 修改角色



* 1. 删除角色



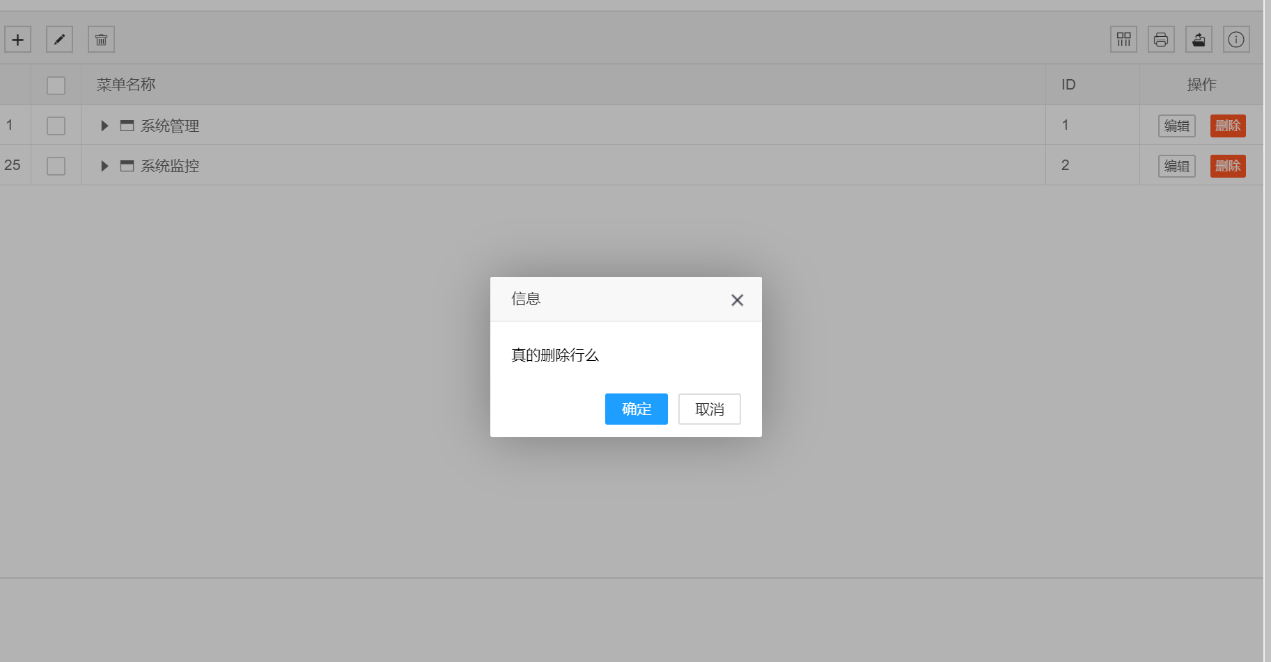
1. 菜单管理
   1. 查询菜单
   2. 添加菜单



* 1. 修改菜单



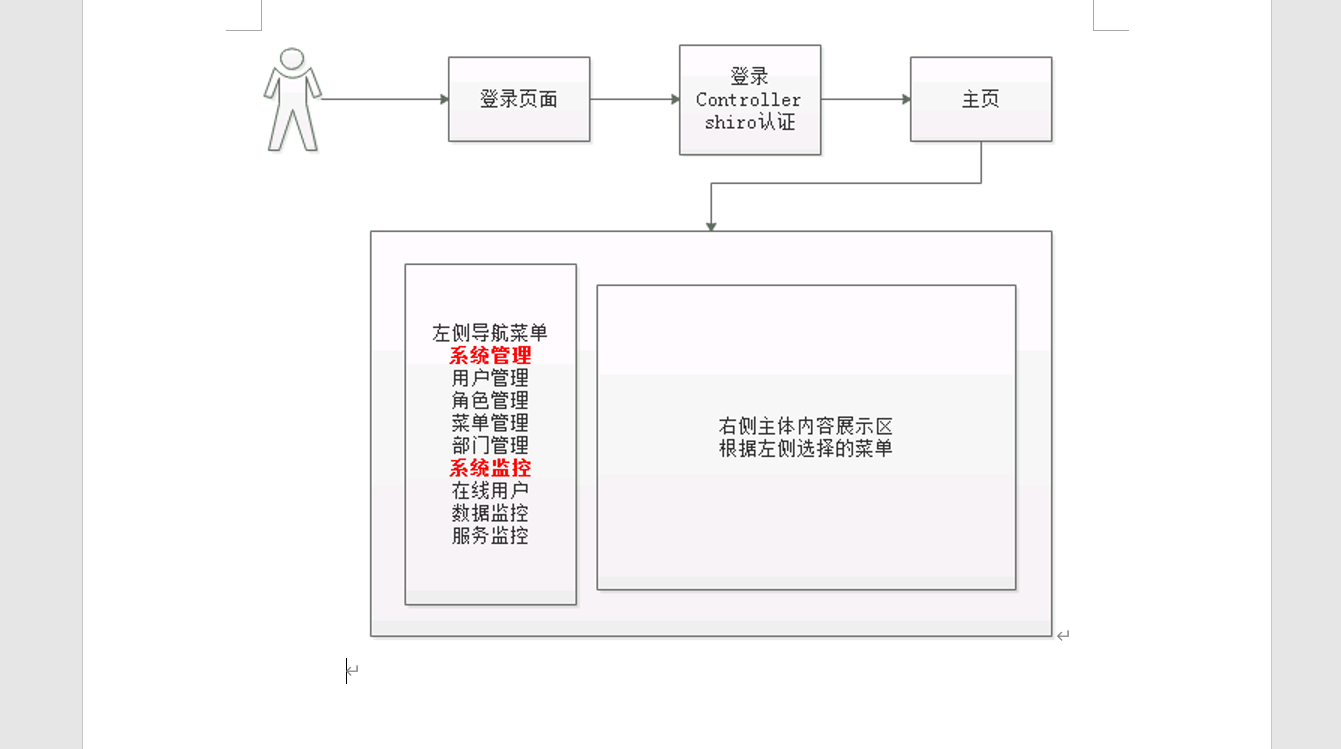
* 1. 删除菜单



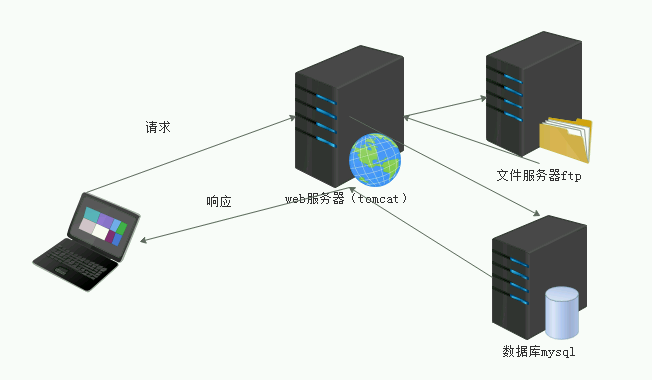
## 4、原型设计

产品经理设计产品原形

## 5、架构设计



### 5.1、整体架构设计



### 5.7、数据模型设计

#### 5.7.1、用户表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户信息表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| userid | int | 50 | not null,pk | 用户ID |
| loginname | varchar | 50 | not null | 登录名称 |
| usrname | varchar | 50 | not null | 用户名称 |
| department | varchar | 50 | not null | 部门 |
| telnum | varchar | 50 | not null | 手机 |
| state | varchar | 50 | not null,fk | 状态 |
| ct | varchar | 50 | not null,fk | 创建时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 部门信息表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| dtname | varchar | 50 | not nul，pk | 部门名称 |
| sort | int | 50 | not nul | 排序 |
| state | varchar | 50 | not nul,fk | 状态 |
| ct | varchar | 50 | not nul | 创建时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 角色信息表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| roleid | int | 50 | not null，pk | 角色编号 |
| rolename | varchar | 50 | not null | 角色名称 |
| pmchar | varchar | 50 | not null | 权限字符 |
| draworder | int | 50 | not null | 显示顺序 |
| state | varchar | 50 | not null,fk | 状态 |
| ct | varchar | 50 | not null,fk | 创建时间 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 菜单信息表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| mname | varchar | 50 | not null,pk | 菜单名称 |
| sort | int | 50 | not null | 排序 |
| reqaddress | varchar | 50 | not null | 请求地址 |
| type | varchar | 50 | not null | 类型 |
| visible | varchar | 50 | not null | 可见 |
| pmID | varchar | 50 | not null | 权限标识 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 用户角色关系表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| userid | int | 50 | not null | 用户ID |
| roleid | int | 50 | not null | 角色编号 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 角色菜单关系表 | | | | |
| 字段名 | 数据类型 | 长度 | 说明 | 描述 |
| roleid | int | 50 | not null | 角色编号 |
| mname | varchar | 50 | not null | 菜单名称 |

## 代码开发

### src/main/java/com/example/Util/ProjectConstants.java

package com.example.Util;  
  
public class ProjectConstants {  
  
 public static final int successCode = 0;//增删改成功  
  
 public static final int failCode = 1;//增删改失败  
  
 public static final String insertSuccessMsg = "添加成功";  
  
 public static final String insertFailMsg = "添加失败";  
  
 public static final String updateSuccessMsg = "修改成功";  
  
 public static final String updateFailMsg = "修改失败";  
  
 public static final String deleteSuccessMsg = "删除成功";  
  
 public static final String deleteFailMsg = "删除失败";  
  
 public static final String algorithmName = "MD5";//加密类型  
  
 public static final int hashIterations = 1000;//加密次数  
}

### src/main/java/com/example/Util/TreeUtils.java

package com.example.Util;  
  
import com.example.entity.Menu;  
import com.example.entity.MsTree;  
  
import java.util.ArrayList;  
import java.util.Iterator;  
import java.util.List;  
  
//树状菜单工具类  
public class TreeUtils {  
  
 */\*\*  
 \* 根据父节点的ID获取所有子节点  
 \** ***@param*** *list 分类表  
 \** ***@param*** *parentId 传入的父节点ID  
 \** ***@return*** *String  
 \*/* public static List<MsTree> getChildPerms(List<Menu> list, int parentId) {  
 List<MsTree> returnList = new ArrayList<MsTree>();  
 for (Menu menu : list) {  
 //一、根据传入的某个父节点ID，遍历该父节点的所有子节点  
 if (menu.getParentId() == parentId) {  
 MsTree tree = fromMenuToTree(menu);  
 //开始递归，把所有菜单和当前菜单放入  
 recursionFn(list, tree);  
 returnList.add(tree);  
 }  
 }  
 return returnList;  
 }  
  
 */\*\*  
 \* 递归列表  
 \** ***@param*** *list  
 \** ***@param*** *tree  
 \*/* private static void recursionFn(List list, MsTree tree) {  
 //得到t的子节点列表  
 List<MsTree> childList = getChildList(list, tree);  
 tree.setChildren(childList);  
 for (MsTree treeChild :childList) {  
 if (hasChild(list, treeChild)) {  
 //判断是否有子节点  
 Iterator<MsTree> it = childList.iterator();  
 while (it.hasNext()) {  
 MsTree next = it.next();  
 recursionFn(list, next);  
 }  
 }  
 }  
 }  
  
 */\*\*  
 \* 得到子节点列表  
 \*/* private static List<MsTree> getChildList(List<Menu> list, MsTree tree) {  
 List<MsTree> treeList = new ArrayList<MsTree>();  
 Iterator<Menu> it = list.iterator();  
 while (it.hasNext()) {  
 Menu menu = (Menu) it.next();  
 if (menu.getParentId() == tree.getId()) {  
 MsTree msTree = fromMenuToTree(menu);  
 treeList.add(msTree);  
 }  
 }  
 return treeList;  
 }  
  
 */\*\*  
 \* 判断是否有子节点  
 \*/* private static boolean hasChild(List<Menu> list, MsTree tree) {  
 int size = getChildList(list, tree).size();  
 if (size > 0) {  
 return true;  
 } else {  
 return false;  
 }  
 }  
  
 */\*\*  
 \* 将menu对象转换成tree对象  
 \** ***@param*** *menu  
 \** ***@return*** *\*/* private static MsTree fromMenuToTree(Menu menu) {  
 //构造tree对象  
 MsTree tree = new MsTree();  
 tree.setId(menu.getMenuId());  
 tree.setTitle(menu.getMenuName());  
 tree.setChecked(false);  
 tree.setUrl(menu.getUrl());  
 return tree;  
 }  
}

### src/main/java/com/example/shiro/ShiroConfig.java

package com.example.shiro;  
  
import com.example.util.MyConstants;  
import org.apache.shiro.authc.credential.CredentialsMatcher;  
import org.apache.shiro.authc.credential.HashedCredentialsMatcher;  
import org.apache.shiro.spring.web.ShiroFilterFactoryBean;  
import org.apache.shiro.web.mgt.DefaultWebSecurityManager;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
  
import java.util.LinkedHashMap;  
import java.util.Map;  
  
@Configuration  
public class ShiroConfig {  
  
 */\*\*  
 \* realm  
 \* 定义一个bean，id为方法名  
 \* <bean id="" class=""></bean>  
 \** ***@return*** *\*/* @Bean  
 public ShiroRealm shiroRealm() {  
 ShiroRealm shiroRealm = new ShiroRealm();  
 shiroRealm.setCredentialsMatcher(credentialsMatcher());  
 return shiroRealm;  
 }  
  
 */\*\*  
 \* securityManager  
 \*/* @Bean  
 public DefaultWebSecurityManager defaultWebSecurityManager() {  
 DefaultWebSecurityManager defaultWebSecurityManager = new DefaultWebSecurityManager();  
 defaultWebSecurityManager.setRealm(shiroRealm());  
 return defaultWebSecurityManager;  
 }  
  
 */\*\*  
 \* shiroFilterFactoryBean  
 \*/* @Bean  
 public ShiroFilterFactoryBean shiroFilterFactoryBean() {  
 ShiroFilterFactoryBean shiroFilterFactoryBean = new ShiroFilterFactoryBean();  
 shiroFilterFactoryBean.setSecurityManager(defaultWebSecurityManager());  
 */\*\*  
 \* 认证过滤器：  
 \* anon: 无需认证  
 \* authc: 必须认证才能到达  
 \* user: 使用remember me（一周内免登录）的时候使用  
 \* perms: 访问的资源需要某个权限才能到达  
 \* roles: 访问的资源需要某个角色才能到达  
 \*/* //LinkedHashMap是一个有序的HashMap，是一个链条  
 Map<String, String> map = new LinkedHashMap<>();  
 //放行的页面请求  
 map.put("/toLogin", "anon");  
 map.put("/login", "anon");  
 //添加到达页面需要的权限  
 //User相关  
 map.put("/toUser", "perms[/systemUserView]");  
 map.put("/user/selectAllUser", "perms[/systemUserSelect]");  
 map.put("/user/insertUser", "perms[/systemUserInsert]");  
 map.put("/user/updateUser", "perms[/systemUserUpdate]");  
 map.put("/user/deleteUser", "perms[/systemUserDelete]");  
 map.put("/user/selectByLoginName", "perms[/systemUserSelect]");  
 //Role相关  
 map.put("/toRole", "perms[/systemRoleView]");  
 map.put("/role/selectAll", "perms[/systemRoleSelect]");  
 map.put("/role/addRole", "perms[/systemRoleInsert]");  
 map.put("/role/delRole", "perms[/systemRoleDelete]");  
 map.put("/role/editRole", "perms[/systemRoleUpdate]");  
 map.put("/role/deleteRole", "perms[/systemRoleDelete]");  
 //菜单相关  
 map.put("/toMenu", "perms[/systemMenuView]");  
 map.put("/menu/selectAllMenu", "perms[/systemMenuSelect]");  
 map.put("/menu/selectAll", "perms[/systemMenuSelect]");  
 map.put("/menu/addMenu", "perms[/systemMenuInsert]");  
 map.put("/menu/delMenu", "perms[/systemMenuDelete]");  
 map.put("/menu/deleteMenu", "perms[/systemMenuDelete]");  
 map.put("/menu/editMenu", "perms[/systemMenuUpdate]");  
 //部门相关  
 map.put("/toDept", "perms[/systemDeptView]");  
 map.put("/dept/selectAll", "perms[/systemDeptSelect]");  
 map.put("/dept/addDept", "perms[/systemDeptInsert]");  
 map.put("/dept/deleteDept", "perms[/systemDeptDelete]");  
 map.put("/dept/delDept", "perms[/systemDeptDelete]");  
 map.put("/dept/editDept", "perms[/systemDeptUpdate]");  
 //过滤的页面请求(需要授权)  
 map.put("/\*", "authc");  
  
 shiroFilterFactoryBean.setFilterChainDefinitionMap(map);  
 //修改默认登录页面  
 shiroFilterFactoryBean.setLoginUrl("/toLogin");  
 //指定未授权提示页面  
 shiroFilterFactoryBean.setUnauthorizedUrl("/toUnau");  
 return shiroFilterFactoryBean;  
 }  
  
 */\*\*  
 \* 实例化密码比较器  
 \*/* @Bean  
 public CredentialsMatcher credentialsMatcher() {  
 HashedCredentialsMatcher credentialsMatcher =  
 new HashedCredentialsMatcher();  
 //使用MD5加密  
 credentialsMatcher.setHashAlgorithmName(MyConstants.algorithmName);  
 //加密1000次  
 credentialsMatcher.setHashIterations(MyConstants.hashIterations);  
 return credentialsMatcher;  
 }  
}

### src/main/java/com/example/shiro/ShiroRealm.java

package com.example.shiro;  
  
import com.example.biz.MenuBiz;  
import com.example.biz.RoleMenuBiz;  
import com.example.biz.UserBiz;  
import com.example.biz.UserRoleBiz;  
import com.example.entity.User;  
import org.apache.shiro.SecurityUtils;  
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.apache.shiro.util.ByteSource;  
import org.springframework.beans.factory.annotation.Autowired;  
  
import java.util.ArrayList;  
import java.util.List;  
  
public class ShiroRealm extends AuthorizingRealm {  
  
 @Autowired  
 private UserBiz userBizImpl;  
  
 @Autowired  
 private UserRoleBiz userRoleBizImpl;  
  
 @Autowired  
 private RoleMenuBiz roleMenuBizImpl;  
  
 @Autowired  
 private MenuBiz menuBizImpl;  
  
 */\*\*  
 \* shiro安全框架的授权  
 \** ***@param*** *principalCollection  
 \** ***@return*** *AuthorizationInfo  
 \*/* @Override  
 protected AuthorizationInfo doGetAuthorizationInfo(PrincipalCollection principalCollection) {  
 SimpleAuthorizationInfo simpleAuthorizationInfo =  
 new SimpleAuthorizationInfo();  
  
 //获取用户信息  
 User user = (User) SecurityUtils.getSubject().getPrincipal();  
 //权限字符串从数据库中获取  
 int roleId = userRoleBizImpl.selectRoleIdByUserId(user.getUserId());//从数据库中获取角色ID  
 List<Integer> menuIdList = roleMenuBizImpl.selectMenuIdByRoleId(roleId);//从数据库中获取菜单ID  
 List<String> permsList = new ArrayList<>();  
 for (int menuId : menuIdList) {  
 permsList.add(menuBizImpl.selectPermsByMenuId(menuId));//将每个菜单ID对应的权限字符串从数据库中搜索出来  
 }  
 for (String perms : permsList) {  
 simpleAuthorizationInfo.addStringPermission(perms);//将所有权限字符串添加到用户权限中  
 }  
 return simpleAuthorizationInfo;  
 }  
  
 */\*\*  
 \* shiro安全框架的认证  
 \** ***@param*** *authenticationToken  
 \** ***@return*** *AuthenticationInfo 返回为null时说明认证失败  
 \** ***@throws*** *AuthenticationException  
 \** ***@Return*** *AuthenticationInfo  
 \*/* @Override  
 protected AuthenticationInfo doGetAuthenticationInfo(AuthenticationToken authenticationToken) throws AuthenticationException {  
 //取出令牌信息  
 UsernamePasswordToken usernamePasswordToken =  
 (UsernamePasswordToken) authenticationToken;  
  
 //登录认证分两个步骤，步骤一：查询用户是否存在  
 String loginName = usernamePasswordToken.getUsername();  
 User user = userBizImpl.selectByLoginName(loginName);  
 if (user == null) {  
 return null;  
 }  
  
 //从数据库中获取密码和盐  
 String password = user.getPassword();  
 String salt = user.getSalt();  
 ByteSource byteSource = ByteSource.Util.bytes(salt);  
  
 //创建简单的shiro认证消息对象  
 SimpleAuthenticationInfo simpleAuthenticationInfo;  
  
 //步骤二：查询密码是否正确  
 */\*\*  
 \* Object principal 实体，例如这里就是User对象  
 \* Object credentials 数据库中的密码  
 \* ByteSource credentialsSalt 创建密码密文时加密用的盐(从数据库中获取)  
 \* String realmName 当前realm的name，用getRealm()方法获取，getRealm()方法在这个类的曾祖父类中，已被继承  
 \*/* simpleAuthenticationInfo = new SimpleAuthenticationInfo(user, password, byteSource, getName());  
  
 return simpleAuthenticationInfo;  
 }  
}

### src/main/java/com/example/shiro/ShiroUtil.java

package com.example.shiro;  
  
import org.apache.shiro.crypto.hash.SimpleHash;  
import org.springframework.util.DigestUtils;  
  
import java.util.UUID;  
  
public class ShiroUtil {  
  
 */\*\*  
 \* 传入一个明文，返回一个加密后的密文  
 \*/* public static String getCiphertext(String plaintext) {  
 String ciphertext = DigestUtils.md5DigestAsHex(plaintext.getBytes());  
 return ciphertext;  
 }  
  
 */\*\*  
 \* 加盐加密  
 \** ***@param*** *salt  
 \** ***@param*** *message  
 \** ***@return*** *\*/* public static String encryptionBySalt(String salt, String message){  
 */\*\*  
 \* shiro加密：  
 \* String algorithmName 加密算法  
 \* Object source 明文  
 \* Object salt 盐值  
 \* int hashIterations 加密次数  
 \*/* String algorithmName = com.example.util.MyConstants.algorithmName;  
 int hashIterations = com.example.util.MyConstants.hashIterations;  
 SimpleHash simpleHash = new SimpleHash(algorithmName, message, salt, hashIterations);  
 return simpleHash.toString();  
 }  
  
 public static void main(String[] args) {  
 System.out.println(encryptionBySalt("root", "root"));  
 System.out.println(encryptionBySalt("admin", "admin"));  
 }  
}

### src/main/java/com/example/entity/Dept.java

package com.example.entity;  
  
public class Dept {  
 private Integer deptId;  
  
 private String deptName;  
  
 private Integer orderNum;  
  
 private String status;  
  
 private String delFlag;  
  
 public Integer getDeptId() {  
 return deptId;  
 }  
  
 public void setDeptId(Integer deptId) {  
 this.deptId = deptId;  
 }  
  
 public String getDeptName() {  
 return deptName;  
 }  
  
 public void setDeptName(String deptName) {  
 this.deptName = deptName == null ? null : deptName.trim();  
 }  
  
 public Integer getOrderNum() {  
 return orderNum;  
 }  
  
 public void setOrderNum(Integer orderNum) {  
 this.orderNum = orderNum;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status == null ? null : status.trim();  
 }  
  
 public String getDelFlag() {  
 return delFlag;  
 }  
  
 public void setDelFlag(String delFlag) {  
 this.delFlag = delFlag == null ? null : delFlag.trim();  
 }  
}

### src/main/java/com/example/entity/Menu.java

package com.example.entity;  
  
public class Menu {  
 private Integer menuId;  
  
 private String menuName;  
  
 private Integer parentId;  
  
 private Integer orderNum;  
  
 private String url;  
  
 private String menuType;  
  
 private String visible;  
  
 private String perms;  
  
 private String icon;  
  
 public Integer getMenuId() {  
 return menuId;  
 }  
  
 public void setMenuId(Integer menuId) {  
 this.menuId = menuId;  
 }  
  
 public String getMenuName() {  
 return menuName;  
 }  
  
 public void setMenuName(String menuName) {  
 this.menuName = menuName == null ? null : menuName.trim();  
 }  
  
 public Integer getParentId() {  
 return parentId;  
 }  
  
 public void setParentId(Integer parentId) {  
 this.parentId = parentId;  
 }  
  
 public Integer getOrderNum() {  
 return orderNum;  
 }  
  
 public void setOrderNum(Integer orderNum) {  
 this.orderNum = orderNum;  
 }  
  
 public String getUrl() {  
 return url;  
 }  
  
 public void setUrl(String url) {  
 this.url = url == null ? null : url.trim();  
 }  
  
 public String getMenuType() {  
 return menuType;  
 }  
  
 public void setMenuType(String menuType) {  
 this.menuType = menuType == null ? null : menuType.trim();  
 }  
  
 public String getVisible() {  
 return visible;  
 }  
  
 public void setVisible(String visible) {  
 this.visible = visible == null ? null : visible.trim();  
 }  
  
 public String getPerms() {  
 return perms;  
 }  
  
 public void setPerms(String perms) {  
 this.perms = perms == null ? null : perms.trim();  
 }  
  
 public String getIcon() {  
 return icon;  
 }  
  
 public void setIcon(String icon) {  
 this.icon = icon == null ? null : icon.trim();  
 }  
}

### src/main/java/com/example/entity/MsJson.java

package com.example.entity;  
  
import lombok.Data;  
  
import java.util.List;  
  
@Data  
public class MsJson {  
 private int code;  
 private String msg;  
 private long count;  
 private List<?> data;  
}

### src/main/java/com/example/entity/MsTree.java

package com.example.entity;  
  
import lombok.Data;  
  
import java.util.List;  
  
@Data  
public class MsTree {  
  
 private String title;  
 private int id;  
 private String field;  
 private boolean checked;  
 private boolean spread;  
 private String url;  
 private List<MsTree> children;  
}

### src/main/java/com/example/entity/Role.java

package com.example.entity;  
  
public class Role {  
 private Integer roleId;  
  
 private String roleName;  
  
 private String roleKey;  
  
 private Integer roleSort;  
  
 private String status;  
  
 private String delFlag;  
  
 public Integer getRoleId() {  
 return roleId;  
 }  
  
 public void setRoleId(Integer roleId) {  
 this.roleId = roleId;  
 }  
  
 public String getRoleName() {  
 return roleName;  
 }  
  
 public void setRoleName(String roleName) {  
 this.roleName = roleName == null ? null : roleName.trim();  
 }  
  
 public String getRoleKey() {  
 return roleKey;  
 }  
  
 public void setRoleKey(String roleKey) {  
 this.roleKey = roleKey == null ? null : roleKey.trim();  
 }  
  
 public Integer getRoleSort() {  
 return roleSort;  
 }  
  
 public void setRoleSort(Integer roleSort) {  
 this.roleSort = roleSort;  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status == null ? null : status.trim();  
 }  
  
 public String getDelFlag() {  
 return delFlag;  
 }  
  
 public void setDelFlag(String delFlag) {  
 this.delFlag = delFlag == null ? null : delFlag.trim();  
 }  
}

### src/main/java/com/example/entity/RoleMenu.java

package com.example.entity;  
  
public class RoleMenu {  
 private Integer roleId;  
  
 private Integer menuId;  
  
 public Integer getRoleId() {  
 return roleId;  
 }  
  
 public void setRoleId(Integer roleId) {  
 this.roleId = roleId;  
 }  
  
 public Integer getMenuId() {  
 return menuId;  
 }  
  
 public void setMenuId(Integer menuId) {  
 this.menuId = menuId;  
 }  
}

### src/main/java/com/example/entity/User.java

package com.example.entity;  
  
public class User {  
 private Integer userId;  
  
 private Integer deptId;  
  
 private String loginName;  
  
 private String userName;  
  
 private String email;  
  
 private String phonenumber;  
  
 private String sex;  
  
 private String avatar;  
  
 private String password;  
  
 private String salt;  
  
 private String status;  
  
 private String delFlag;  
  
 public Integer getUserId() {  
 return userId;  
 }  
  
 public void setUserId(Integer userId) {  
 this.userId = userId;  
 }  
  
 public Integer getDeptId() {  
 return deptId;  
 }  
  
 public void setDeptId(Integer deptId) {  
 this.deptId = deptId;  
 }  
  
 public String getLoginName() {  
 return loginName;  
 }  
  
 public void setLoginName(String loginName) {  
 this.loginName = loginName == null ? null : loginName.trim();  
 }  
  
 public String getUserName() {  
 return userName;  
 }  
  
 public void setUserName(String userName) {  
 this.userName = userName == null ? null : userName.trim();  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public void setEmail(String email) {  
 this.email = email == null ? null : email.trim();  
 }  
  
 public String getPhonenumber() {  
 return phonenumber;  
 }  
  
 public void setPhonenumber(String phonenumber) {  
 this.phonenumber = phonenumber == null ? null : phonenumber.trim();  
 }  
  
 public String getSex() {  
 return sex;  
 }  
  
 public void setSex(String sex) {  
 this.sex = sex == null ? null : sex.trim();  
 }  
  
 public String getAvatar() {  
 return avatar;  
 }  
  
 public void setAvatar(String avatar) {  
 this.avatar = avatar == null ? null : avatar.trim();  
 }  
  
 public String getPassword() {  
 return password;  
 }  
  
 public void setPassword(String password) {  
 this.password = password == null ? null : password.trim();  
 }  
  
 public String getSalt() {  
 return salt;  
 }  
  
 public void setSalt(String salt) {  
 this.salt = salt == null ? null : salt.trim();  
 }  
  
 public String getStatus() {  
 return status;  
 }  
  
 public void setStatus(String status) {  
 this.status = status == null ? null : status.trim();  
 }  
  
 public String getDelFlag() {  
 return delFlag;  
 }  
  
 public void setDelFlag(String delFlag) {  
 this.delFlag = delFlag == null ? null : delFlag.trim();  
 }  
}

### src/main/java/com/example/entity/UserRole.java

package com.example.entity;  
  
public class UserRole {  
 private Integer userId;  
  
 private Integer roleId;  
  
 public Integer getUserId() {  
 return userId;  
 }  
  
 public void setUserId(Integer userId) {  
 this.userId = userId;  
 }  
  
 public Integer getRoleId() {  
 return roleId;  
 }  
  
 public void setRoleId(Integer roleId) {  
 this.roleId = roleId;  
 }  
}

### src/main/java/com/example/dao/DeptMapper.java

package com.example.dao;  
  
import com.example.entity.Dept;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Param;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Mapper  
@Repository  
public interface DeptMapper {  
 int deleteByPrimaryKey(Integer deptId);  
  
 int insert(Dept record);  
  
 int insertSelective(Dept record);  
  
 Dept selectByPrimaryKey(Integer deptId);  
  
 int updateByPrimaryKeySelective(Dept record);  
  
 int updateByPrimaryKey(Dept record);  
  
 List<Dept> selectAllDept();  
  
 int delDeptByID( @Param("ids") List<Integer> ids);  
}

### src/main/java/com/example/dao/MenuMapper.java

package com.example.dao;  
  
import com.example.entity.Menu;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Param;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Mapper  
@Repository  
public interface MenuMapper {  
 int deleteByPrimaryKey(Integer menuId);  
  
 int insert(Menu record);  
  
 int insertSelective(Menu record);  
  
 Menu selectByPrimaryKey(Integer menuId);  
  
 int updateByPrimaryKeySelective(Menu record);  
  
 int updateByPrimaryKey(Menu record);  
  
 String selectPermsByMenuId(Integer menuId);  
  
 List<Menu> selectAllMenu();  
  
 List<Menu> selectMenuByUserLoginName(String loginName);  
  
 int delMenuByID( @Param("ids") List<Integer> ids);  
}

### src/main/java/com/example/dao/RoleMapper.java

package com.example.dao;  
  
import com.example.entity.Dept;  
import com.example.entity.Role;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Param;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Mapper  
@Repository  
public interface RoleMapper {  
 int deleteByPrimaryKey(Integer roleId);  
  
 int insert(Role record);  
  
 int insertSelective(Role record);  
  
 Role selectByPrimaryKey(Integer roleId);  
  
 int updateByPrimaryKeySelective(Role record);  
  
 int updateByPrimaryKey(Role record);  
  
 List<Role> selectAllRole();  
  
 int delRoleByID( @Param("ids") List<Integer> ids);  
  
 List<Role> selectAll();  
}

### src/main/java/com/example/dao/RoleMenuMapper.java

package com.example.dao;  
  
import com.example.entity.RoleMenu;  
import org.apache.ibatis.annotations.Mapper;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Mapper  
@Repository  
public interface RoleMenuMapper {  
 int deleteByPrimaryKey(RoleMenu key);  
  
 int insert(RoleMenu record);  
  
 int insertSelective(RoleMenu record);  
  
 List<Integer> selectMenuIdByRoleId(Integer roleId);  
}

### src/main/java/com/example/dao/UserMapper.java

package com.example.dao;  
  
import com.example.entity.User;  
import org.apache.ibatis.annotations.Mapper;  
import org.apache.ibatis.annotations.Param;  
import org.springframework.stereotype.Repository;  
  
import java.util.List;  
  
@Mapper  
@Repository  
public interface UserMapper {  
 int deleteByPrimaryKey(Integer userId);  
  
 int delUserByID( @Param("ids") List<String> ids);  
  
 int insert(User record);  
  
 int insertSelective(User record);  
  
 User selectByPrimaryKey(Integer userId);  
  
 User selectByLoginName(String loginName);  
  
 int updateByPrimaryKeySelective(User record);  
  
 int updateByPrimaryKey(User record);  
  
 List<User> selectAllUser();  
}

### src/main/java/com/example/dao/UserRoleMapper.java

package com.example.dao;  
  
import com.example.entity.UserRole;  
import org.apache.ibatis.annotations.Mapper;  
import org.springframework.stereotype.Repository;  
  
@Mapper  
@Repository  
public interface UserRoleMapper {  
 int deleteByPrimaryKey(UserRole key);  
  
 int insert(UserRole record);  
  
 int insertSelective(UserRole record);  
  
 Integer selectRoleIdByUserId(Integer userId);  
}

### src/main/java/com/example/controller/DeptController.java

package com.example.controller;  
  
import com.alibaba.fastjson.JSON;  
import com.example.biz.DeptBiz;  
import com.example.entity.Dept;  
import com.example.entity.MsJson;  
import com.example.util.MyConstants;  
import com.github.pagehelper.PageInfo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Controller  
@RequestMapping("/dept")  
public class DeptController {  
  
 @Autowired  
 private DeptBiz deptBizImpl;  
  
 //查询部门所有数据  
 @RequestMapping("/selectAll")  
 @ResponseBody  
 public MsJson selectAll(int page, int limit){  
 //开始查询  
 PageInfo<Dept> pageInfo = deptBizImpl.selectAllDept(page, limit);  
 MsJson json = new MsJson();  
 json.setCode(0);  
 json.setMsg("返回消息");  
 //设置分页之后的返回值  
 json.setCount(pageInfo.getTotal());  
 json.setData(pageInfo.getList());  
 return json;  
 }  
  
 //添加部门  
 @RequestMapping(value = "/addDept", method = RequestMethod.POST)  
 @ResponseBody  
 public Object addDept(Dept record){  
 Map map = (Map) deptBizImpl.insert(record);  
 return map;  
 }  
  
 //删除部门  
 @RequestMapping(value = "/deleteDept",method = RequestMethod.GET)  
 @ResponseBody  
 public Object deleteDept(Integer deptId){  
 Map map = (Map) deptBizImpl.deleteByPrimaryKey(deptId);  
 return map;  
 }  
  
 //批量删除部门  
 @RequestMapping(value = "/delDept",method = RequestMethod.POST)  
 @ResponseBody  
 public Object delDept( @RequestParam(value = "ids") String ids){  
 //将json字符串转换成list对象  
 List<Integer> list= (List<Integer>) JSON.parse(ids);  
 int i = deptBizImpl.delDeptByID(list);  
 Map map= new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 //编辑部门信息  
 @RequestMapping(value = "/editDept",method = RequestMethod.POST)  
 @ResponseBody  
 public Object editDept(Dept record){  
 Map map = (Map) deptBizImpl.updateByPrimaryKey(record);  
 return map;  
 }  
}

### src/main/java/com/example/controller/LoginController.java

package com.example.controller;  
  
import com.example.biz.MenuBiz;  
import com.example.biz.RoleBiz;  
import com.example.biz.RoleMenuBiz;  
import com.example.biz.UserRoleBiz;  
import com.example.entity.MsTree;  
import com.example.entity.Role;  
import org.apache.shiro.SecurityUtils;  
import org.apache.shiro.authc.IncorrectCredentialsException;  
import org.apache.shiro.authc.UnknownAccountException;  
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.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
  
import java.util.List;  
  
@Controller  
public class LoginController {  
  
 @Autowired  
 private MenuBiz menuBizImpl;  
  
 @RequestMapping(value = "/login", method = RequestMethod.POST)  
 public String login(String loginName, String password, Model model) {  
 //登录验证  
 //获取shiro的主体  
 Subject subject = SecurityUtils.getSubject();  
 //构建用户登录令牌  
 UsernamePasswordToken usernamePasswordToken =  
 new UsernamePasswordToken(loginName, password);  
  
 try {  
 subject.login(usernamePasswordToken);  
 } catch (UnknownAccountException e) {  
 model.addAttribute("message", "用户名错误");  
 return "/login";  
 }  
 catch (IncorrectCredentialsException e) {  
 model.addAttribute("message", "密码错误");  
 return "/login";  
 }  
  
 //将要去index页面之前，保存部分数据到model  
 model.addAttribute("loginName", loginName);  
 //放入所有的菜单，根据当前登录的用户  
 List<MsTree> menuList = menuBizImpl.selectMenuByUserLoginName(loginName);  
 model.addAttribute("menus", menuList);  
 return "/menu/index";  
 }  
  
  
 */\*\*  
 \* 注销  
 \*/* @RequestMapping("/logout")  
 public String logout(){  
 Subject subject = SecurityUtils.getSubject();  
 subject.logout();  
 return "/login";  
 }  
}

### src/main/java/com/example/controller/MenuController.java

package com.example.controller;  
  
import com.alibaba.fastjson.JSON;  
import com.example.biz.MenuBiz;  
import com.example.entity.Dept;  
import com.example.entity.Menu;  
import com.example.entity.MsJson;  
import com.example.entity.MsTree;  
import com.example.util.MyConstants;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Controller  
@RequestMapping("/menu")  
public class MenuController {  
  
 @Autowired  
 private MenuBiz menuBizImpl;  
  
 @RequestMapping("/selectAllMenu")  
 @ResponseBody  
 public List<MsTree> selectAllMenu(){  
 List<MsTree> layUiTrees = menuBizImpl.selectAllMenu();  
 return layUiTrees;  
 }  
  
 @RequestMapping("/selectAll")  
 @ResponseBody  
 public Object selectAll(){  
 List<Menu> menus = menuBizImpl.selectAll();  
 MsJson msJson = new MsJson();  
 msJson.setCode(0);  
 msJson.setMsg("");  
 msJson.setCount(menus.size());  
 msJson.setData(menus);  
 return msJson;  
 }  
  
 //添加菜单  
 @RequestMapping(value = "/addMenu", method = RequestMethod.POST)  
 @ResponseBody  
 public Object addDept(Menu record){  
 Map map = (Map) menuBizImpl.insert(record);  
 return map;  
 }  
  
 //批量删除菜单  
 @RequestMapping(value = "/delMenu",method = RequestMethod.POST)  
 @ResponseBody  
 public Object delDept( @RequestParam(value = "ids") String ids){  
 //将json字符串转换成list对象  
 List<Integer> list= (List<Integer>) JSON.parse(ids);  
 int i = menuBizImpl.delMenuByID(list);  
 Map map= new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 //删除部门  
 @RequestMapping(value = "/deleteMenu",method = RequestMethod.GET)  
 @ResponseBody  
 public Object deleteMenu(Integer menuId){  
 Map map = (Map) menuBizImpl.deleteByPrimaryKey(menuId);  
 return map;  
 }  
  
 //编辑菜单信息  
 @RequestMapping(value = "/editMenu",method = RequestMethod.POST)  
 @ResponseBody  
 public Object editDept(Menu record){  
 Map map = (Map) menuBizImpl.updateByPrimaryKey(record);  
 return map;  
 }  
}

### src/main/java/com/example/controller/PageController.java

package com.example.controller;  
  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
  
@Controller  
public class PageController {  
  
 //登录页面跳转  
 @RequestMapping(value = "/toLogin")  
 public String toLogin() {  
 return "/login";  
 }  
  
 //用户管理界面跳转  
 @RequestMapping("/toUser")  
 public String toUser(){  
 return "menu/userdb.html";  
 }  
  
 //菜单管理界面跳转  
 @RequestMapping("/toMenu")  
 public String toMenu(){  
 return "menu/menudb.html";  
 }  
  
 //角色管理界面跳转  
 @RequestMapping("/toRole")  
 public String toRole(){  
 return "menu/roledb.html";  
 }  
  
 //部门管理界面跳转  
 @RequestMapping("/toDept")  
 public String toDept(){  
 return "menu/deptdb.html";  
 }  
  
 //登录页面跳转  
 @RequestMapping(value = "/toUnau")  
 public String toUnau() {  
 return "menu/showMenuTree2";  
 }  
}

### src/main/java/com/example/controller/RoleController.java

package com.example.controller;  
  
import com.alibaba.fastjson.JSON;  
import com.example.biz.RoleBiz;  
import com.example.entity.\*;  
import com.example.util.MyConstants;  
import com.github.pagehelper.PageInfo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Controller  
@RequestMapping("/role")  
public class RoleController {  
  
 @Autowired  
 private RoleBiz roleBizImpl;  
  
 //查询角色所有数据  
 @RequestMapping("/selectAll")  
 @ResponseBody  
 public MsJson selectAll(int page, int limit){  
 //开始查询  
 PageInfo<Role> pageInfo = roleBizImpl.selectAllDept(page, limit);  
 MsJson json = new MsJson();  
 json.setCode(0);  
 json.setMsg("返回消息");  
 //设置分页之后的返回值  
 json.setCount(pageInfo.getTotal());  
 json.setData(pageInfo.getList());  
 return json;  
 }  
  
 //添加角色  
 @RequestMapping(value = "/addRole",method = RequestMethod.POST)  
 @ResponseBody  
 public Object addRole(Role role){  
 Map map = (Map)roleBizImpl.insert(role);  
 return map;  
 }  
  
 @RequestMapping(value = "/insertRoleMenu")  
 @ResponseBody  
 public Object insertRoleMenu(@RequestParam(value = "df") String df) {  
 System.out.println("权限");  
 List<MsTree> list = (List<MsTree>) JSON.parse(df);  
  
 System.out.println(list);  
  
 int i = 1;// = roleMenuBizImpl.insert(roleMenu);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.saveSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 //批量删除角色  
 @RequestMapping(value = "/delRole",method = RequestMethod.POST)  
 @ResponseBody  
 public Object delRole( @RequestParam(value = "ids") String ids){  
 //将json字符串转换成list对象  
 List<Integer> list= (List<Integer>) JSON.parse(ids);  
 int i = roleBizImpl.delRoleByID(list);  
 Map map= new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 //编辑角色信息  
 @RequestMapping(value = "/editRole",method = RequestMethod.POST)  
 @ResponseBody  
 public Object editDept(Role record){  
 System.out.println(record);  
 Map map = (Map) roleBizImpl.updateByPrimaryKey(record);  
 return map;  
 }  
  
 //删除角色  
 @RequestMapping(value = "/deleteRole",method = RequestMethod.GET)  
 @ResponseBody  
 public Object deleteDept(Integer roleId){  
 Map map = (Map) roleBizImpl.deleteByPrimaryKey(roleId);  
 return map;  
 }  
}

### src/main/java/com/example/controller/UserController.java

package com.example.controller;  
  
import com.alibaba.fastjson.JSON;  
import com.example.biz.RoleBiz;  
import com.example.biz.UserBiz;  
import com.example.biz.UserRoleBiz;  
import com.example.entity.MsJson;  
import com.example.entity.Role;  
import com.example.entity.User;  
import com.example.entity.UserRole;  
import com.example.shiro.ShiroUtil;  
import com.example.util.MyConstants;  
import com.github.pagehelper.PageHelper;  
import com.github.pagehelper.PageInfo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Controller;  
import org.springframework.stereotype.Repository;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestMethod;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.ResponseBody;  
  
import java.util.\*;  
  
@Controller  
@RequestMapping("/user")//窄化请求  
public class UserController {  
  
 @Autowired  
 private UserBiz userBizImpl;  
  
 @Autowired  
 private RoleBiz roleBizImpl;  
  
 @Autowired  
 private UserRoleBiz userRoleBizImpl;  
  
 @RequestMapping("/selectAllUser")  
 @ResponseBody  
 public Object selectAllUser(int page, int limit){  
 //开始分页  
 PageHelper.startPage(page, limit);  
 //开始查询  
 //(开始分页和开始查询的代码之间不能有其他任何不相关的代码)  
 List<User> userList = userBizImpl.selectAllUser();  
 //结束分页  
 PageInfo<User> pageInfo = new PageInfo<>(userList);  
  
 MsJson json = new MsJson();  
 json.setCode(0);  
 json.setMsg("返回消息");  
 json.setCount(pageInfo.getTotal());  
 json.setData(pageInfo.getList());  
 return json;  
 }  
  
 @RequestMapping(value = "/insertUser")  
 @ResponseBody  
 public Object insertUser(User user){  
 String salt = UUID.randomUUID().toString();  
 String password = ShiroUtil.encryptionBySalt(salt, user.getPassword());  
 user.setPassword(password);  
 user.setSalt(salt);  
 int i = userBizImpl.insertSelective(user);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.saveSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 @RequestMapping(value = "/updateUser")  
 @ResponseBody  
 public Object updateUser(User user){  
 int i = userBizImpl.updateByPrimaryKeySelective(user);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.editSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.editFailMsg);  
 }  
 return map;  
 }  
  
 @RequestMapping(value = "/deleteUser")  
 @ResponseBody  
 public Object deleteUser(@RequestParam(value = "ids") String ids){  
 //将json字符串转换成list对象  
 List<String> list = (List<String>) JSON.parse(ids);  
 int i = userBizImpl.delUserByID(list);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.delSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 @RequestMapping(value = "/resetPassword")  
 @ResponseBody  
 public Object resetPassword(int userId, String newPassword) {  
 String salt = UUID.randomUUID().toString();  
 String password = ShiroUtil.encryptionBySalt(salt, newPassword);  
 User user = new User();  
 user.setUserId(userId);  
 user.setPassword(password);  
 user.setSalt(salt);  
 int i = userBizImpl.updateByPrimaryKeySelective(user);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.editSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.editFailMsg);  
 }  
 return map;  
 }  
  
 @RequestMapping(value = "/getRole")  
 @ResponseBody  
 public Object getRole() {  
 List<Role> roleList = roleBizImpl.selectAll();  
 MsJson json = new MsJson();  
 json.setCode(0);  
 json.setMsg("角色下拉框");  
 json.setCount(roleList.size());  
 json.setData(roleList);  
 return json;  
 }  
  
 @RequestMapping(value = "/setRole")  
 @ResponseBody  
 public Object setRole(UserRole userRole) {  
 int i = userRoleBizImpl.insert(userRole);  
 Map map = new HashMap<>();  
 if (i > 0) {  
 map.put("code", MyConstants.successCode);  
 map.put("message", MyConstants.saveSuccessMsg);  
 } else {  
 map.put("code", MyConstants.failCode);  
 map.put("message", MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 @RequestMapping(value = "/selectByLoginName",method = RequestMethod.GET)  
 @ResponseBody  
 public Object selectByLoginName(String loginName) {  
 User user = userBizImpl.selectByLoginName(loginName);  
 List<User> userList = new ArrayList<>();  
 userList.add(user);  
 MsJson json = new MsJson();  
 json.setCode(0);  
 json.setMsg("返回消息");  
 json.setCount(100);  
 json.setData(userList);  
 return json;  
 }  
  
 //查看用户民是否存在  
 @RequestMapping(value = "/checkUserName",method = RequestMethod.POST)  
 @ResponseBody  
 public Boolean checkUserName(String loginName){  
 return userBizImpl.checkUserLoginName(loginName);  
 }  
}

### src/main/java/com/example/biz/DeptBiz.java

package com.example.biz;  
  
import com.example.entity.Dept;  
import com.github.pagehelper.PageInfo;  
import org.apache.ibatis.annotations.Param;  
  
import java.util.List;  
  
  
public interface DeptBiz {  
 PageInfo<Dept> selectAllDept(int page, int limit);  
 Object insert(Dept record);  
 Object deleteByPrimaryKey(Integer deptId);  
 int delDeptByID( @Param("ids") List<Integer> ids);  
 Object updateByPrimaryKey(Dept record);  
}

### src/main/java/com/example/biz/MenuBiz.java

package com.example.biz;  
  
import com.example.entity.Dept;  
import com.example.entity.Menu;  
import com.example.entity.MsTree;  
  
import java.util.List;  
  
public interface MenuBiz {  
  
 String selectPermsByMenuId(Integer menuId);  
  
 List<MsTree> selectAllMenu();  
  
 List<MsTree> selectMenuByUserLoginName(String loginName);  
  
 List<Menu> selectAll();  
  
 Object insert(Menu record);  
  
 int delMenuByID(List<Integer> ids);  
  
 Object deleteByPrimaryKey(Integer menuId);  
  
 Object updateByPrimaryKey(Menu record);  
}

### src/main/java/com/example/biz/RoleBiz.java

package com.example.biz;  
  
import com.example.entity.Dept;  
import com.example.entity.Role;  
import com.github.pagehelper.PageInfo;  
  
import java.util.List;  
  
public interface RoleBiz {  
  
 Role selectByPrimaryKey(Integer roleId);  
 PageInfo<Role> selectAllDept(int page, int limit);  
 Object insert(Role record);  
 int delRoleByID(List<Integer> ids);  
 Object updateByPrimaryKey(Role record);  
 Object deleteByPrimaryKey(Integer deptId);  
 List<Role> selectAll();  
}

### src/main/java/com/example/biz/RoleMenuBiz.java

package com.example.biz;  
  
import com.example.entity.RoleMenu;  
  
import java.util.List;  
  
public interface RoleMenuBiz {  
  
 List<Integer> selectMenuIdByRoleId(Integer roleId);  
}

### src/main/java/com/example/biz/UserBiz.java

package com.example.biz;  
  
import com.example.entity.User;  
  
import java.util.List;  
  
public interface UserBiz {  
  
 User selectByPrimaryKey(Integer userId);  
  
 User selectByLoginName(String loginName);  
  
 List<User> selectAllUser();  
  
 int insert(User record);  
  
 int insertSelective(User record);  
  
 int deleteByPrimaryKey(Integer userId);  
  
 int delUserByID(List<String> ids);  
  
 int updateByPrimaryKey(User record);  
  
 int updateByPrimaryKeySelective(User record);  
  
 boolean checkUserLoginName(String loginName);  
}

### src/main/java/com/example/biz/UserRoleBiz.java

package com.example.biz;  
  
import com.example.entity.UserRole;  
  
public interface UserRoleBiz {  
  
 Integer selectRoleIdByUserId(Integer userId);  
  
 int insert(UserRole record);  
}

### src/main/java/com/example/biz/impl/UserRoleBizImpl.java

package com.example.biz.impl;  
  
import com.example.biz.UserRoleBiz;  
import com.example.dao.UserRoleMapper;  
import com.example.entity.UserRole;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
@Service  
public class UserRoleBizImpl implements UserRoleBiz {  
  
 @Autowired  
 private UserRoleMapper userRoleMapper;  
  
 @Override  
 public Integer selectRoleIdByUserId(Integer userId) {  
 return userRoleMapper.selectRoleIdByUserId(userId);  
 }  
  
 @Override  
 public int insert(UserRole record) {  
 return userRoleMapper.insert(record);  
 }  
}

### src/main/java/com/example/biz/impl/UserBizImpl.java

package com.example.biz.impl;  
  
import com.example.biz.UserBiz;  
import com.example.dao.UserMapper;  
import com.example.entity.User;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class UserBizImpl implements UserBiz {  
  
 @Autowired  
 private UserMapper userMapper;  
  
 @Override  
 public User selectByPrimaryKey(Integer userId) {  
 return userMapper.selectByPrimaryKey(userId);  
 }  
  
 @Override  
 public User selectByLoginName(String loginName) {  
 return userMapper.selectByLoginName(loginName);  
 }  
  
 @Override  
 public List<User> selectAllUser() {  
 return userMapper.selectAllUser();  
 }  
  
 @Override  
 public int insert(User record) {  
 return userMapper.insert(record);  
 }  
  
 @Override  
 public int insertSelective(User record) {  
 return userMapper.insertSelective(record);  
 }  
  
 @Override  
 public int deleteByPrimaryKey(Integer userId) {  
 return userMapper.deleteByPrimaryKey(userId);  
 }  
  
 @Override  
 public int delUserByID(List<String> ids) {  
 return userMapper.delUserByID(ids);  
 }  
  
 @Override  
 public int updateByPrimaryKey(User record) {  
 return userMapper.updateByPrimaryKey(record);  
 }  
  
 @Override  
 public int updateByPrimaryKeySelective(User record) {  
 return userMapper.updateByPrimaryKeySelective(record);  
 }  
  
 @Override  
 public boolean checkUserLoginName(String loginName) {  
 User user = userMapper.selectByLoginName(loginName);  
 System.out.println(user);  
 if(user == null){  
 return true;  
 }else  
 return false;  
 }  
}

### src/main/java/com/example/biz/impl/RoleMenuBizImpl.java

package com.example.biz.impl;  
  
import com.example.biz.RoleMenuBiz;  
import com.example.dao.RoleMenuMapper;  
import com.example.entity.RoleMenu;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.List;  
  
@Service  
public class RoleMenuBizImpl implements RoleMenuBiz {  
  
 @Autowired  
 private RoleMenuMapper roleMenuMapper;  
  
 @Override  
 public List<Integer> selectMenuIdByRoleId(Integer roleId) {  
 return roleMenuMapper.selectMenuIdByRoleId(roleId);  
 }  
}

### src/main/java/com/example/biz/impl/RoleBizImpl.java

package com.example.biz.impl;  
  
import com.example.biz.RoleBiz;  
import com.example.dao.RoleMapper;  
import com.example.entity.Dept;  
import com.example.entity.Role;  
import com.example.util.MyConstants;  
import com.github.pagehelper.PageHelper;  
import com.github.pagehelper.PageInfo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Service  
public class RoleBizImpl implements RoleBiz {  
  
 @Autowired  
 private RoleMapper roleMapper;  
  
  
 @Override  
 public Role selectByPrimaryKey(Integer roleId) {  
 return roleMapper.selectByPrimaryKey(roleId);  
 }  
  
 @Override  
 public PageInfo<Role> selectAllDept(int page, int limit) {  
 //开始分页,第一个参数是当前第几页，第二个参数是一页显示多少行  
 PageHelper.startPage(page,limit);  
 List<Role> myUserInfos = roleMapper.selectAllRole();  
 //结束分页,pageInfo封装了分页之后所有数据  
 PageInfo<Role> pageInfo = new PageInfo(myUserInfos);  
 return pageInfo;  
 }  
  
 @Override  
 public Object insert(Role record) {  
 int i = roleMapper.insert(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.saveSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public int delRoleByID(List<Integer> ids) {  
 return roleMapper.delRoleByID(ids);  
 }  
  
 @Override  
 public Object updateByPrimaryKey(Role record) {  
 int i = roleMapper.updateByPrimaryKey(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.editSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.editFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public Object deleteByPrimaryKey(Integer roleId) {  
 int i = roleMapper.deleteByPrimaryKey(roleId);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public List<Role> selectAll() {  
 return roleMapper.selectAll();  
 }  
}

### src/main/java/com/example/biz/impl/MenuBizImpl.java

package com.example.biz.impl;  
  
import com.example.Util.TreeUtils;  
import com.example.biz.MenuBiz;  
import com.example.dao.MenuMapper;  
import com.example.entity.Menu;  
import com.example.entity.MsTree;  
import com.example.util.MyConstants;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Service  
public class MenuBizImpl implements MenuBiz {  
  
 @Autowired  
 private MenuMapper menuMapper;  
  
 @Override  
 public String selectPermsByMenuId(Integer menuId) {  
 return menuMapper.selectPermsByMenuId(menuId);  
 }  
  
 //查询所有的菜单，组装成tree格式  
 @Override  
 public List<MsTree> selectAllMenu() {  
 //查询所有的菜单  
 List<Menu> menuList = menuMapper.selectAllMenu();  
 //组装成tree格式  
 return TreeUtils.getChildPerms(menuList, 0);  
 }  
  
 //根据用户登录名查询对应的所有菜单  
 @Override  
 public List<MsTree> selectMenuByUserLoginName(String loginName) {  
 //根据用户loginName查询所有的菜单  
 List<Menu> menuList = menuMapper.selectMenuByUserLoginName(loginName);  
 //组装成tree格式  
 return TreeUtils.getChildPerms(menuList, 0);  
 }  
  
 @Override  
 public List<Menu> selectAll() {  
 return menuMapper.selectAllMenu();  
 }  
  
 @Override  
 public Object insert(Menu record) {  
 int i = menuMapper.insert(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.saveSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public int delMenuByID(List<Integer> ids) {  
 return menuMapper.delMenuByID(ids);  
 }  
  
 @Override  
 public Object deleteByPrimaryKey(Integer menuId) {  
 int i = menuMapper.deleteByPrimaryKey(menuId);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public Object updateByPrimaryKey(Menu record) {  
 int i = menuMapper.updateByPrimaryKey(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.editSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.editFailMsg);  
 }  
 return map;  
 }  
}

### src/main/java/com/example/biz/impl/DeptBizImpl.java

package com.example.biz.impl;  
  
import com.example.biz.DeptBiz;  
import com.example.dao.DeptMapper;  
import com.example.entity.Dept;  
import com.example.util.MyConstants;  
import com.github.pagehelper.PageHelper;  
import com.github.pagehelper.PageInfo;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
  
@Service  
public class DeptBIzImpl implements DeptBiz {  
  
 @Autowired  
 private DeptMapper deptMapper;  
  
  
 @Override  
 public PageInfo<Dept> selectAllDept(int page, int limit) {  
 //开始分页,第一个参数是当前第几页，第二个参数是一页显示多少行  
 PageHelper.startPage(page,limit);  
 List<Dept> myUserInfos = deptMapper.selectAllDept();  
 //结束分页,pageInfo封装了分页之后所有数据  
 PageInfo<Dept> pageInfo = new PageInfo(myUserInfos);  
 return pageInfo;  
 }  
  
 @Override  
 public Object insert(Dept record) {  
 int i = deptMapper.insert(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.saveSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.saveFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public Object deleteByPrimaryKey(Integer deptId) {  
 int i = deptMapper.deleteByPrimaryKey(deptId);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.delSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.delFailMsg);  
 }  
 return map;  
 }  
  
 @Override  
 public int delDeptByID(List<Integer> ids) {  
 return deptMapper.delDeptByID(ids);  
 }  
  
 @Override  
 public Object updateByPrimaryKey(Dept record) {  
 int i = deptMapper.updateByPrimaryKey(record);  
 Map map = new HashMap<>();  
 if(i>0){  
 map.put("code", MyConstants.successCode);  
 map.put("message",MyConstants.editSuccessMsg);  
 }else {  
 map.put("code",MyConstants.failCode);  
 map.put("message",MyConstants.editFailMsg);  
 }  
 return map;  
 }  
}