**技术栈**

**后端技术栈**

1.SpringBoot  
2.SpringSecurity  
3.MyBatis  
4.MySQL

**前端技术栈**

1.Vue  
2.ElementUI  
3.axios  
4.vue-router

还有其他一些琐碎的技术就不一一列举了。

**快速部署**

1.clone项目到本地git@github.com:lenve/vhr.git

2.数据库脚本放在hrserver项目的resources目录下，在MySQL中执行数据库脚本

3.数据库配置在hrserver项目的resources目录下的application.properties文件中

4.在IntelliJ IDEA中运行hrserver项目

**OK，至此，服务端就启动成功了，此时我们直接在地址栏输入http://localhost:8082/index.html即可访问我们的项目，如果要做二次开发，请继续看第五、六步。**

5.进入到vuehr目录中，在命令行依次输入如下命令：

# 安装依赖

npm install

# 在 localhost:8080 启动项目 要保持服务端是开启状态的。

npm run dev

由于我在vuehr项目中已经配置了端口转发，将数据转发到SpringBoot上，因此项目启动之后，在浏览器中输入http://localhost:8080就可以访问我们的前端项目了，所有的请求通过端口转发将数据传到SpringBoot中（注意此时不要关闭SpringBoot项目）。

6.最后可以用WebStorm等工具打开vuehr项目，继续开发，开发完成后，当项目要上线时，依然进入到vuehr目录，然后执行如下命令：

npm run build

该命令执行成功之后，vuehr目录下生成一个dist文件夹，将该文件夹中的两个文件static和index.html拷贝到SpringBoot项目中resources/static/目录下，然后就可以像第4步那样直接访问了。

**步骤5中需要大家对NodeJS、NPM等有一定的使用经验，不熟悉的小伙伴可以先自行搜索学习下，推荐[Vue官方教程](https://cn.vuejs.org/v2/guide/)。**

**文档**

文档是对项目开发过程中遇到的一些问题的详细记录，主要是为了帮助没有基础的小伙伴快速理解这个项目。

1.[权限数据库设计](https://github.com/lenve/vhr/wiki/1.%E6%9D%83%E9%99%90%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1)  
2.[服务端环境搭建](https://github.com/lenve/vhr/wiki/2.%E6%9C%8D%E5%8A%A1%E7%AB%AF%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)  
3.[动态处理角色和资源的关系](https://github.com/lenve/vhr/wiki/3.%E5%8A%A8%E6%80%81%E5%A4%84%E7%90%86%E8%A7%92%E8%89%B2%E5%92%8C%E8%B5%84%E6%BA%90%E7%9A%84%E5%85%B3%E7%B3%BB)  
4.[密码加密并加盐](https://github.com/lenve/vhr/wiki/4.%E5%AF%86%E7%A0%81%E5%8A%A0%E5%AF%86%E5%B9%B6%E5%8A%A0%E7%9B%90)  
5.[服务端异常的统一处理](https://github.com/lenve/vhr/wiki/5.%E6%9C%8D%E5%8A%A1%E7%AB%AF%E5%BC%82%E5%B8%B8%E7%9A%84%E7%BB%9F%E4%B8%80%E5%A4%84%E7%90%86)  
6.[axios请求封装,请求异常统一处理](https://github.com/lenve/vhr/wiki/6.axios%E8%AF%B7%E6%B1%82%E5%B0%81%E8%A3%85,%E8%AF%B7%E6%B1%82%E5%BC%82%E5%B8%B8%E7%BB%9F%E4%B8%80%E5%A4%84%E7%90%86)  
7.[将请求方法挂到Vue上](https://github.com/lenve/vhr/wiki/7.%E5%B0%86%E8%AF%B7%E6%B1%82%E6%96%B9%E6%B3%95%E6%8C%82%E5%88%B0Vue%E4%B8%8A)  
8.[登录状态的保存](https://github.com/lenve/vhr/wiki/8.%E7%99%BB%E5%BD%95%E7%8A%B6%E6%80%81%E7%9A%84%E4%BF%9D%E5%AD%98)  
9.[登录成功后动态加载组件](https://github.com/lenve/vhr/wiki/9.%E7%99%BB%E5%BD%95%E6%88%90%E5%8A%9F%E5%90%8E%E5%8A%A8%E6%80%81%E5%8A%A0%E8%BD%BD%E7%BB%84%E4%BB%B6)  
10.[角色资源关系管理](https://github.com/lenve/vhr/wiki/10.%E8%A7%92%E8%89%B2%E8%B5%84%E6%BA%90%E5%85%B3%E7%B3%BB%E7%AE%A1%E7%90%86)  
11.[用户角色关系管理](https://github.com/lenve/vhr/wiki/11.%E7%94%A8%E6%88%B7%E8%A7%92%E8%89%B2%E5%85%B3%E7%B3%BB%E7%AE%A1%E7%90%86)

**更新记录**

**2018.1.10 更新**

本次更新版本：v20180110

本次更新完成了部门管理功能，页面在 **[系统管理->基础信息设置->部门管理]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

12.[部门数据库设计与存储过程编写](https://github.com/lenve/vhr/wiki/12.%E9%83%A8%E9%97%A8%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1%E4%B8%8E%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B%E7%BC%96%E5%86%99)  
13.[递归查询与存储过程调用](https://github.com/lenve/vhr/wiki/13.%E9%80%92%E5%BD%92%E6%9F%A5%E8%AF%A2%E4%B8%8E%E5%AD%98%E5%82%A8%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8)  
14.[Tree树形控件使用要点](https://github.com/lenve/vhr/wiki/14.Tree%E6%A0%91%E5%BD%A2%E6%8E%A7%E4%BB%B6%E4%BD%BF%E7%94%A8%E8%A6%81%E7%82%B9)

**2018.1.12 更新**

本次更新版本：v20180112

本次更新完成了职称管理和职位管理，页面在 **[系统管理->基础信息设置->职位管理]** 和 **[系统管理->基础信息设置->职称管理]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

15.[职位管理和职称管理功能介绍](https://github.com/lenve/vhr/wiki/15.%E8%81%8C%E4%BD%8D%E7%AE%A1%E7%90%86%E5%92%8C%E8%81%8C%E7%A7%B0%E7%AE%A1%E7%90%86%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)  
16.[组件复用](https://github.com/lenve/vhr/wiki/16.%E7%BB%84%E4%BB%B6%E5%A4%8D%E7%94%A8)  
17.[[题外话]利用git标签回退至任意版本](https://github.com/lenve/vhr/wiki/17.%5B%E9%A2%98%E5%A4%96%E8%AF%9D%5D%E5%88%A9%E7%94%A8git%E6%A0%87%E7%AD%BE%E5%9B%9E%E9%80%80%E8%87%B3%E4%BB%BB%E6%84%8F%E7%89%88%E6%9C%AC)

**2018.1.15 更新**

本次更新版本：v20180115

本次更新完成了员工基本信息管理，页面在 **[员工资料->基本资料]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

18.[员工基本信息管理功能介绍](https://github.com/lenve/vhr/wiki/18.%E5%91%98%E5%B7%A5%E5%9F%BA%E6%9C%AC%E4%BF%A1%E6%81%AF%E7%AE%A1%E7%90%86%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)  
19.[SpringBoot中自定义参数绑定](https://github.com/lenve/vhr/wiki/19.SpringBoot%E4%B8%AD%E8%87%AA%E5%AE%9A%E4%B9%89%E5%8F%82%E6%95%B0%E7%BB%91%E5%AE%9A)

**2018.1.16 更新**

本次更新版本：v20180116

本次更新完成了员工的高级搜索功能，页面在 **[员工资料->基本资料]**

**相关文档：**

20.[高级搜索功能介绍](https://github.com/lenve/vhr/wiki/20.%E9%AB%98%E7%BA%A7%E6%90%9C%E7%B4%A2%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)

**2018.1.17 更新**

本次更新版本：v20180117

本次更新完成了员工数据的导入导出功能，即可将员工数据导出为Excel，也可以将外部Excel导入到员工数据表中，页面在 **[员工资料->基本资料]**

**相关文档：**

21.[Excel导入导出效果图](https://github.com/lenve/vhr/wiki/21.Excel%E5%AF%BC%E5%85%A5%E5%AF%BC%E5%87%BA%E6%95%88%E6%9E%9C%E5%9B%BE)  
22.[SpringMVC文件下载的两种方式](https://github.com/lenve/vhr/wiki/22.SpringMVC%E6%96%87%E4%BB%B6%E4%B8%8B%E8%BD%BD%E7%9A%84%E4%B8%A4%E7%A7%8D%E6%96%B9%E5%BC%8F)  
23.[POI生成Excel](https://github.com/lenve/vhr/wiki/23.POI%E7%94%9F%E6%88%90Excel)  
24.[axios下载文件](https://github.com/lenve/vhr/wiki/24.axios%E4%B8%8B%E8%BD%BD%E6%96%87%E4%BB%B6)  
25.[使用POI实现Excel导入](https://github.com/lenve/vhr/wiki/25.%E4%BD%BF%E7%94%A8POI%E5%AE%9E%E7%8E%B0Excel%E5%AF%BC%E5%85%A5)

**2018.1.19 更新**

本次更新版本：v20180119

本次更新主要实现了当管理员添加一个用户时，添加成功后，会根据该用户的邮箱自动向用户发送一封欢迎入职邮件，页面在 **[员工资料->基本资料->添加员工]** 详情可以参考下面的文档。 **注意：邮件发送需要小伙伴小伙伴自己配置授权码，配置方式参考下面的文档，配置文件在**[**src/main/java/org/sang/common/EmailRunnable.java**](https://github.com/lenve/vhr/blob/master/hrserver/src/main/java/org/sang/common/EmailRunnable.java)

**相关文档：**

26.[SpringBoot中使用Freemarker邮件模板生成邮件](https://github.com/lenve/vhr/wiki/26.SpringBoot%E4%B8%AD%E4%BD%BF%E7%94%A8Freemarker%E9%82%AE%E4%BB%B6%E6%A8%A1%E6%9D%BF%E7%94%9F%E6%88%90%E9%82%AE%E4%BB%B6)  
27.[Java中邮件的发送](https://github.com/lenve/vhr/wiki/27.Java%E4%B8%AD%E9%82%AE%E4%BB%B6%E7%9A%84%E5%8F%91%E9%80%81)  
28.[SpringBoot中使用新线程发送邮件](https://github.com/lenve/vhr/wiki/28.SpringBoot%E4%B8%AD%E4%BD%BF%E7%94%A8%E6%96%B0%E7%BA%BF%E7%A8%8B%E5%8F%91%E9%80%81%E9%82%AE%E4%BB%B6)

**2018.1.25 更新**

本次更新版本：v20180125

本次更新主要完成了工资账套管理功能，页面在 **[薪资管理->工资账套管理]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

29.[工资账套管理功能介绍](https://github.com/lenve/vhr/wiki/29.%E5%B7%A5%E8%B5%84%E8%B4%A6%E5%A5%97%E7%AE%A1%E7%90%86%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)

**2018.1.26 更新**

本次更新版本：v20180126

本次更新主要完成了员工账套设置功能，页面在 **[薪资管理->员工账套设置]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

30.[员工账套设置功能介绍](https://github.com/lenve/vhr/wiki/30.%E5%91%98%E5%B7%A5%E8%B4%A6%E5%A5%97%E8%AE%BE%E7%BD%AE%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)

**2018.2.2 更新**

本次更新版本：v20180202

本次更新完成了HR在线聊天功能，页面在 **[Home页->右上角铃铛->好友聊天]**

**相关文档：**

31.[在线聊天功能介绍](https://github.com/lenve/vhr/wiki/31.%E5%9C%A8%E7%BA%BF%E8%81%8A%E5%A4%A9%E5%8A%9F%E8%83%BD%E4%BB%8B%E7%BB%8D)  
32.[在线聊天功能实现](https://github.com/lenve/vhr/wiki/32.%E5%9C%A8%E7%BA%BF%E8%81%8A%E5%A4%A9%E5%8A%9F%E8%83%BD%E5%AE%9E%E7%8E%B0)

**2018.2.5 更新**

本次更新版本：v20180205

本次更新完成了管理员发送系统通知功能，页面在 **[Home页->右上角铃铛->系统通知]**

本次更新也更新了数据库脚本，小伙伴们需要重新下载数据库脚本执行。

**相关文档：**

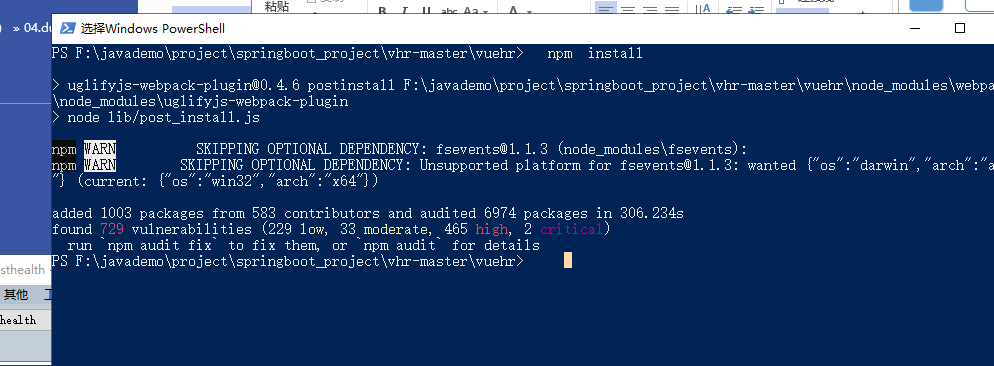
33.[系统通知功能实现](https://github.com/lenve/vhr/wiki/33.%E7%B3%BB%E7%BB%9F%E9%80%9A%E7%9F%A5%E5%8A%9F%E8%83%BD%E5%AE%9E%E7%8E%B0)

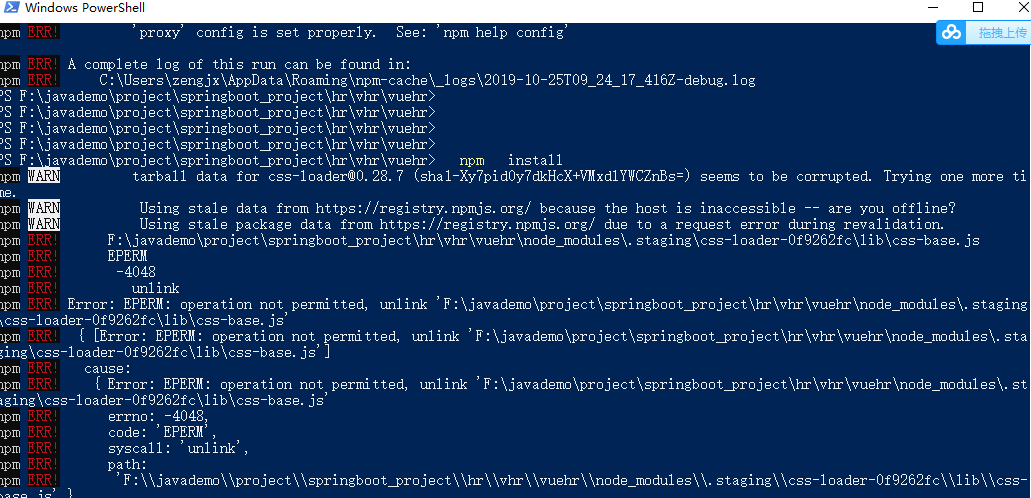
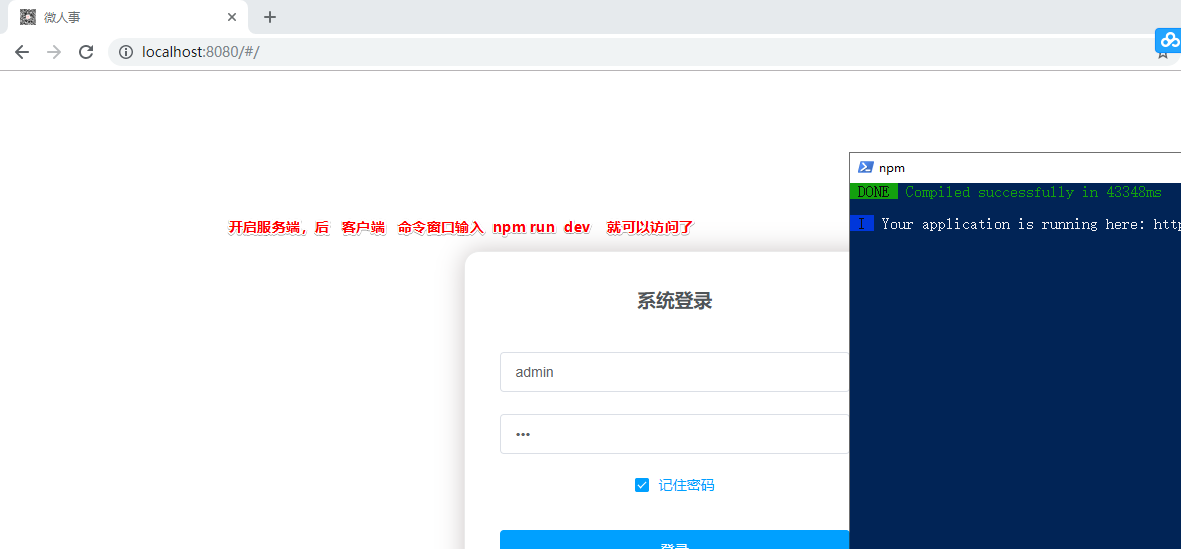
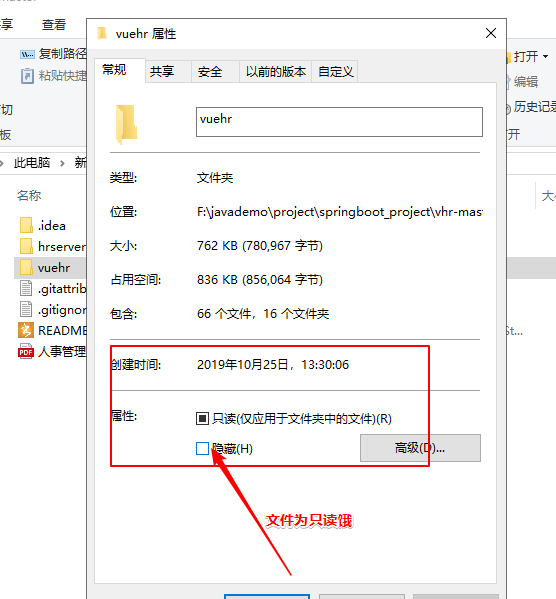
**其他资料**

关注公众号**江南一点雨**，专注于 Spring Boot+微服务，定期视频教程分享，关注后回复 Java ，领取松哥为你精心准备的 Java 干货！

[](https://camo.githubusercontent.com/cf41e65d8db1c04d74dc4b57f367e4c47270eb0b/687474703a2f2f7777772e6a617661626f792e6f72672f696d616765732f73622f6a617661626f792e6a7067)

客户端输入 rpm install 提示没有权限 是因为文件为只读的。





# 1.权限数据库设计

江南一点雨 edited this page on 6 Jan 2018 · [10 revisions](https://github.com/lenve/vhr/wiki/1.%E6%9D%83%E9%99%90%E6%95%B0%E6%8D%AE%E5%BA%93%E8%AE%BE%E8%AE%A1/_history)

权限数据库主要包含了五张表，分别是资源表、角色表、用户表、资源角色表、用户角色表，数据库关系模型如下：

关于这个表，我说如下几点：

1.hr表是用户表，存放了用户的基本信息。

2.role是角色表，name字段表示角色的英文名称，按照SpringSecurity的规范，将以ROLE\_开始，nameZh字段表示角色的中文名称。

3.menu表是一个资源表，该表涉及到的字段有点多，由于我的前端采用了Vue来做，因此当用户登录成功之后，系统将根据用户的角色动态加载需要的模块，所有模块的信息将保存在menu表中，menu表中的path、component、iconCls、keepAlive、requireAuth等字段都是Vue-Router中需要的字段，也就是说menu中的数据到时候会以json的形式返回给前端，再由vue动态更新router，menu中还有一个字段url，表示一个url pattern，即路径匹配规则，假设有一个路径匹配规则为/admin/\*\*,那么当用户在客户端发起一个/admin/user的请求，将被/admin/\*\*拦截到，系统再去查看这个规则对应的角色是哪些，然后再去查看该用户是否具备相应的角色，进而判断该请求是否合法。

下图分别是用户表、角色表以及资源表中的部分数据(数据库脚本可以在文末的项目地址中下载，位置[resources/vhr.sql](https://github.com/lenve/vhr/blob/master/hrserver/src/main/resources/vhr.sql))：

# 10.角色资源关系管理

江南一点雨 edited this page on 5 Jan 2018 · [7 revisions](https://github.com/lenve/vhr/wiki/10.%E8%A7%92%E8%89%B2%E8%B5%84%E6%BA%90%E5%85%B3%E7%B3%BB%E7%AE%A1%E7%90%86/_history)

这个主要是给不同角色分配不同的资源。

## 角色展示

角色的展示采用了ElementUI中的**Collapse 折叠面板**，并且采用了手风琴模式，即一次只打开一个角色，如下图：

角色中资源的展示则采用了ElementUI中的树形控件，管理员可以直接直接点击勾选，然后点击修改按钮，进行资源的分配。

## 核心思路

核心代码如下：

<el-collapse v-model="activeColItem" accordion style="width: 500px;" @change="collapseChange">

<el-collapse-item v-for="(item,index) in roles" :title="item.nameZh" :name="item.id" :key="item.name">

<el-card class="box-card">

<div slot="header">

<span>可访问的资源</span>

<el-button type="text"

style="color: #f6061b;margin: 0px;float: right; padding: 3px 0;width: 15px;height:15px"

icon="el-icon-delete" @click="deleteRole(item.id,item.nameZh)"></el-button>

</div>

<div>

<el-tree :props="props"

:key="item.id"

:data="treeData"

:default-checked-keys="checkedKeys"

node-key="id"

ref="tree"

show-checkbox

highlight-current

@check-change="handleCheckChange">

</el-tree>

</div>

<div style="display: flex;justify-content: flex-end;margin-right: 10px">

<el-button size="mini" @click="cancelUpdateRoleMenu">取消修改</el-button>

<el-button type="primary" size="mini" @click="updateRoleMenu(index)">确认修改</el-button>

</div>

</el-card>

</el-collapse-item>

</el-collapse>

核心思路如下：

1.通过for循环渲染出el-collapse-item，将角色展示出来。  
2.el-collapse-item的内容就是一个树形控件，很明显，树形控件的数量和el-collapse-item的数量是一样多的，但是考虑到el-collapse-item使用了手风琴模式，即一次只有一个折叠面板被打开，因此树形控件的数据源只有一个，即多个树形控件共用一个数据源，为了避免数据紊乱，我采取了这样的数据加载方式：当用户每次点击折叠面板的时候，我都根据当前折叠面板所对应的角色去查询该角色所对应的资源，同时也查询所有的资源，将查到的数据交给树形控件去展示。这样可以避免为每一个树形控件都准备一份数据。