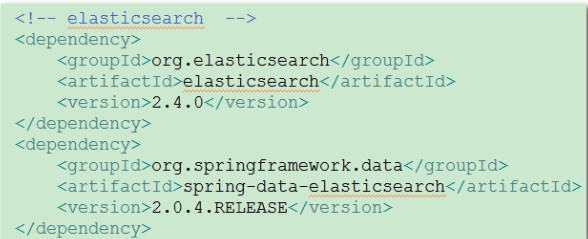
# 主要讲解点：

1. 运单录入功能运用SpringDataElasticSearch
2. 运单查询功能运用SpringDataElasticSearch（重点）
3. ApacheShiro权限框架（重点）

# 运单录入建立索引(实体类配置索引和映射信息)

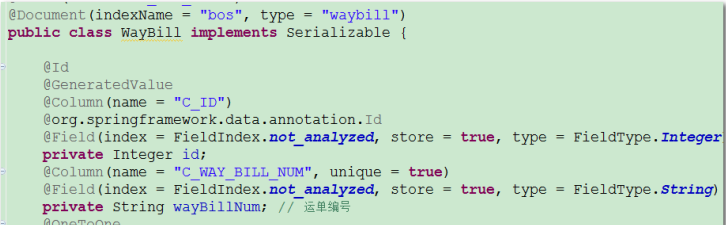
1. 在bos\_management引入elasticsearch和spring data elasticsearch支持



注意：提升spring依赖版本4.1.7到4.2.8

2、需要在实体类WayBill对象，添加elasticsearch索引和映射信息

操作bos\_domain项目



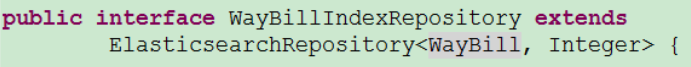
# 二、运单录入建立索引(配置Elasticsearch、编写DAO)

3、配置elasticsearch

applicationContext-elasticsearch.xml



1. 操作索引库，创建DAO



1. 修改Service代码



1. Bug修复：空指针异常



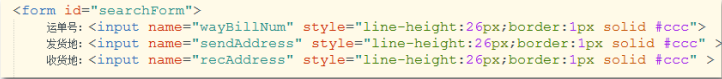
# 三、运单管理基于索引库查询（前端代码）



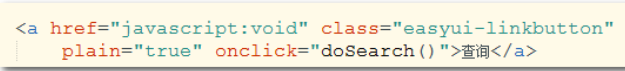
通过索引库，完成查询

1、 将查询form的数据，转换json格式，绑定数据表格上

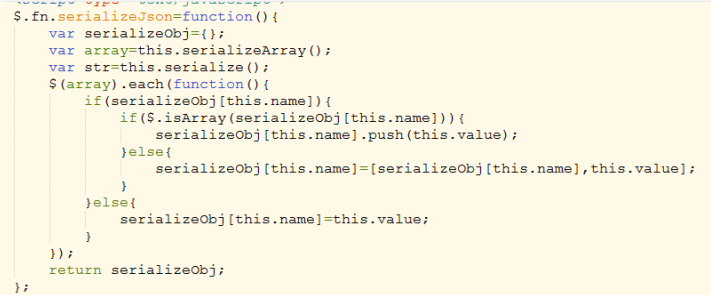
1）对查询内容，添加form元素

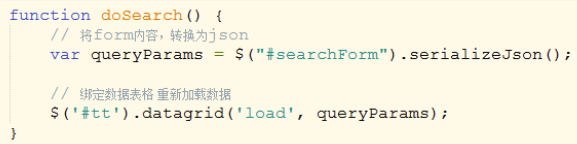


2）点击查询按钮，执行doSearch方法



3）将form数据转换成json，绑定到datagrid上

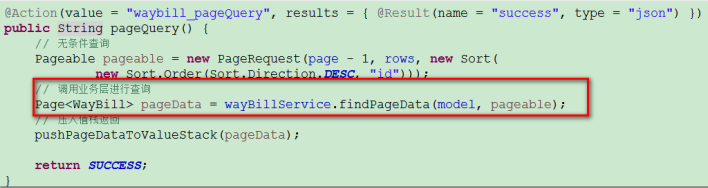




# 运单管理基于索引库查询（服务器实现单词条匹配）

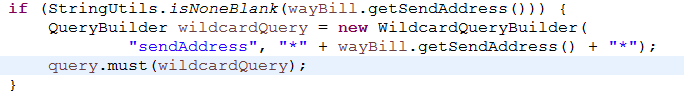
1. 修改WayBillAction的pageQuery方法

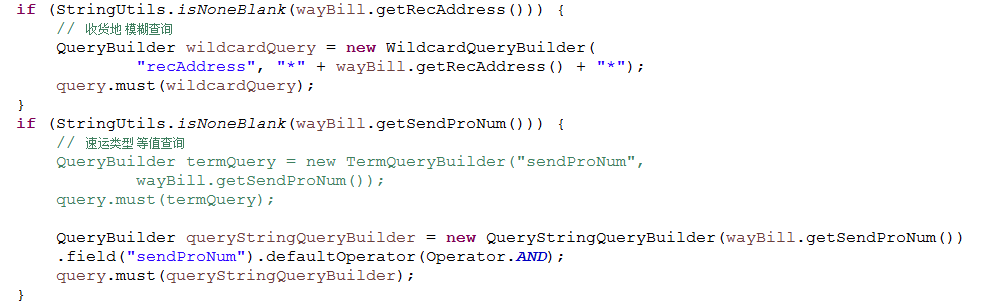


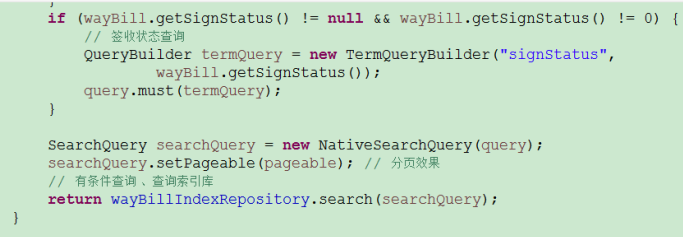


3、编写业务层findPageData方法









# 运单管理基于索引库查询（服务器实现多词条分词匹配）

1. 运单搜索，基于索引库词条完成搜索



2、添加多个词条组合，分词查询效果代码



# 第九章主要内容概述

1、权限控制的两种方式：粗粒度基于URL级别权限控制、细粒度基于方法级别权限控制

2、基于Apache Shiro实现登录认证和权限控制，重点讲解shiro权限控制流程、自定义Realm对象控制系统认证和授权

3、Apache Shiro实现细粒度方法级别权限控制

4、动态系统菜单显示功能

5、对认证和授权数据进行缓存优化

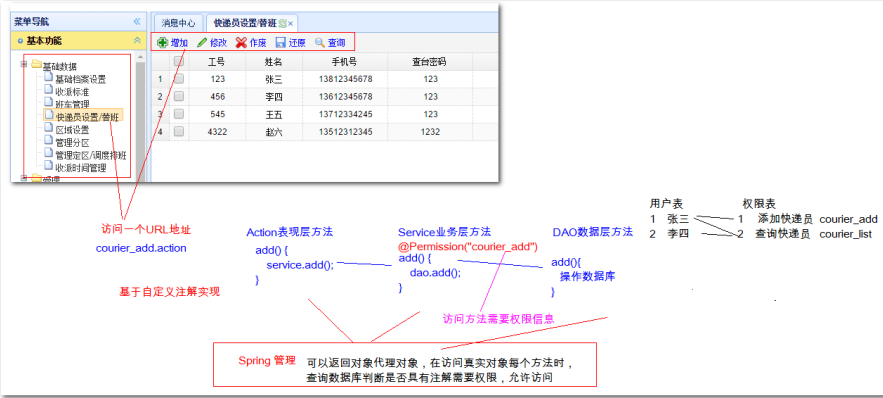
# 权限控制方式\_粗粒度URL级别权限控制



可以基于Filter实现

在数据库中存放用户、权限、访问URL对应关系，当前用户访问一个URL地址，查询数据库判断用户当前具有权限，是否包含这个URL，如果包含允许访问，如果不包含则权限不足

# 八、权限控制方式\_细粒度方法级别权限控制



可以代理、自定义注解实现，访问目标对象方法，在方法上添加权限注解信息，对目标对象创建代理对象，访问真实对象先访问代理对象，在代理对象查询数据库判断是否具有注解上描述需要权限，具有权限允许访问，不具有权限，拦截访问，提示权限不足

# 九、权限相关数据表分析和创建

1、准备工作

1）用户（User）：系统登录用户

2）权限（Permission）：描述权限信息（粗粒度权限控制，在权限表描述访问资源URL信息）

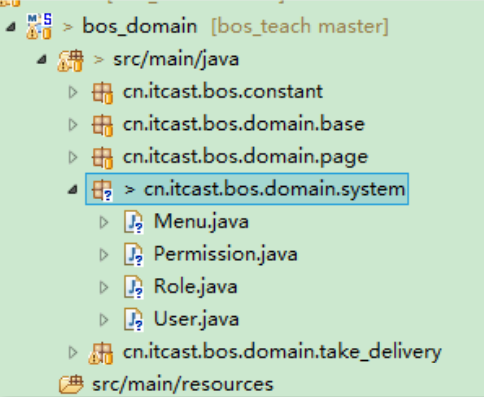
3）角色（Role）：方便用户进行授权，角色就是权限的集合

用户 \*---\* 角色 \*---\* 权限（5张表）

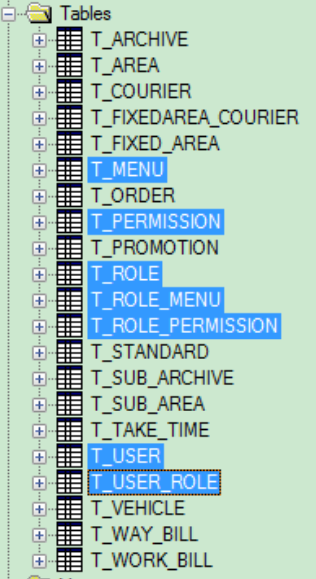
4）Menu菜单，为了方便进行动态菜单管理，为不同用户定制不同系统菜单

不同用户系统菜单，可以根据用户角色进行管理 角色\*---\*菜单（2张表）

1. 具体操作
2. 在bos\_domain建立cn.itcast.bos.domain.system包
3. 将实体类复制到包中，启动项目自动建表



3）对bos\_domain进行maven install



# 十、ApacheShiro简介和项目导入

1、实现权限控制，可以自己写代码实现（粗粒度权限控制：Filter、细粒度权限控制：自定义注解、代理、反射技术）

1. 自己写权限框架
2. Spring Security（使用复杂，依赖Spring）
3. Apache Shiro（更轻量，使用更简单，并不完全依赖spring，可以独立使用）

2、Apache Shiro介绍

官网：<http://shiro.apache.org/>

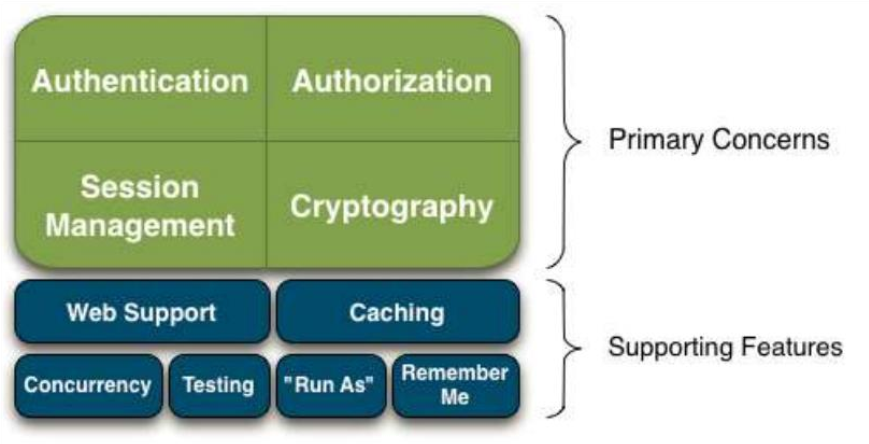
1. Apache Shiro体系结构

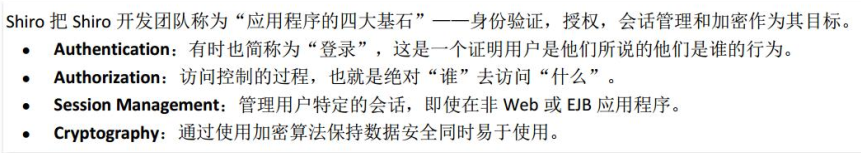


1. Authentication认证：用户登录，身份识别 who are you?
2. Authorization 授权：用户具有哪些权限、角色 what can you do?
3. Cryptography安全数据加密
4. Session Management 会话管理
5. Web Integration web系统集成

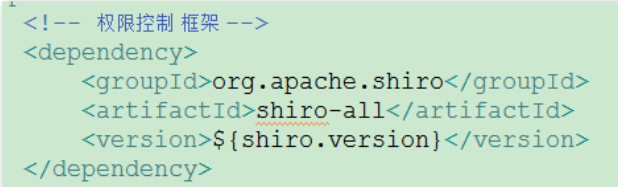
6）Integrations 集成其他应用，spring、缓存框架

4、参考Apache\_Shiro\_reference(中文版).pdf

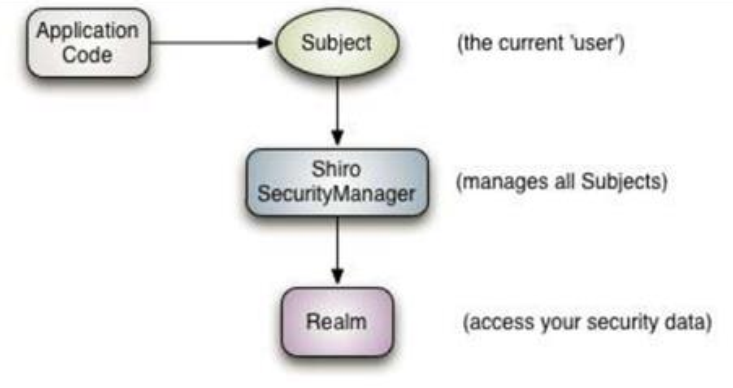




5、项目中pom引入



# 十一、ApacheShiro运行流程和权限控制方式分析



1、核心介绍

1）Application Code用户编写代码

2）Subject就是shiro管理的用户

3）SecurityManager安全管理器，就是shiro权限控制核心对象，在编程时，只需要操作Subject方法，底层调用SecurityManager方法，无需直接编程操作SecurityManager

4）Realm应用程序和安全数据之间连接器，应用程序进行权限控制读取安全数据（数据表、文件、网络…）通过Realm对象完成

2、Shiro执行流程

应用程序--->Subject--->SecurityManager--->Realm--->安全数据

3、Shiro进行权限控制的四种主要方式

1）在程序中通过Subject编程方式进行权限控制

2）配置Filter实现URL级别粗粒度权限控制

3）配置代理，基于注解实现细粒度权限控制

4）在页面中使用shiro自定义标签实现，页面显示权限控制

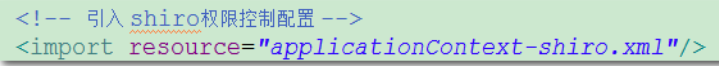
# 十二、配置ShiroFilter实现URL级别权限控制

1. 配置web.xml



2、配置applicationContext-shiro.xml

第一步：applicationContext.xml



第二步：applicationContext-shiro.xml

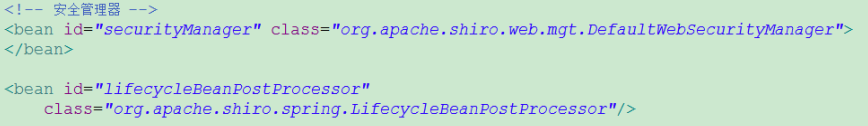


/login.html后面加上\*是因为会有如下情况：

<http://localhost:9001/bos_management/login.html;jsessionid=68A360BFE7413C7CEAE81E5069F1EE81>

/\*\*=authc放在最后

第三步：配置安全管理器和Shiro生命周期处理器LifecycleBeanPostProcessor



LifecycleBeanPostProcessor用于在实现了Initializable接口的Shiro bean初始化时调用Initializable接口回调，在实现了Destroyable接口的Shiro bean销毁时调用 Destroyable接口回调。

3、Shiro过滤器配置参考



anon未认证可以访问

authc认证后可以访问

perms需要特定权限才能访问

roles需要特定角色才能访问

user需要特定用户才能访问

port需要特定端口才能访问（不常用）

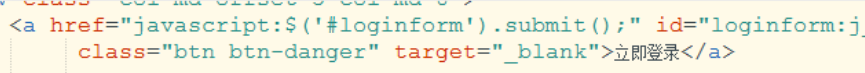
rest根据指定HTTP请求才能访问（不常用）

# 十三、自定义Realm实现用户登录功能（上）

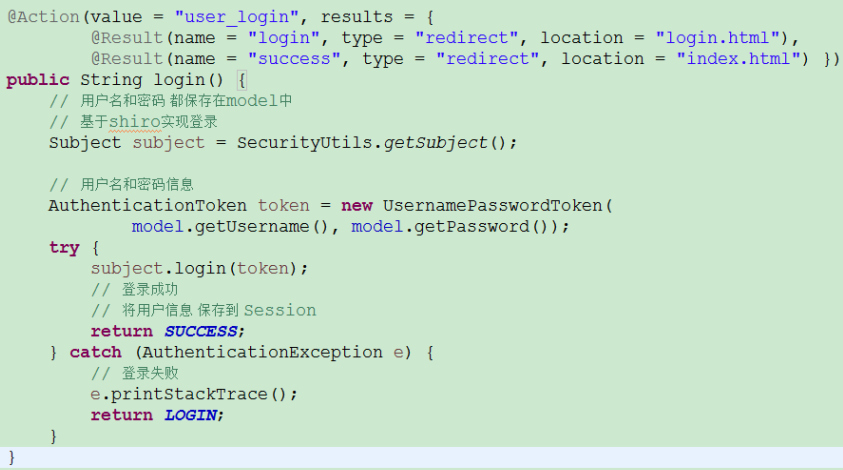
1. 检查login.html表单元素



点击立即登录，提交form表单

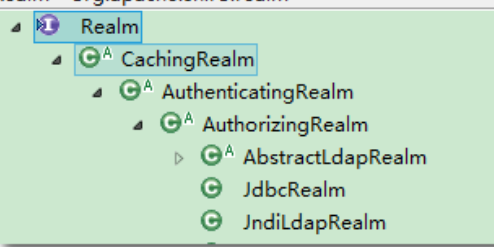


2、编写UserAction提供login登录方法

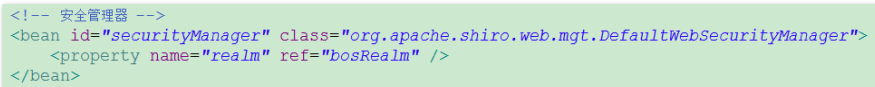


Shiro执行流程：应用程序--->Subject--->SecurityManager--->Realm--->安全数据

3、自定义Realm对象，实现认证方法（实际开发中，只需要继承AuthorizingRealm）

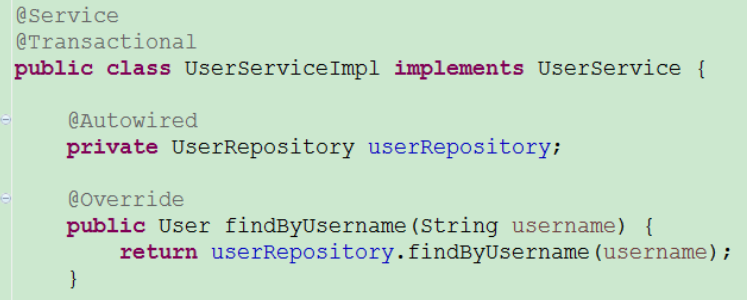


将自定义Realm注入安全管理器SecurityManager当中

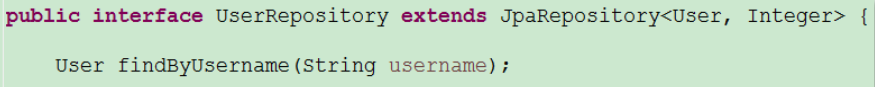


# 十四、自定义Realm实现用户登录功能（下）

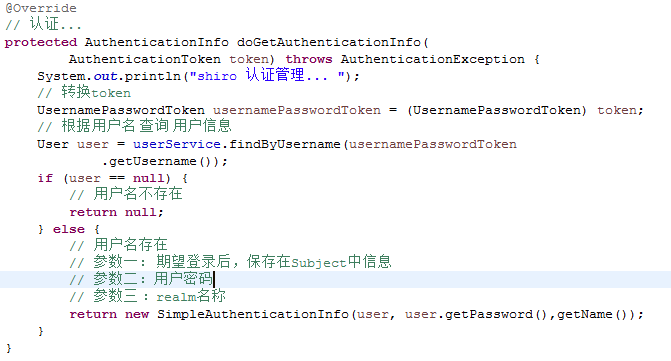
1）Service代码



2）DAO代码



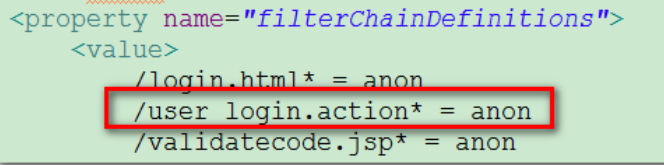
1. Realm



4、在user表插入一些数据



5、在shiroFilter配置中将user\_login.action放行



6、注意问题

1）当用户名不存在时，抛出异常



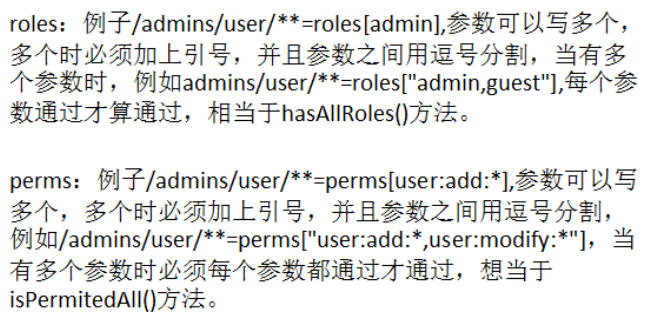
2）当密码错误时，抛出异常



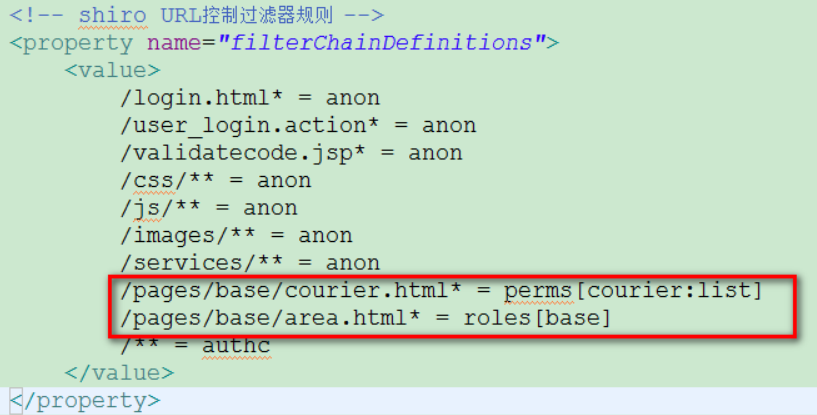
# 十五、对页面访问添加授权控制

1. 用户授权功能主要是解决用户能操作哪些功能的问题
2. 参考





1. 修改applicationContext-shiro.xml配置shiroFilter权限过滤程序



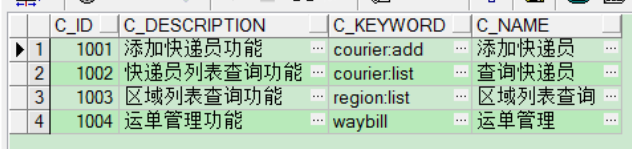
1. 访问courier.html时，因为需要特别权限，调用自定义Realm的doGetAuthorizationInfo进行授权，如果没有权限，跳转



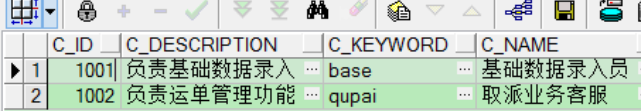
# 十六、自定义Realm实现用户授权功能（上）

1、在数据表中导入预定义角色、权限数据，导入t\_user\_role\_permission.sql

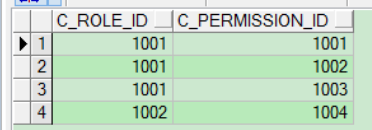
1）t\_permission权限数据



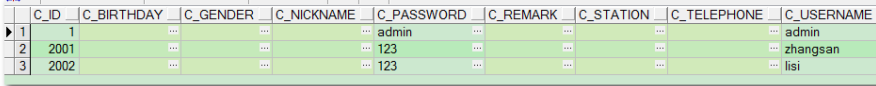
2）t\_role角色数据



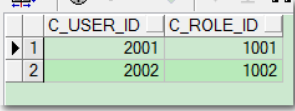
3）t\_role\_permission角色权限关系表



4）t\_user用户表



5）t\_user\_role用户角色关系表



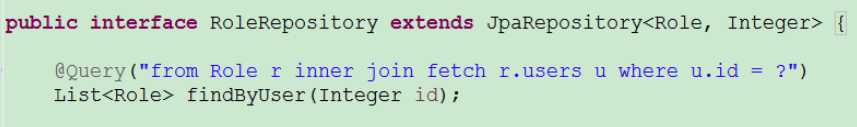
2、实现Realm的授权方法



# 十七、自定义Realm实现用户授权功能（下）

1、创建角色的Service DAO





2、创建权限的Service、DAO

