5张表+shiro的权限实现；

用户权限模型1：---5

资源；

角色；

用户；

角色资源表；

用户角色表；

~

此中，一个用户一个角色；| 简单实现；

/////////////////////////////////////////////////////////////////////////////////////////////////////

表：

资源表：---无限级；自外键；

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| resource\_id | bigint | 自增主键 | 自增、无符号、not null | 主键 not null |
| sup\_resource\_id | bigint | 父类id | s=superior | 自外键 |
| res\_name | varchar(50) | 资源名 |  | not null |
| res\_action | varchar(100) | 动作 |  |  |
| res\_url | varchar(100) | url |  |  |
| res\_code | varchar(20) | 代码 |  |  |
| res\_sequence | int | 排序 | 默认0 | not null |
| res\_remarks | varchar(200) | 备注 |  |  |

~

模块，对应的action、url为空；

角色表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| role\_id | bigint |  |  | 主键 not null |
| role\_name | varchar(50) | 角色名 |  | not null |
| role\_status | Tinyint | 状态 | 0关闭1正常 | not null |
| role\_sequence | int | 排序 |  | not null |
| role\_remarks | varchar(200) | 备注 |  |  |

用户表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| user\_id | bigint |  |  | 主键 not null |
| area\_id | bigint | 区域 | 暂时清除外键，给定默认值0； | 外键 not null |
| dept\_id | bigint | 部门id | 暂时清除外键，给定默认值0； | 外键 not null |
| Jobs\_id | bigint | 岗位id | 暂时清除外键，给定默认值0； | 外键 not null |
| user\_name | varchar(50) | 用户名 |  | not null |
| user\_pwd | varchar(50) | 密码 |  | not null |
| user\_addTime | datetime | 建立时间 |  | not null |
| user\_updateTime | datetime | 修改时间 |  | not null |
| user\_status | Tinyint | 状态 | 0锁定1正常2注销 | not null |
|  |  | 真实姓名 |  |  |
|  |  | 身份证号 |  |  |
|  |  | 是否实名认证 |  |  |
|  |  | 邮箱 |  |  |
|  |  | 是否邮箱认证 |  |  |
|  |  | 手机号 |  |  |
|  |  | 是否手机认证 |  |  |
| user\_remarks | varchar(200) | 备注 |  |  |

角色-资源表：t\_sys\_role\_resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| role\_resource\_id | bigint |  |  | 主键 not null |
| role\_id | bigint | 角色id |  | 外键 not null |
| resource\_id | bigint | 资源id |  | 外键 not null |

用户-角色表：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| user\_role\_id | bigint |  |  | 主键 not null |
| user\_id | bigint | 用户id |  | 外键 not null |
| role\_id | bigint | 角色id |  | 外键 not null |

////////////////////////////////////////////////////////////////////

实现：

1，配置：

Web.xml；

添加过滤器，

Spring配置：写一个专门的配置文件,spring-shiro.xml；

需验证资源的配置；添加securityManager安全管理

<!—缓存EhCache --> | *ehcache-shiro.xml文件，*

2，realm类，---认证、授权处理；

public class ShiroRealm extends AuthorizingRealm{

//业务逻辑类，获取用户信息方法，

@Autowired

**private** UsersService usersService;

//获取认证信息---登录时，

protected AuthenticationInfo doGetAuthenticationInfo(

AuthenticationToken token) throws AuthenticationException {

UsernamePasswordToken usernamePasswordToken = (UsernamePasswordToken) token;

//通过表单获取的用户名，

String username = usernamePasswordToken.getUsername();

//验证用户信息；

根据用//验证用户信息；

根据用户名密码，数据库查询；

**if** (user== **null**) {

**throw** **new** ErrorAccountException("用户名或密码错误");

}

return new SimpleAuthenticationInfo(

user,getPassword(),getName() );

户名密码，数据库查询；

**if** (user== **null**) {

**throw** **new** ErrorAccountException("用户名或密码错误");

}

return new SimpleAuthenticationInfo(

user,getPassword(),getName() );

}

return null;

//获取授权信息 ---访问页面，action时；

protected AuthorizationInfo doGetAuthorizationInfo(

PrincipalCollection principals) {

String username = (String) principals.fromRealm(

getName()).iterator().next();

//将用户许可信息放入AuthorizationInfo中；

if( username != null ){

// 查询用户授权信息

Collection<String> pers= pubUsersService.queryPermissions(username);

if( pers != null && !pers.isEmpty() ){

SimpleAuthorizationInfo info = new SimpleAuthorizationInfo();

for( String each:pers )

info.addStringPermissions( each );

return info;

}

}

return null;

3，业务类中的使用；

3.1登录中，

//shiro登录，

UsernamePasswordToken token = new UsernamePasswordToken("lonestarr", "vespa");

token.setRememberMe(true);

Subject currentUser= SecurityUtils.getSubject(); //获得当前主体；

try {

currentUser.login(token); //认证操作；---调用执行realm类，

} catch (UnknownAccountException uae) {

log.info("用户名不存在 " + token.getPrincipal());

} catch (IncorrectCredentialsException ice) {

log.info("密码错误 " + token.getPrincipal() );

} catch (LockedAccountException lae) {

log.info("帐号锁定 " + token.getPrincipal() );

}

登录成功时，跳转到用户首页；

3.2其它业务类中，

获取登录信息，

Subject currentUser= SecurityUtils.getSubject();

4，需要判断登录的页面action中：

在spring-shiro.xml中配置；或类方法前加验证注解；

注解，

@RequiresAuthentication

4.2页面中判断角色，权限：

5，登出：

//删除用户身份信息，无效session；

currentUser.logout();