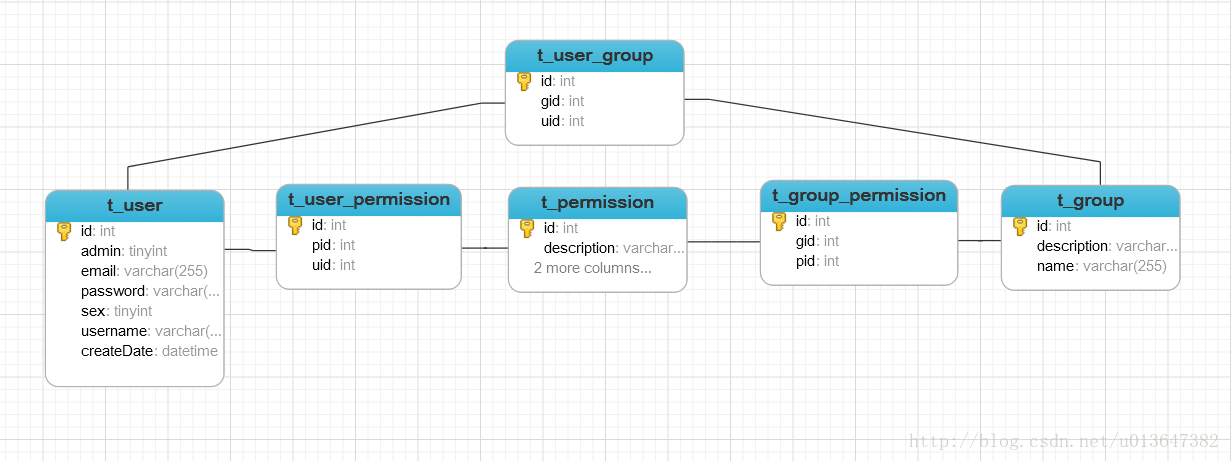
###### 通俗易懂权限管理模块设计-Java

最近一直在做CMS系统，发现一些内容其实都是重复出现的，例如**权限管理模块**。权限管理模块就是为了管理用户是否有权利访问某个权限，如果不能则拒绝访问。其实Java中已经有很成熟的权限管理框架，例如 Shiro，Spring Security等，也推荐大家使用。

但是一些设计上的东西还是要重复再造过轮子才能发现里面的精髓，所以这份代码，供不太明白的同学也参考参考。

**下面我将详细介绍一下这个简单的权限管理模块，以便大家可以拿去直接使用。**

# 数据表



## 主要三张表

* **t\_user：用户表，访问系统的对象**
* **t\_permission：权限表，每条记录就是一个权限，也就是一个 url 地址**
* **t\_group：用户组表，一个组可以包含多个权限，用户如果在这个组中，则享有所有权限**

## 简单例子

来个简单的例子，例如一个CMS系统中，有一些权限：添加文章( /admin/article/add)，查看文章( /admin/article/{id})，删除文章( /admin/delete/{id})，修改文章( /admin/article/update/{id})，每个权限其实对应一个controller的 requet 地址。这些内容存放在 **t\_permission** 中。

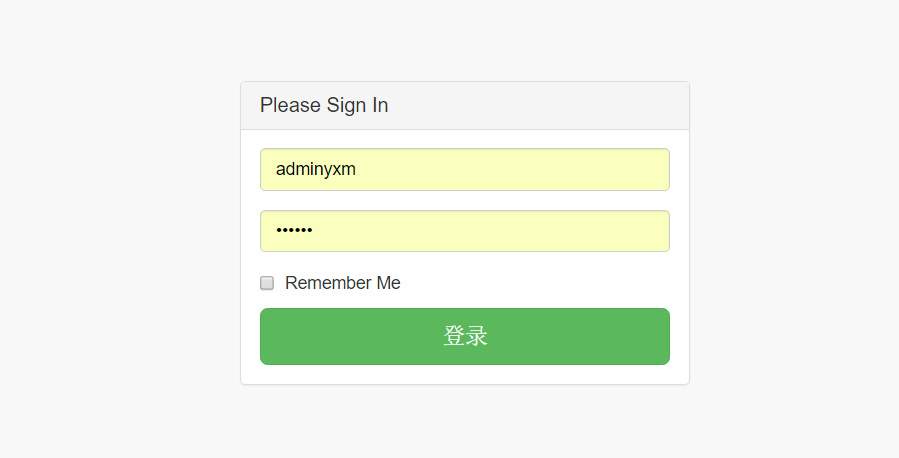
* 然后系统中**有一些用户都放在 t\_user**中，然后我可以**为 每个user 分配一些权限，**一个user叫 张三，我可以给他**直接添加权限：查看文章。那么他只能查看**，不能增删改。
* 然后**用户组表可以看作一个部门，例如一个用户组叫：文章管理组，**那么这个组可以添加：文章的增删改查4项权限。现在我再将 **user分配到 文章管理组 中，**那么张三就拥有了 文章的增删改查的所有权限。
* 最主要的思想就是：**用户的权限，可以直接分配，也可以通过用户组来分配，用户的所有权限就是两者的并集**

# 框架

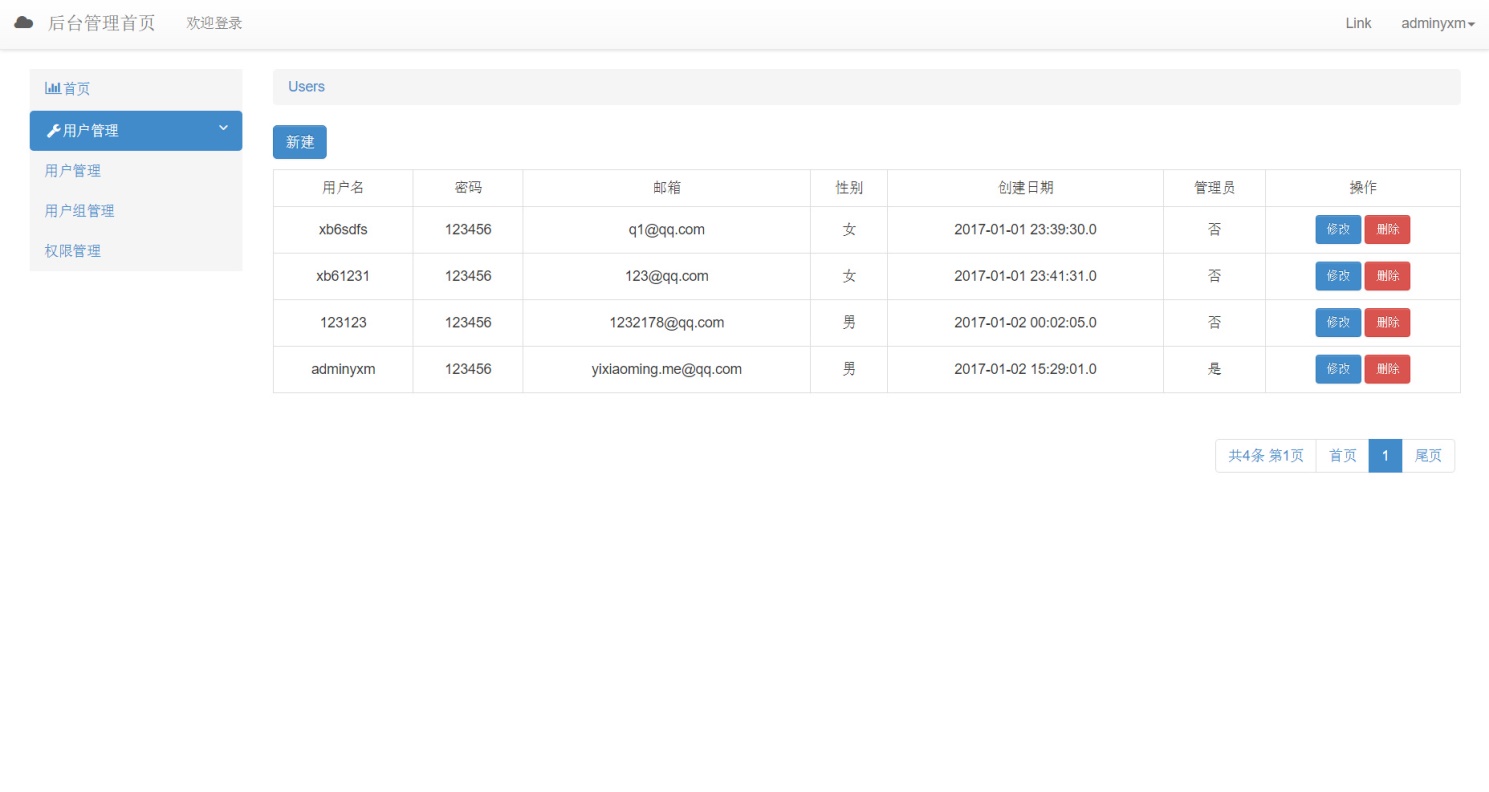
1. Spring 4.26
2. Hibernate 5.10
3. Java 8

例子采用 SpringMVC + Hibernate + Bootstrap搭建，简单实用，对于小项目使用绝对没有问题。   
然后里面也用到了一些开源的Web前端项目，例如SB-Admin，Bootstrap Multiselect，MetisMenu，Bootstrap Validation等，由于我本来不是做前端的，所以UI上的东西只能借助开源项目来完成。下面展示一下效果：

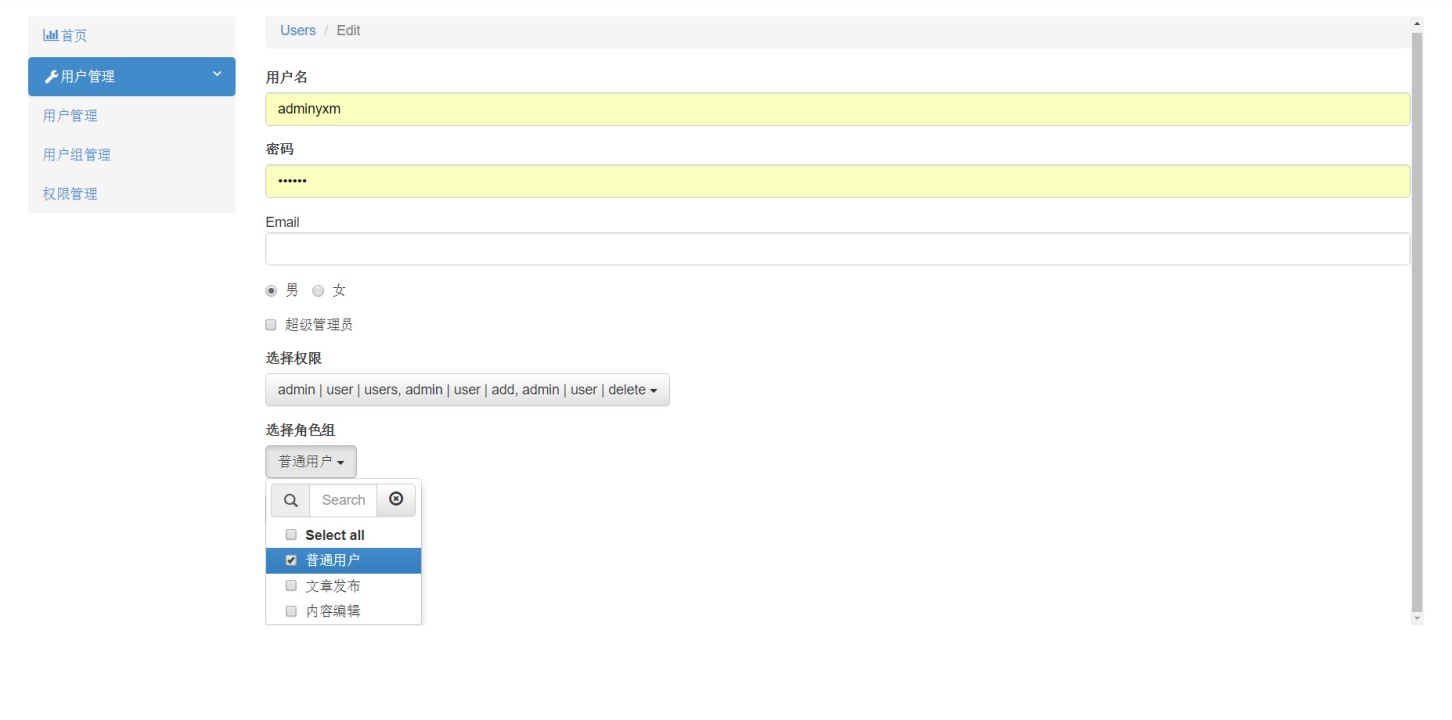
登陆页



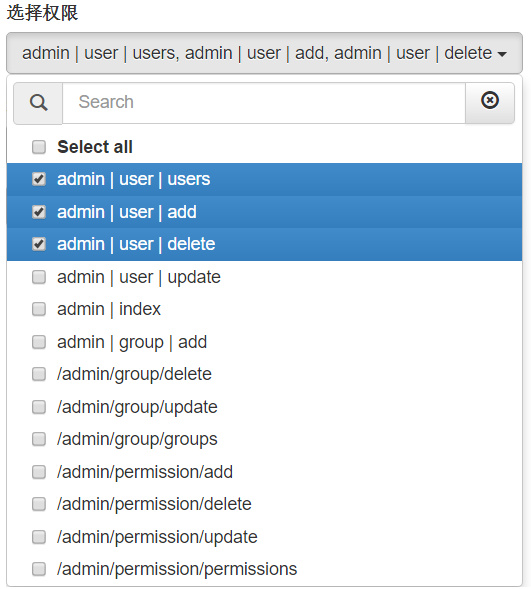
* **用户列表页，用户组列表页，权限列表页 都类似**



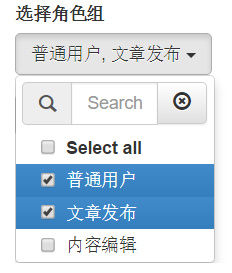
用户添加，修改页：



选择权限



选择用户组



当选定了权限和用户组，那么用户的权限就是 **权限+角色组所有的权限** 的并集。

登陆成功后将所有权限放在session中：

LoginController

List<Permission> permissions = userService.listUserPermissions(user.getId());

List<Integer> gids = userService.listUserGids(user.getId());

List<Permission> groupPermissions = groupService.listGroupsPermissions(ListUtil.list2array(gids));

for (Permission p : groupPermissions) {

if (!permissions.contains(p)) {

permissions.add(p);

}

}

request.getSession().setAttribute(Constant.LOGIN\_USER, user);

request.getSession().setAttribute(Constant.LOGIN\_PERMISSIONS, permissions);

然后写两个个拦截器，一个判断登录，一个判断权限，每次访问链接前先判断是否有这个权限，如果没有则抛出异常。**这里有一点还需要注意，如果用户的类型使 Admin 的话，那么默认就拥有所有权限，所以不需要验证。通过 user 的 isAdmin 字段判断**

LoginInterceptor

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object o) throws Exception {

User user = (User) request.getSession().getAttribute(Constant.LOGIN\_USER);

if (request.getRequestURL().toString().contains("/admin")) {

if (user == null) {

response.sendRedirect(request.getContextPath() + "/login");

return false;

}

}

return true;

}

PermissionInterceptor

public boolean preHandle(HttpServletRequest request, HttpServletResponse response, Object o) throws Exception {

String url = request.getRequestURL().toString();

if (url.contains("/admin")) {

List<Permission> permissions = (List<Permission>) request.getSession().getAttribute(Constant.LOGIN\_PERMISSIONS);

User user = (User) request.getSession().getAttribute(Constant.LOGIN\_USER);

if (permissions == null || user == null) {

response.sendRedirect(request.getContextPath() + "/login");

return false;

}

// 如果是admin，就不需要权限验证

if (user.getAdmin()) {

return true;

}

boolean hasPermission = false;

for (Permission permission : permissions) {

if (url.contains(permission.getUrl())) {

hasPermission = true;

break;

}

}

if (!hasPermission) {

throw new CmsException("没有权限访问：" + url);

}

}

return true;

}

# 总结

这里洋洋洒洒的写了一些权限模块中的内容，只是里面的主线，细节的代码大家可以参考Github上的代码地址链接，大神略过。希望能给正在学习这个内容的同学一点帮助，反正造这个轮子，我学到了不少。。。