QueryWrapper<Files> queryWrapper=new QueryWrapper<>();  
queryWrapper.eq("md5",md5);  
fileMapper.selectList(queryWrapper);  
Files files=fileMapper.selectOne(queryWrapper);

@GetMapping("/name/{name}")  
public Result findSuppliers(@PathVariable String name) {  
 QueryWrapper<Supplier> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("name",name);  
 List<Supplier> list = supplierService.list(queryWrapper);  
 return Result.*success*(list);  
}

@GetMapping("/id")  
public Result findDepartment(@PathVariable String name) {  
 QueryWrapper<Department> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("name",name);  
 List<Department> list = departmentService.list(queryWrapper);  
 return Result.*success*(list);

@GetMapping("/name/{name}")  
public Result findDepartment(@PathVariable String name) {  
 QueryWrapper<Department> queryWrapper = new QueryWrapper<>();  
 queryWrapper.eq("name", name);  
 List<Department> list = departmentService.list(queryWrapper);  
 return Result.*success*(list);  
}

/\*@PostMapping  
public boolean save(@RequestBody **${**entity**} ${**table.entityPath**}**) {  
 return **${**table.entityPath**}**Service.saveOrUpdate(**${**table.entityPath**}**);  
 }\*/

/\*@DeleteMapping("/{id}")  
public Boolean delete(@PathVariable Integer id) {  
 return **${**table.entityPath**}**Service.removeById(id);  
 }\*/

/\*@PostMapping("/del/batch")  
public boolean deleteBatch(@RequestBody List<Integer> ids) {  
 return **${**table.entityPath**}**Service.removeByIds(ids);  
 }\*/

/\*@GetMapping  
public List<**${**entity**}**> findAll() {  
 return **${**table.entityPath**}**Service.list();  
 }\*/

/\*@GetMapping("/{id}")  
public **${**entity**}** findOne(@PathVariable Integer id) {  
 return **${**table.entityPath**}**Service.getById(id);  
 }\*/

@GetMapping("/page")  
public Page<**${**entity**}**> findPage(@RequestParam Integer pageNum,  
@RequestParam Integer pageSize) {  
 QueryWrapper<User> queryWrapper = new QueryWrapper<>();  
 queryWrapper.orderByDesc("id");  
 return **${**table.entityPath**}**Service.page(new Page<>(pageNum, pageSize), queryWrapper);  
 }  
  
}

<el-form-item label="页面路径">  
 <el-input v-model="form.pPath" autocomplete="off"></el-input>  
</el-form-item>

<el-table-column prop="pPath" label="页面路径"></el-table-column>

<el-table-column prop="num" label="顺序"></el-table-column>

<el-form-item label="顺序">  
 <el-input v-model="form.num" autocomplete="off"></el-input>  
</el-form-item>

<el-menu-item index="/">  
 <template slot="title">  
 <i class="el-icon-house"></i>  
 <span slot="title">主页</span>  
 </template>  
</el-menu-item>  
<el-menu-item index="/plan">  
 <template slot="title">  
 <i class="el-icon-tickets"></i>  
 <span slot="title">采购计划</span>  
 </template>  
</el-menu-item>  
<el-submenu index="1">  
 <template slot="title"><i class="el-icon-star-on"></i>  
 <span slot="title">交易管理</span>  
 </template>  
 <el-menu-item-group>  
 <el-menu-item index="/order"><i class="el-icon-shopping-bag-1"></i>订单管理</el-menu-item>  
 <el-menu-item index="/document"><i class="el-icon-document-copy"></i>合同管理</el-menu-item>  
 <el-menu-item index="/return"><i class="el-icon-shopping-cart-1"></i>履约管理</el-menu-item>  
 </el-menu-item-group>  
</el-submenu>  
<el-submenu index="2">  
 <template slot="title"><i class="el-icon-s-data"></i>  
 <span slot="title">信息管理</span>  
 </template>  
 <el-menu-item-group>  
 <el-menu-item index="/department"><i class="el-icon-office-building"></i>机构管理</el-menu-item>  
 <el-menu-item index="/supplier"><i class="el-icon-user-solid"></i>供应商管理</el-menu-item>  
 <el-menu-item index="/goods"><i class="el-icon-s-goods"></i>商品管理</el-menu-item>  
 </el-menu-item-group>  
</el-submenu>  
<el-submenu index="3">  
 <template slot="title"><i class="el-icon-s-tools"></i>  
 <span slot="title">系统管理</span>  
 </template>  
 <el-menu-item-group>  
 <el-menu-item index="/user"><i class="el-icon-user"></i>用户管理</el-menu-item>  
 <el-menu-item index="/file"><i class="el-icon-notebook-2"></i>文件管理</el-menu-item>  
 <el-menu-item index="/role"><i class="el-icon-key"></i>角色管理</el-menu-item>  
 <el-menu-item index="/menu"><i class="el-icon-menu"></i>菜单管理</el-menu-item>  
 </el-menu-item-group>  
</el-submenu>

QueryWrapper<Supplier> queryWrapper = new QueryWrapper<>();  
if (!"".equals(name)) {  
 queryWrapper.like("name", name);  
}  
queryWrapper.orderByDesc("id");  
return Result.*success*(supplierService.page(new Page<>(pageNum, pageSize), queryWrapper));

{  
 path: '/',  
 name: 'Manage',  
 component: () => import('../views/Manage'),  
 redirect:"/home",  
 children:[  
 {path: 'home', name: '首页', component: () => import('../views/Home'),},  
 {path: 'user', name: '用户管理', component: () => import('../views/User'),},  
 {path: 'person', name: '用户信息', component: () => import('../views/Person'),},  
 {path: 'file', name: '文件管理', component: () => import('../views/File'),},  
 {path: 'password', name: '修改密碼', component: () => import('../views/Password'),},  
 {path: 'role', name: '角色管理', component: () => import('../views/Role'),},  
 {path: 'menu', name: '菜单管理', component: () => import('../views/Menu'),},  
 {path: 'plan', name: '采购计划', component: () => import('../views/Plan'),},  
 {path: 'order', name: '订单管理', component: () => import('../views/Order'),},  
 {path: 'document', name: '合同管理', component: () => import('../views/Document'),},  
 {path: 'return', name: '履约管理', component: () => import('../views/Return'),},  
 {path: 'department', name: '机构管理', component: () => import('../views/Department'),},  
 {path: 'supplier', name: '供应商管理', component: () => import('../views/Supplier'),},  
 {path: 'goods', name: '商品管理', component: () => import('../views/Goods'),}  
 ]  
},

// 路由守卫  
***router***.beforeEach((to, from, next) => {  
 ***localStorage***.setItem("currentPathName", to.name) // 设置当前的路由名称，为了在Header组件中去使用  
 ***store***.commit("setPath") // 触发store的数据更新  
 next() // 放行路由  
})

<el-table-column prop="charger" label="负责人" align="center" >  
</el-table-column>  
<el-table-column prop="phone" label="联系方式" width="120" align="center" >  
</el-table-column>  
<el-table-column prop="email" label="电子邮箱" width="150" align="center" >  
</el-table-column>  
<el-table-column prop="telephone" label="办公电话" width="150" align="center" >  
</el-table-column>

this.request.get("/role").then(res => {  
 this.roles = res.data  
})

{value: '选项1', label: '国家机关'},{value: '选项2', label: '事业单位'},  
 {value: '选项3', label: '群众团体'},{value: '选项4', label: '政协机关'},{value: '选项5', label: '人大机关'},  
 {value: '选项6', label: '其他机关'}

<select id="findPage" resultType="com.example.springboot.entity.Good">  
 select in\_good.\* ,in\_supplier.name as supplier from in\_good  
 left join in\_supplier  
 on in\_good.supplier\_id=in\_supplier.id  
 <where>  
 <if test="goodname!=null and goodname!=''">  
 and in\_good.goodname like concat('%',#{goodname},'%')  
 </if>  
 </where>  
</select>

Page<Good>page = goodService.findPage(new Page<>(pageNum, pageSize), goodname);  
return Result.*success*(page);

Page<Good> findPage(Page<Good> page, String goodname);

Page<Good> findPage(Page<Good> page, @Param("goodname") String goodname);

@Resource  
private GoodMapper goodMapper;  
  
@Override  
public Page<Good> findPage(Page<Good> page, String goodname) {  
 return goodMapper.findPage(page, goodname);  
}

<el-upload action="http://localhost:9090/order/import" :show-file-list="false" accept="xlsx" :on-success="handleExcelImportSuccess" style="display: inline-block">  
 <el-button type="primary" >导入<i class="el-icon-bottom"></i></el-button>  
</el-upload>  
<el-button type="primary" @click="exp" style="margin-left: 10px">导出<i class="el-icon-top"></i></el-button>

**包括两个子系统的设计，分别包括管理功能子系统和采购功能子系统。在登陆的时候可以根据用户的不同的权限登陆到不同的子系统。如果权限为管理员就登陆到管理功能子系统，如果权限为操作员和采购员则登陆到采购功能子系统。**

**6.1.1 管理功能子系统**

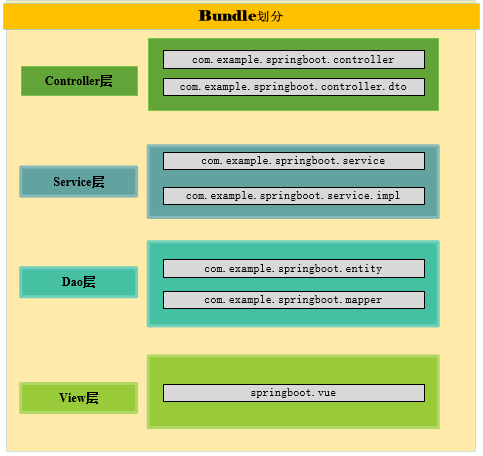
**该子系统是管理员进行操作的系统。它主要包括系统设置、货品信息维护、用户管理、采购信息记录等功能。用户管理设置了两种用户权限，管理员和操作员，其中操作员权限由管理员设置。**

**6.1.2 采购功能子系统**

**该子系统是为了方便企业采购货品方面的有擅管理的。该子系统是操作员进行操作的系统。该系统的功能主要是采购管理、单据查询、信息录入的功能。单据查询中可以查询商品信息、采购信息、供应商信息、合同信息等。**

**6.2 系统模块设计**

整个系统由Controller层、Service层、Dao层和View层四个层次组成，系统bundle划分图如图5.3所示



**图6.2 bundle划分图**

采购信息管理

采购预算管理

采购交易管理

**政府采购管理系统**