===================================Main.js=====================================

import Vue from 'vue'  
  
import Cookies from 'js-cookie'  
  
import Element from 'element-ui'  
import './assets/styles/element-variables.scss'  
  
import '@/assets/styles/index.scss' // global css  
import '@/assets/styles/ruoyi.scss' // ruoyi css  
import App from './App'  
import store from './store'  
import router from './router'  
import directive from './directive' // directive  
import plugins from './plugins' // plugins  
import { download } from '@/utils/request'  
  
import './assets/icons' // icon  
import './permission' // permission control  
import { getDicts } from "@/api/system/dict/data";  
import { getConfigKey } from "@/api/system/config";  
import { parseTime, resetForm, addDateRange, selectDictLabel, selectDictLabels, handleTree } from "@/utils/ruoyi";  
// 分页组件  
import Pagination from "@/components/Pagination";  
// 自定义表格工具组件  
import RightToolbar from "@/components/RightToolbar"  
// 富文本组件  
import Editor from "@/components/Editor"  
// 文件上传组件  
import FileUpload from "@/components/FileUpload"  
// 图片上传组件  
import ImageUpload from "@/components/ImageUpload"  
// 图片预览组件  
import ImagePreview from "@/components/ImagePreview"  
// 字典标签组件  
import DictTag from '@/components/DictTag'  
// 头部标签组件  
import VueMeta from 'vue-meta'  
// 字典数据组件  
import DictData from '@/components/DictData'  
  
// 全局方法挂载  
Vue.prototype.getDicts = getDicts  
Vue.prototype.getConfigKey = getConfigKey  
Vue.prototype.parseTime = parseTime  
Vue.prototype.resetForm = resetForm  
Vue.prototype.addDateRange = addDateRange  
Vue.prototype.selectDictLabel = selectDictLabel  
Vue.prototype.selectDictLabels = selectDictLabels  
Vue.prototype.download = download  
Vue.prototype.handleTree = handleTree  
  
// 全局组件挂载  
Vue.component('DictTag', DictTag)  
Vue.component('Pagination', Pagination)  
Vue.component('RightToolbar', RightToolbar)  
Vue.component('Editor', Editor)  
Vue.component('FileUpload', FileUpload)  
Vue.component('ImageUpload', ImageUpload)  
Vue.component('ImagePreview', ImagePreview)  
  
Vue.use(directive)  
Vue.use(plugins)  
Vue.use(VueMeta)  
DictData.install()  
  
/\*\*  
 \* If you don't want to use mock-server  
 \* you want to use MockJs for mock api  
 \* you can execute: mockXHR()  
 \*  
 \* Currently MockJs will be used in the production environment,  
 \* please remove it before going online! ! !  
 \*/  
  
Vue.use(Element, {  
 size: Cookies.get('size') || 'medium' // set element-ui default size  
})  
  
Vue.config.productionTip = false  
  
new Vue({  
 el: '#app',  
 router,  
 store,  
 render: h => h(App)  
})

===================================Package.json=================================

{  
 "name": "hnust",  
 "version": "3.8.5",  
 "description": "特种设备管理系统",  
 "author": "HNUST",  
 "license": "MIT",  
 "scripts": {  
 "dev": "vue-cli-service serve",  
 "build:prod": "vue-cli-service build",  
 "build:stage": "vue-cli-service build --mode staging",  
 "preview": "node build/index.js --preview",  
 "lint": "eslint --ext .js,.vue src"  
 },  
 "husky": {  
 "hooks": {  
 "pre-commit": "lint-staged"  
 }  
 },  
 "lint-staged": {  
 "src/\*\*/\*.{js,vue}": [  
 "eslint --fix",  
 "git add"  
 ]  
 },  
 "keywords": [  
 "vue",  
 "admin",  
 "dashboard",  
 "element-ui",  
 "boilerplate",  
 "admin-template",  
 "management-system"  
 ],  
 "repository": {  
 "type": "git",  
 "url": "https://gitee.com/y\_project/RuoYi-Vue.git"  
 },  
 "dependencies": {  
 "@riophae/vue-treeselect": "0.4.0",  
 "axios": "0.24.0",  
 "clipboard": "2.0.8",  
 "core-js": "3.25.3",  
 "echarts": "5.4.0",  
 "element-ui": "2.15.12",  
 "file-saver": "2.0.5",  
 "fuse.js": "6.4.3",  
 "highlight.js": "9.18.5",  
 "js-beautify": "1.13.0",  
 "js-cookie": "3.0.1",  
 "jsencrypt": "3.0.0-rc.1",  
 "nprogress": "0.2.0",  
 "quill": "1.3.7",  
 "screenfull": "5.0.2",  
 "sortablejs": "1.10.2",  
 "vue": "2.6.12",  
 "vue-count-to": "1.0.13",  
 "vue-cropper": "0.5.5",  
 "vue-meta": "2.4.0",  
 "vue-router": "3.4.9",  
 "vuedraggable": "2.24.3",  
 "vuex": "3.6.0"  
 },  
 "devDependencies": {  
 "@vue/cli-plugin-babel": "4.4.6",  
 "@vue/cli-plugin-eslint": "4.4.6",  
 "@vue/cli-service": "4.4.6",  
 "babel-eslint": "10.1.0",  
 "babel-plugin-dynamic-import-node": "2.3.3",  
 "chalk": "4.1.0",  
 "compression-webpack-plugin": "5.0.2",  
 "connect": "3.6.6",  
 "eslint": "7.15.0",  
 "eslint-plugin-vue": "7.2.0",  
 "lint-staged": "10.5.3",  
 "runjs": "4.4.2",  
 "sass": "1.32.13",  
 "sass-loader": "10.1.1",  
 "script-ext-html-webpack-plugin": "2.1.5",  
 "svg-sprite-loader": "5.1.1",  
 "vue-template-compiler": "2.6.12"  
 },  
 "engines": {  
 "node": ">=8.9",  
 "npm": ">= 3.0.0"  
 },  
 "browserslist": [  
 "> 1%",  
 "last 2 versions"  
 ]  
}

===================================vue.config.js================================

'use strict'  
const path = require('path')  
  
function resolve(dir) {  
 return path.join(\_\_dirname, dir)  
}  
  
const CompressionPlugin = require('compression-webpack-plugin')  
  
const name = process.env.VUE\_APP\_TITLE || '特种设备管理系统' // 网页标题  
  
const port = process.env.port || process.env.npm\_config\_port || 80 // 端口  
  
// vue.config.js 配置说明  
//官方vue.config.js 参考文档 https://cli.vuejs.org/zh/config/#css-loaderoptions  
// 这里只列一部分，具体配置参考文档  
module.exports = {  
 // 部署生产环境和开发环境下的URL。  
 // 默认情况下，Vue CLI 会假设你的应用是被部署在一个域名的根路径上  
 // 例如 https://www.ruoyi.vip/。如果应用被部署在一个子路径上，你就需要用这个选项指定这个子路径。例如，如果你的应用被部署在 https://www.ruoyi.vip/admin/，则设置 baseUrl 为 /admin/。  
 publicPath: process.env.NODE\_ENV === "production" ? "/" : "/",  
 // 在npm run build 或 yarn build 时 ，生成文件的目录名称（要和baseUrl的生产环境路径一致）（默认dist）  
 outputDir: 'dist',  
 // 用于放置生成的静态资源 (js、css、img、fonts) 的；（项目打包之后，静态资源会放在这个文件夹下）  
 assetsDir: 'static',  
 // 是否开启eslint保存检测，有效值：ture | false | 'error'  
 lintOnSave: process.env.NODE\_ENV === 'development',  
 // 如果你不需要生产环境的 source map，可以将其设置为 false 以加速生产环境构建。  
 productionSourceMap: false,  
 // webpack-dev-server 相关配置  
 devServer: {  
 host: '0.0.0.0',  
 port: port,  
 open: true,  
 proxy: {  
 // detail: https://cli.vuejs.org/config/#devserver-proxy  
 [process.env.VUE\_APP\_BASE\_API]: {  
 target: `http://localhost:8080`,  
 changeOrigin: true,  
 pathRewrite: {  
 ['^' + process.env.VUE\_APP\_BASE\_API]: ''  
 }  
 }  
 },  
 disableHostCheck: true  
 },  
 css: {  
 loaderOptions: {  
 sass: {  
 sassOptions: { outputStyle: "expanded" }  
 }  
 }  
 },  
 configureWebpack: {  
 name: name,  
 resolve: {  
 alias: {  
 '@': resolve('src')  
 }  
 },  
 plugins: [  
 // http://doc.ruoyi.vip/ruoyi-vue/other/faq.html#使用gzip解压缩静态文件  
 new CompressionPlugin({  
 cache: false, // 不启用文件缓存  
 test: /\.(js|css|html)?$/i, // 压缩文件格式  
 filename: '[path].gz[query]', // 压缩后的文件名  
 algorithm: 'gzip', // 使用gzip压缩  
 minRatio: 0.8 // 压缩率小于1才会压缩  
 })  
 ],  
 },  
 chainWebpack(config) {  
 config.plugins.delete('preload') // TODO: need test  
 config.plugins.delete('prefetch') // TODO: need test  
  
 // set svg-sprite-loader  
 config.module  
 .rule('svg')  
 .exclude.add(resolve('src/assets/icons'))  
 .end()  
 config.module  
 .rule('icons')  
 .test(/\.svg$/)  
 .include.add(resolve('src/assets/icons'))  
 .end()  
 .use('svg-sprite-loader')  
 .loader('svg-sprite-loader')  
 .options({  
 symbolId: 'icon-[name]'  
 })  
 .end()  
  
 config  
 .when(process.env.NODE\_ENV !== 'development',  
 config => {  
 config  
 .plugin('ScriptExtHtmlWebpackPlugin')  
 .after('html')  
 .use('script-ext-html-webpack-plugin', [{  
 // `runtime` must same as runtimeChunk name. default is `runtime`  
 inline: /runtime\..\*\.js$/  
 }])  
 .end()  
 config  
 .optimization.splitChunks({  
 chunks: 'all',  
 cacheGroups: {  
 libs: {  
 name: 'chunk-libs',  
 test: /[\\/]node\_modules[\\/]/,  
 priority: 10,  
 chunks: 'initial' // only package third parties that are initially dependent  
 },  
 elementUI: {  
 name: 'chunk-elementUI', // split elementUI into a single package  
 priority: 20, // the weight needs to be larger than libs and app or it will be packaged into libs or app  
 test: /[\\/]node\_modules[\\/]\_?element-ui(.\*)/ // in order to adapt to cnpm  
 },  
 commons: {  
 name: 'chunk-commons',  
 test: resolve('src/components'), // can customize your rules  
 minChunks: 3, // minimum common number  
 priority: 5,  
 reuseExistingChunk: true  
 }  
 }  
 })  
 config.optimization.runtimeChunk('single'),  
 {  
 from: path.resolve(\_\_dirname, './public/robots.txt'), //防爬虫文件  
 to: './' //到根目录下  
 }  
 }  
 )  
 }  
}

===================================permission.js=================================

import router from './router'  
import store from './store'  
import { Message } from 'element-ui'  
import NProgress from 'nprogress'  
import 'nprogress/nprogress.css'  
import { getToken } from '@/utils/auth'  
import { isRelogin } from '@/utils/request'  
  
NProgress.configure({ showSpinner: false })  
  
const whiteList = ['/login', '/register']  
  
router.beforeEach((to, from, next) => {  
 NProgress.start()  
 if (getToken()) {  
 to.meta.title && store.dispatch('settings/setTitle', to.meta.title)  
 /\* has token\*/  
 if (to.path === '/login') {  
 next({ path: '/' })  
 NProgress.done()  
 } else {  
 if (store.getters.roles.length === 0) {  
 isRelogin.show = true  
 // 判断当前用户是否已拉取完user\_info信息  
 store.dispatch('GetInfo').then(() => {  
 isRelogin.show = false  
 store.dispatch('GenerateRoutes').then(accessRoutes => {  
 // 根据roles权限生成可访问的路由表  
 router.addRoutes(accessRoutes) // 动态添加可访问路由表  
 next({ ...to, replace: true }) // hack方法 确保addRoutes已完成  
 })  
 }).catch(err => {  
 store.dispatch('LogOut').then(() => {  
 Message.error(err)  
 next({ path: '/' })  
 })  
 })  
 } else {  
 next()  
 }  
 }  
 } else {  
 // 没有token  
 if (whiteList.indexOf(to.path) !== -1) {  
 // 在免登录白名单，直接进入  
 next()  
 } else {  
 next(`/login?redirect=${to.fullPath}`) // 否则全部重定向到登录页  
 NProgress.done()  
 }  
 }  
})  
  
router.afterEach(() => {  
 NProgress.done()  
})

===================================login.vue====================================<template>  
 <div class="all">  
 <div class="box">  
 <div class="forms">  
 <div class="tips">  
 <span class="log-login-btn">登录界面</span>  
 <span class="log-register-btn" v-if="register" v-cloak><router-link :to="'/register'">注册界面</router-link></span>  
 </div>  
 <div class="login">  
 <div class="form-title">  
 <h2 style="font-weight: bold">H N <span style="color: red;">U </span>S T</h2>  
 <h1 style="font-weight: bold">特种设备管理系统</h1>  
 </div>  
  
 <el-form ref="loginForm" :model="loginForm" :rules="loginRules">  
 <div class="form">  
 <el-form-item id="username" class="username input-item" prop="username">  
 <el-input class="ipts"  
 v-model="loginForm.username"  
 type="text"  
 auto-complete="off"  
 placeholder="请输入账号"  
 >  
 </el-input>  
 </el-form-item>  
 <el-form-item class="password input-item" prop="password">  
 <el-input class="ipts"  
 v-model="loginForm.password"  
 type="password"  
 auto-complete="off"  
 placeholder="请输入密码"  
 @keyup.enter.native="handleLogin"  
 ></el-input>  
 </el-form-item>  
  
 <!-- 验证码-->  
 <el-form-item class="code input-item" prop="code" v-if="captchaEnabled">  
 <el-input class="ipts"  
 v-model="loginForm.code"  
 auto-complete="off"  
 placeholder="请输入验证码"  
 style="width: 60%"  
 @keyup.enter.native="handleLogin"  
 ></el-input>  
<!-- <div class="login-code">-->  
<!-- <img :src="codeUrl" @click="getCode" class="login-code-img"/>-->  
<!-- </div>-->  
 <div class="login-code" v-if="showCodeLoading || captchaEnabled">  
 <img :src="codeUrl" @click="getCode" class="login-code-img" v-show="!showCodeLoading"/>  
 <div v-if="showCodeLoading" class="loading-indicator">Loading...</div>  
 </div>  
 </el-form-item>  
  
  
 <div class="other-select">  
 <div class="rem-pwd">  
 <input type="checkbox" id="check">  
 <label for="check" class="rem-pwd-tips" v-model="loginForm.rememberMe">记住密码</label>  
 </div>  
 <!-- <span class="fogot-pwd-btn">忘记密码？</span>-->  
 </div>  
 <el-form-item style="width:100%;">  
 <el-button class="btn"  
 :loading="loading"  
 size="medium"  
 type="primary"  
 @click.native.prevent="handleLogin"  
 >  
 <span v-if="!loading">登 录</span>  
 <span v-else>登 录 中...</span>  
 </el-button>  
 </el-form-item>  
 </div>  
 </el-form>  
 </div>  
 </div>  
 </div>  
 </div>  
</template>  
  
<script>  
import { getCodeImg } from "@/api/login";  
import Cookies from "js-cookie";  
import { encrypt, decrypt } from '@/utils/jsencrypt'  
  
export default {  
 name: "Login",  
 data() {  
 return {  
 codeUrl: "",  
 loginForm: {  
 username: "admin",  
 password: "admin123",  
 rememberMe: false,  
 code: "",  
 uuid: ""  
 },  
 loginRules: {  
 username: [  
 { required: true, trigger: "blur", message: "请输入您的账号" }  
 ],  
 password: [  
 { required: true, trigger: "blur", message: "请输入您的密码" }  
 ],  
 code: [{ required: true, trigger: "change", message: "请输入验证码" }]  
 },  
 loading: false,  
 // 验证码开关  
 captchaEnabled: true,  
 // 注册开关  
 register: true,  
 redirect: undefined,  
 showCodeLoading: false // 控制加载动画显示  
 };  
 },  
 watch: {  
 $route: {  
 handler: function(route) {  
 this.redirect = route.query && route.query.redirect;  
 // 路由变化时重新获取验证码  
 this.getCode();  
 },  
 immediate: true  
 }  
 },  
 created() {  
 // this.getCode();  
 this.preloadCodeImage(); // 在组件创建时预加载验证码图片  
 this.getCookie();  
 },  
 methods: {  
 preloadCodeImage() {  
 // 在创建组件时预加载验证码图片  
 this.showCodeLoading = true;  
 getCodeImg().then(res => {  
 this.captchaEnabled = res.captchaEnabled === undefined ? true : res.captchaEnabled;  
 if (this.captchaEnabled) {  
 this.codeUrl = "data:image/gif;base64," + res.img;  
 this.loginForm.uuid = res.uuid;  
 }  
 this.showCodeLoading = false;  
 });  
 },  
 getCode() {  
 getCodeImg().then(res => {  
 this.captchaEnabled = res.captchaEnabled === undefined ? true : res.captchaEnabled;  
 if (this.captchaEnabled) {  
 this.codeUrl = "data:image/gif;base64," + res.img;  
 this.loginForm.uuid = res.uuid;  
 }  
 });  
 },  
 getCookie() {  
 const username = Cookies.get("username");  
 const password = Cookies.get("password");  
 const rememberMe = Cookies.get('rememberMe')  
 this.loginForm = {  
 username: username === undefined ? this.loginForm.username : username,  
 password: password === undefined ? this.loginForm.password : decrypt(password),  
 rememberMe: rememberMe === undefined ? false : Boolean(rememberMe)  
 };  
 },  
 handleLogin() {  
 this.$refs.loginForm.validate(valid => {  
 if (valid) {  
 this.loading = true;  
 if (this.loginForm.rememberMe) {  
 Cookies.set("username", this.loginForm.username, { expires: 30 });  
 Cookies.set("password", encrypt(this.loginForm.password), { expires: 30 });  
 Cookies.set('rememberMe', this.loginForm.rememberMe, { expires: 30 });  
 } else {  
 Cookies.remove("username");  
 Cookies.remove("password");  
 Cookies.remove('rememberMe');  
 }  
 this.$store.dispatch("Login", this.loginForm).then(() => {  
 this.$router.push({ path: this.redirect || "/" }).catch(()=>{});  
 }).catch(() => {  
 this.loading = false;  
 if (this.captchaEnabled) {  
 this.getCode();  
 }  
 });  
 }  
 });  
 }  
 }  
};  
</script>  
  
<style rel="stylesheet/scss" lang="scss">  
@charset "UTF-8";  
@font-face {  
 font-family: NeueMachina-Regular;  
 font-weight: 400;  
 src: url(./login/font/NeueMachina-Regular.woff2) format("truetype");  
}  
.loading-indicator {  
 display: flex;  
 justify-content: center;  
 align-items: center;  
 height: 60px;  
 color: #fff;  
 font-size: 16px;  
}  
\* {  
 padding: 0;  
 margin: 0;  
}  
  
.all {  
 background-color: rgb(22, 30, 49);  
 width: 100vw;  
 height: 100vh;  
 /\* background: url(./login/bg\_hnust.jpg); \*/  
}  
  
.box {  
 width: 80vw;  
 height: 80vh;  
 background: url(./login/bg\_hnust.jpg);  
 background-attachment: fixed;  
 background-size: cover;  
 position: absolute;  
 left: 50%;  
 top: 50%;  
 transform: translate(-50%, -50%);  
 border-radius: 30px;  
 font-family: NeueMachina-Regular;  
 overflow: hidden;  
}  
.box .forms {  
 width: 50%;  
 height: 100%;  
 background-color: rgb(45, 50, 80);  
 border-radius: 30px 0 0 30px;  
 color: rgb(255, 255, 255);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 box-sizing: border-box;  
 position: relative;  
}  
.box .forms .tips {  
 width: 70%;  
 height: 10%;  
 box-sizing: border-box;  
 font-size: 25px;  
 margin: 4vh;  
}  
.box .forms .tips span {  
 margin: 0 20px 0 0;  
 cursor: pointer;  
 padding-bottom: 5px;  
 transition: 0.1s;  
 color: rgb(100, 108, 154);  
}  
.box .forms .tips span:hover {  
 border-bottom: 4px solid rgb(115, 150, 250);  
 color: #fff;  
}  
.box .forms .tips .log-login-btn {  
 border-bottom: 4px solid rgb(115, 150, 250);  
 color: #fff;  
}  
.box .forms .tips .log-register-btn {  
 border-bottom: 0 solid rgb(115, 150, 250);  
}  
.box .forms .login {  
 width: 70%;  
 top: 20%;  
}  
.box .forms .login .form-title h2 {  
 font-size: 28px;  
 margin-top: -10px;  
}  
.box .forms .login .form-title h1 {  
 font-size: 45px;  
 margin-bottom: -10px;  
}  
.box .forms .login .form {  
 margin-top: 3vw;  
}  
//.box .forms .login .form .username::before {  
// content: "UserName";  
//}  
//.box .forms .login .form .password::before {  
// content: "Password";  
//}  
.login-code {  
 width: 35%;  
 height: 38px;  
 float: right;  
 img {  
 width: 100%;  
 height: 60px;  
 border-radius: 5px;  
 cursor: pointer;  
 vertical-align: middle;  
 }  
}  
.box .forms .login .form .code {  
 position: relative;  
}  
//.box .forms .login .form .code::before {  
// content: "Code";  
//}  
.box .forms .login .form .code .veri-code-tips {  
 position: absolute;  
 right: 20px;  
 top: 36px;  
 color: rgb(113, 123, 185);  
 cursor: pointer;  
}  
.form{  
 .el-input{  
 height: 60px;  
 input{  
 height: 60px;  
 }  
 }  
}  
.box .forms .login .form .input-item {  
 width: 100%;  
 height: 70px;  
 margin: 10px 0;  
 transition: 0.5s;  
}  
.box .forms .login .form .input-item::before {  
 display: block;  
 position: absolute;  
 color: rgb(100, 108, 154);  
 padding: 10px 5px 1px 20px;  
 font-size: 14px;  
}  
.box .forms .login .form .input-item input {  
 width: 100%;  
 height: 100%;  
 background-color: rgb(66, 71, 105);  
 outline: none;  
 border: 0;  
 padding: 20px 5px 1px 20px;  
 box-sizing: border-box;  
 color: #fff;  
 font-size: 20px;  
 transition: none; /\* 移除过渡效果 \*/  
}  
.box .forms .login .form .input-item input:focus {  
 border-left: 3px solid rgb(115, 150, 250);  
}  
.box .forms .login .form .other-select {  
 display: flex;  
 justify-content: space-between;  
 color: rgb(100, 108, 154);  
 margin-top: 1vw;  
}  
.box .forms .login .form .other-select #check:checked + .rem-pwd-tips {  
 color: #fff;  
}  
.box .forms .login .form .other-select input[type=checkbox] {  
 width: 20px;  
 height: 20px;  
 margin-right: 10px;  
 position: relative;  
 border: 0;  
}  
.box .forms .login .form .other-select input[type=checkbox]::after {  
 position: absolute;  
 top: 0;  
 color: #000;  
 width: 20px;  
 height: 20px;  
 display: inline-block;  
 visibility: visible;  
 padding-left: 0;  
 text-align: center;  
 background-color: rgb(66, 71, 105);  
 content: " ";  
}  
.box .forms .login .form .other-select input[type=checkbox]:checked::after {  
 content: "✓";  
 color: rgb(115, 150, 250);  
 font-size: 30px;  
 line-height: 20px;  
 background-color: rgb(66, 71, 105);  
 border: 1px solid rgb(115, 150, 250);  
 border-radius: 5px;  
}  
.box .forms .login .form .other-select .rem-pwd .rem-pwd-tips {  
 cursor: pointer;  
}  
.box .forms .login .form .other-select .fogot-pwd-btn {  
 cursor: pointer;  
 color: rgb(100, 108, 154);  
}  
.box .forms .login .form .btn {  
 width: 100%;  
 height: 50px;  
 cursor: pointer;  
 border: 0;  
 background-color: rgb(115, 150, 250);  
 color: rgb(45, 50, 80);  
 margin: 10px 0;  
 border-radius: 5px;  
 font-size: 20px;  
 box-shadow: 0px 0px 10px rgb(115, 150, 250);  
 transition: 0.3s;  
 margin-top: 1vw;  
 font-family: NeueMachina-Regular;  
}  
.box .forms .login .form .btn:hover {  
 box-shadow: 0px 0px 15px rgb(150, 180, 240);  
}  
.box .forms .login .form .other-login {  
 color: rgb(100, 108, 154);  
 margin-top: 5vw;  
 font-weight: 900;  
 display: flex;  
 align-items: baseline;  
}  
.box .forms .login .form .other-login .login-img img {  
 width: 40px;  
 height: 40px;  
 vertical-align: middle;  
 margin-left: 10px;  
 border-radius: 5px;  
 cursor: pointer;  
 vertical-align: middle;  
}  
.box .forms .login .form .other-login span {  
 display: inline-block;  
 margin-right: 2vw;  
}  
.box .forms .login {  
 position: absolute;  
 opacity: 1;  
 z-index: 10;  
 animation: into 1s ease;  
}  
@media (max-width: 1024px) {  
 .box {  
 width: 100vw;  
 height: 100vh;  
 border-radius: 0;  
 }  
 .box .forms {  
 width: 100%;  
 border-radius: 0;  
 }  
 .box .forms .other-select {  
 justify-content: flex-end !important;  
 }  
 .box .forms .other-select .rem-pwd {  
 display: none;  
 }  
 .box .forms .other-login {  
 flex-direction: column;  
 position: relative;  
 justify-content: center;  
 }  
 .box .forms .other-login .login-img {  
 position: absolute;  
 right: 0;  
 top: 5vh;  
 }  
}  
@keyframes into {  
 0% {  
 transform: translateY(5vh);  
 opacity: 0;  
 }  
 100% {  
 transform: translateY(0px);  
 opacity: 1;  
 }  
}  
  
</style>

===================================register.Vue================================

<template>  
 <div class="all">  
 <div class="box">  
 <div class="forms">  
 <div class="tips">  
 <span class="reg-login-btn" v-cloak><router-link :to="'/login'">登录界面</router-link></span>  
 <span class="reg-register-btn" >注册界面</span>  
 </div>  
 <div class="register">  
<!-- <div class="form-title">-->  
<!-- <h3 style="font-weight: bold">H N <span style="color: red;">U </span>S T</h3>-->  
<!-- <h1 style="font-weight: bold">特种设备管理系统</h1>-->  
<!-- </div>-->  
 <el-form ref="registerForm" :model="registerForm" :rules="registerRules">  
 <div class="form">  
 <el-form-item class="username input-item" prop="username">  
 <el-input class="ipts"  
 v-model="registerForm.username"  
 type="text"  
 auto-complete="off"  
 placeholder="请输入用户名"  
 >  
 </el-input>  
 </el-form-item>  
 <el-form-item class="password input-item" prop="password">  
 <el-input class="ipts"  
 v-model="registerForm.password"  
 type="password"  
 auto-complete="off"  
 placeholder="请输入密码"  
 @keyup.enter.native="handleRegister"  
 ></el-input>  
 </el-form-item>  
 <el-form-item class="password input-item" prop="confirmPassword">  
 <el-input class="ipts"  
 v-model="registerForm.confirmPassword"  
 type="password"  
 auto-complete="off"  
 placeholder="请输入确认密码"  
 @keyup.enter.native="handleRegister"  
 ></el-input>  
 </el-form-item>  
 <el-form-item class="phone input-item" prop="phonenumber">  
 <el-input class="ipts"  
 v-model="registerForm.phonenumber"  
 type="text"  
 auto-complete="off"  
 placeholder="请输入手机号码"  
 @keyup.enter.native="handleRegister"  
 ></el-input>  
 </el-form-item>  
 <!-- 验证码-->  
 <el-form-item class="code input-item" prop="code" v-if="captchaEnabled">  
 <el-input class="ipts" v-model="registerForm.code" auto-complete="off" placeholder="请输入验证码" @keyup.enter.native="handleRegister">  
 <template v-slot:append>  
 <!-- 将获取验证码按钮放置在输入框后面 -->  
 <el-button class="code-input-btn" v-if="isCode" @click="getCode" style="width: 160px; font-size: 20px">获取验证码</el-button>  
 <el-button class="code-input-btn" v-else :disabled="isDisabled" style="width: 160px; font-size: 20px">{{ second }}秒后获取</el-button>  
 </template>  
 </el-input>  
 </el-form-item>  
 <el-form-item style="width:100%;">  
 <el-button class="btn"  
 size="medium"  
 type="primary"  
 @click.native.prevent="handleRegister"  
 >  
 <span v-if="!loading">注 册</span>  
 <span v-else>注 册 中...</span>  
 </el-button>  
 </el-form-item>  
 </div>  
 </el-form>  
 </div>  
 </div>  
 </div>  
 </div>  
</template>  
  
<script>  
import {getCodeData, getCodeImg, register} from "@/api/login";  
  
export default {  
 name: "Register",  
 data() {  
 const equalToPassword = (rule, value, callback) => {  
 if (this.registerForm.password !== value) {  
 callback(new Error("两次输入的密码不一致"));  
 } else {  
 callback();  
 }  
 };  
 return {  
 codeUrl: "",  
 registerForm: {  
 username: "",  
 password: "",  
 confirmPassword: "",  
 phonenumber: "",  
 code: "",  
 uuid: "",  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 registerRules: {  
 username: [  
 { required: true, trigger: "blur", message: "请输入您的用户名" },  
 { min: 2, max: 16, message: '用户账号长度必须介于 2 和 16 之间', trigger: 'blur' }  
 ],  
 password: [  
 { required: true, trigger: "blur", message: "请输入您的密码" },  
 { min: 8, max: 16, pattern: /^(?=.\*[a-zA-Z])(?=.\*\d)[a-zA-Z0-9]{8,16}$/, message: '用户密码长度必须介于 8 和 16 之间，且必须包含字母和数字以及不包含空格', trigger: 'blur' },  
 ],  
 confirmPassword: [  
 { required: true, trigger: "blur", message: "请再次输入您的密码" },  
 { required: true, validator: equalToPassword, trigger: "blur" }  
 ],  
 phonenumber: [  
 { required: true, trigger: "blur", message: "请输入您的手机号码" },  
 { required: true, pattern: /^1[3|4|5|6|7|8|9][0-9]\d{8}$/, message: "请输入正确的手机号码", trigger: "blur"}  
 ],  
 code: [{ required: true, trigger: "change", message: "请输入验证码" }]  
 },  
 loading: false,  
 captchaEnabled: true,  
 second: 5, // 用于倒计时  
 isDisabled: false,  
 isCode: true,  
 test: 0,  
 };  
 },  
 created() {  
 },  
 methods: {  
 startCountdown() {  
 this.isCode = false;  
 this.isDisabled = true;  
 let result = setInterval(()=>{  
 --this.second;  
 if(this.second < 0) {  
 clearInterval(result);  
 this.isCode = true;  
 this.isDisabled = false;  
 this.second = 5;  
 }  
 }, 1000);  
 },  
 getCode() {  
 console.log("调用了发送验证码的方法");  
 this.startCountdown();  
 getCodeData().then(res => {  
 this.registerForm.uuid = res.uuid;  
 console.log(res.codeStr)  
 });  
 },  
 handleRegister() {  
 this.$refs.registerForm.validate(valid => {  
 if (valid) {  
 this.loading = true;  
 register(this.registerForm).then(res => {  
 const username = this.registerForm.username;  
 this.$alert("<font color='red'>恭喜你，您的账号 " + username + " 注册成功！</font>", '系统提示', {  
 dangerouslyUseHTMLString: true,  
 type: 'success'  
 }).then(() => {  
 this.$router.push("/login");  
 }).catch(() => {});  
 }).catch(() => {  
 this.loading = false;  
 // if (this.captchaEnabled) {  
 // this.getCode();  
 // }  
 })  
 }  
 });  
 }  
 }  
};  
</script>  
  
<style rel="stylesheet/scss" lang="scss">  
@charset "UTF-8";  
@font-face {  
 font-family: NeueMachina-Regular;  
 font-weight: 400;  
 src: url(./login/font/NeueMachina-Regular.woff2) format("truetype");  
}  
//.code-input-wrapper {  
// display: flex;  
// align-items: center;  
//}  
//.code-input-wrapper .ipts {  
// width: 100%;  
// margin-right: 20px;  
// height: 50px;  
//}  
//.code-input-wrapper-btn {  
// width: 45%;  
// height: 50px;  
// cursor: pointer;  
// border: 0;  
// background-color: rgb(115, 150, 250);  
// color: rgb(45, 50, 80);  
// border-radius: 5px;  
// font-size: 20px;  
// box-shadow: 0px 0px 5px rgb(115, 150, 250);  
// transition: 0.3s;  
// font-family: NeueMachina-Regular;  
//}  
//.code-input-wrapper .code-btn:hover {  
// box-shadow: 0px 0px 15px rgb(150, 180, 240);  
//}  
.box .forms .register .form .code-input-btn{  
 //width: 12vw;  
 width: 60%;  
 font-size: 20px;  
}  
\* {  
 padding: 0;  
 margin: 0;  
}  
  
.all {  
 background-color: rgb(22, 30, 49);  
 width: 100vw;  
 height: 100vh;  
 /\* background: url(./login/bg\_hnust.jpg); \*/  
}  
  
.box {  
 width: 80vw;  
 height: 80vh;  
 background: url(./login/bg\_hnust.jpg);  
 background-attachment: fixed;  
 background-size: cover;  
 position: absolute;  
 left: 50%;  
 top: 50%;  
 transform: translate(-50%, -50%);  
 border-radius: 30px;  
 font-family: NeueMachina-Regular;  
 overflow: hidden;  
}  
.box .forms {  
 width: 50%;  
 height: 100%;  
 background-color: rgb(45, 50, 80);  
 border-radius: 30px 0 0 30px;  
 color: rgb(255, 255, 255);  
 display: flex;  
 flex-direction: column;  
 align-items: center;  
 box-sizing: border-box;  
 position: relative;  
}  
.box .forms .tips {  
 width: 70%;  
 height: 10%;  
 box-sizing: border-box;  
 font-size: 25px;  
 margin: 4vh;  
}  
.box .forms .tips span {  
 margin: 0 20px 0 0;  
 cursor: pointer;  
 padding-bottom: 5px;  
 transition: 0.1s;  
 color: rgb(100, 108, 154);  
}  
.box .forms .tips span:hover {  
 border-bottom: 4px solid rgb(115, 150, 250);  
 color: #fff;  
}  
.box .forms .tips .reg-login-btn {  
 border-bottom: 0 solid rgb(115, 150, 250);  
}  
.box .forms .tips .reg-register-btn {  
 border-bottom: 4px solid rgb(115, 150, 250);  
 color: #fff;  
}  
.box .forms .register {  
 width: 70%;  
 top: 15%;  
}  
.box .forms .register .form-title h1 {  
 margin-bottom: 10px;  
}  
.box .forms .register .form {  
 width: 100%;  
 margin-top: 3vw;  
}  
//.box .forms .register .form .username::before {  
// content: "UserName";  
//}  
//.box .forms .register .form .password::before {  
// content: "Password";  
//}  
//.box .forms .register .form .conform-password::before {  
// content: "Conform Password";  
//}  
.register-code {  
 width: 35%;  
 height: 38px;  
 float: right;  
 img {  
 width: 100%;  
 height: 60px;  
 border-radius: 5px;  
 cursor: pointer;  
 vertical-align: middle;  
 }  
}  
.box .forms .register .form .code {  
 position: relative;  
}  
//.box .forms .register .form .code::before {  
// content: "Code";  
//}  
.box .forms .register .form .code .veri-code-tips {  
 position: absolute;  
 right: 20px;  
 top: 36px;  
 color: rgb(113, 123, 185);  
 cursor: pointer;  
}  
.form{  
 .el-input{  
 height: 60px;  
 input{  
 height: 60px;  
 }  
 }  
}  
.box .forms .register .form .input-item {  
 width: 100%;  
 height: 70px;  
 margin: 10px 0;  
 transition: 0.5s;  
}  
.box .forms .register .form .input-item::before {  
 display: block;  
 position: absolute;  
 color: rgb(100, 108, 154);  
 padding: 10px 5px 1px 20px;  
 font-size: 14px;  
}  
.box .forms .register .form .input-item input {  
 width: 100%;  
 height: 100%;  
 background-color: rgb(66, 71, 105);  
 outline: none;  
 border: 0;  
 padding: 20px 5px 1px 20px;  
 box-sizing: border-box;  
 color: #fff;  
 font-size: 20px;  
 transition: none; /\* 移除过渡效果 \*/  
}  
.box .forms .register .form .input-item input:focus {  
 border-left: 3px solid rgb(115, 150, 250);  
}  
.box .forms .register .form .other-select {  
 display: flex;  
 justify-content: space-between;  
 color: rgb(100, 108, 154);  
 margin-top: 1vw;  
}  
.box .forms .register .form .other-select #check:checked + .rem-pwd-tips {  
 color: #fff;  
}  
.box .forms .register .form .other-select input[type=checkbox] {  
 width: 20px;  
 height: 20px;  
 margin-right: 10px;  
 position: relative;  
 border: 0;  
}  
.box .forms .register .form .other-select input[type=checkbox]::after {  
 position: absolute;  
 top: 0;  
 color: #000;  
 width: 20px;  
 height: 20px;  
 display: inline-block;  
 visibility: visible;  
 padding-left: 0;  
 text-align: center;  
 background-color: rgb(66, 71, 105);  
 content: " ";  
}  
.box .forms .register .form .other-select input[type=checkbox]:checked::after {  
 content: "✓";  
 color: rgb(115, 150, 250);  
 font-size: 30px;  
 line-height: 20px;  
 background-color: rgb(66, 71, 105);  
 border: 1px solid rgb(115, 150, 250);  
 border-radius: 5px;  
}  
.box .forms .register .form .other-select .rem-pwd .rem-pwd-tips {  
 cursor: pointer;  
}  
.box .forms .register .form .other-select .fogot-pwd-btn {  
 cursor: pointer;  
 color: rgb(100, 108, 154);  
}  
.box .forms .register .form .btn {  
 width: 100%;  
 height: 50px;  
 cursor: pointer;  
 border: 0;  
 background-color: rgb(115, 150, 250);  
 color: rgb(45, 50, 80);  
 margin: 10px 0;  
 border-radius: 5px;  
 font-size: 20px;  
 box-shadow: 0px 0px 10px rgb(115, 150, 250);  
 transition: 0.3s;  
 margin-top: 1vw;  
 font-family: NeueMachina-Regular;  
}  
.box .forms .register .form .btn:hover {  
 box-shadow: 0px 0px 15px rgb(150, 180, 240);  
}  
.box .forms .register .form .other-login {  
 color: rgb(100, 108, 154);  
 margin-top: 5vw;  
 font-weight: 900;  
 display: flex;  
 align-items: baseline;  
}  
.box .forms .register .form .other-login .login-img img {  
 width: 40px;  
 height: 40px;  
 vertical-align: middle;  
 margin-left: 10px;  
 border-radius: 5px;  
 cursor: pointer;  
 vertical-align: middle;  
}  
.box .forms .register .form .other-login span {  
 display: inline-block;  
 margin-right: 2vw;  
}  
.box .forms .register {  
 position: absolute;  
 opacity: 1;  
 z-index: 10;  
 animation: into 1s ease;  
}  
@media (max-width: 1024px) {  
 .box {  
 width: 100vw;  
 height: 100vh;  
 border-radius: 0;  
 }  
 .box .forms {  
 width: 100%;  
 border-radius: 0;  
 }  
 .box .forms .other-select {  
 justify-content: flex-end !important;  
 }  
 .box .forms .other-select .rem-pwd {  
 display: none;  
 }  
 .box .forms .other-login {  
 flex-direction: column;  
 position: relative;  
 justify-content: center;  
 }  
 .box .forms .other-login .login-img {  
 position: absolute;  
 right: 0;  
 top: 5vh;  
 }  
}  
@keyframes into {  
 0% {  
 transform: translateY(5vh);  
 opacity: 0;  
 }  
 100% {  
 transform: translateY(0px);  
 opacity: 1;  
 }  
}  
</style>

===================================index.Vue(lift)=================================

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="queryParams.deptId" placeholder="请选择分布单位" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="使用地点" prop="location">  
 <el-input  
 v-model="queryParams.location"  
 placeholder="请输入使用地点"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input  
 v-model="queryParams.administrator"  
 placeholder="请输入管理人员"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="设备名称" prop="liftName">  
 <el-input  
 v-model="queryParams.liftName"  
 placeholder="请输入设备名称"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
<!-- <el-form-item label="年检类型" prop="checkType">-->  
<!-- <el-select v-model="queryParams.checkType" placeholder="请选择年检类型" clearable>-->  
<!-- <el-option-->  
<!-- v-for="dict in dict.type.equipment\_check\_type\_lift"-->  
<!-- :key="dict.value"-->  
<!-- :label="dict.label"-->  
<!-- :value="dict.value"-->  
<!-- />-->  
<!-- </el-select>-->  
<!-- </el-form-item>-->  
<!-- <el-form-item label="维保单位" prop="maintenanceId">-->  
<!-- <el-select v-model="queryParams.maintenanceId" placeholder="请选择维保单位" clearable>-->  
<!-- <el-option-->  
<!-- v-for="dict in dict.type.equipment\_maintenance\_lift"-->  
<!-- :key="dict.value"-->  
<!-- :label="dict.label"-->  
<!-- :value="dict.value"-->  
<!-- />-->  
<!-- </el-select>-->  
<!-- </el-form-item>-->  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="queryParams.status" placeholder="请选择状态" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['equipment:lift:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['equipment:lift:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['equipment:lift:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['equipment:lift:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" :columns="tableColumns" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="liftList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="序号ID" align="center" prop="liftId" v-if="isShowTableColumns('liftId')"/>  
 <el-table-column label="分布单位" align="center" prop="deptId" v-if="isShowTableColumns('deptId')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_dept\_id" :value="scope.row.deptId"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="使用地点" align="center" prop="location" v-if="isShowTableColumns('location')"/>  
<!-- <el-table-column label="注册代码" align="center" prop="regcode" v-if="isShowTableColumns('regcode')"/>-->  
<!-- <el-table-column label="设备代码" align="center" prop="code" v-if="isShowTableColumns('code')"/>-->  
 <el-table-column label="管理人员" align="center" prop="administrator" v-if="isShowTableColumns('administrator')"/>  
 <el-table-column label="联系方式" align="center" prop="phone" v-if="isShowTableColumns('phone')"/>  
 <el-table-column label="设备编码" align="center" prop="number" v-if="isShowTableColumns('number')"/>  
 <el-table-column label="设备名称" align="center" prop="liftName" v-if="isShowTableColumns('liftName')"/>  
<!-- <el-table-column label="购置日期" align="center" prop="buyTime" width="180" v-if="isShowTableColumns('buyTime')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <span>{{ parseTime(scope.row.buyTime, '{y}-{m}-{d}') }}</span>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
<!-- <el-table-column label="额定载程(kg)" align="center" prop="ratedLoad" v-if="isShowTableColumns('ratedLoad')"/>-->  
<!-- <el-table-column label="额定速度(m/s)" align="center" prop="ratedSpeed" v-if="isShowTableColumns('ratedSpeed')"/>-->  
<!-- <el-table-column label="层站" align="center" prop="floor" v-if="isShowTableColumns('floor')"/>-->  
 <el-table-column label="年检类型" align="center" prop="checkType" v-if="isShowTableColumns('checkType')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_check\_type\_lift" :value="scope.row.checkType"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="下次年检时间" align="center" prop="checkTime" width="180" v-if="isShowTableColumns('checkTime')">  
 <template slot-scope="scope">  
 <span>{{ parseTime(scope.row.checkTime, '{y}-{m}-{d}') }}</span>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="维保单位" align="center" prop="maintenanceId" v-if="isShowTableColumns('maintenanceId')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <dict-tag :options="dict.type.equipment\_maintenance\_lift" :value="scope.row.maintenanceId"/>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
<!-- <el-table-column label="是否鼓式制动机" align="center" prop="isDrum" v-if="isShowTableColumns('isDrum')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <dict-tag :options="dict.type.equipment\_is\_drum" :value="scope.row.isDrum"/>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
 <el-table-column label="状态" align="center" prop="status" v-if="isShowTableColumns('status')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_status" :value="scope.row.status"/>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="备注" align="center" prop="remark" v-if="isShowTableColumns('remark')"/>-->  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['equipment:lift:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['equipment:lift:remove']"  
 >删除</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-search"  
 @click="handleDetails(scope.row)"  
 v-hasPermi="['equipment:lift:edit']"  
 >详细信息</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加、修改和查看电梯信息对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px" :disabled="isDisabled">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="form.deptId" :placeholder="getPlaceholder('请选择分布单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="使用地点" prop="location">  
 <el-input v-model="form.location" :placeholder="getPlaceholder('请输入使用地点', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="注册代码" prop="regcode">  
 <el-input v-model="form.regcode" :placeholder="getPlaceholder('请输入注册代码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备代码" prop="code">  
 <el-input v-model="form.code" :placeholder="getPlaceholder('请输入设备代码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input v-model="form.administrator" :placeholder="getPlaceholder('请输入管理人员', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="联系方式" prop="phone">  
 <el-input v-model="form.phone" :placeholder="getPlaceholder('请输入联系方式', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备编码" prop="number">  
 <el-input v-model="form.number" :placeholder="getPlaceholder('请输入设备编码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备名称" prop="liftName">  
 <el-input v-model="form.liftName" :placeholder="getPlaceholder('请输入设备名称', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="购置日期" prop="buyTime">  
 <el-date-picker clearable  
 v-model="form.buyTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择购置日期')">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="额定载程(kg)" prop="ratedLoad">  
 <el-input v-model="form.ratedLoad" :placeholder="getPlaceholder('请输入额定载程', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="额定速度(m/s)" prop="ratedSpeed">  
 <el-input v-model="form.ratedSpeed" :placeholder="getPlaceholder('请输入额定速度', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="层站" prop="floor">  
 <el-input v-model="form.floor" :placeholder="getPlaceholder('请输入层站', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="年检类型" prop="checkType" v-if="isShowCheckType" clearable>  
 <el-select v-model="form.checkType" :placeholder="getPlaceholder('请选择年检类型', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_check\_type\_lift"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="下次年检时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="form.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择下次年检时间', isShowPlaceholder)">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="维保单位" prop="maintenanceId">  
 <el-select v-model="form.maintenanceId" :placeholder="getPlaceholder('请选择维保单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_maintenance\_lift"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="是否鼓式制动机" prop="isDrum">  
 <el-select v-model="form.isDrum" :placeholder="getPlaceholder('请选择是否鼓式制动机', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_is\_drum"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="form.status" :placeholder="getPlaceholder('请选择状态', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" type="textarea" :placeholder="getPlaceholder('请输入备注', isShowPlaceholder)"/>  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer" v-if="isShowSubmission">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
  
 </div>  
</template>  
  
<script>  
import { listLift, getLift, delLift, addLift, updateLift } from "@/api/equipment/lift";  
  
export default {  
 name: "Lift",  
 dicts: ['equipment\_status', 'equipment\_check\_type\_lift', 'equipment\_maintenance\_lift', 'equipment\_dept\_id', 'equipment\_is\_drum'],  
 data() {  
 return {  
 // 显隐列  
 tableColumns: [  
 { key: 'liftId', label: '序号ID', visible: true },  
 { key: 'deptId', label: '分布单元', visible: true },  
 { key: 'location', label: '使用地点', visible: true },  
 // { key: 'regcode', label: '注册代码', visible: true },  
 // { key: 'code', label: '设备代码', visible: true },  
 { key: 'administrator', label: '管理人员', visible: true },  
 { key: 'phone', label: '联系方式', visible: true },  
 { key: 'number', label: '设备编码', visible: true },  
 { key: 'liftName', label: '设备名称', visible: true },  
 // { key: 'buyTime', label: '购置日期', visible: true },  
 // { key: 'ratedLoad', label: '额定载程', visible: true },  
 // { key: 'ratedSpeed', label: '额定速度', visible: true },  
 // { key: 'floor', label: '层站', visible: true },  
 { key: 'checkType', label: '年检类型', visible: true },  
 { key: 'checkTime', label: '下次年检时间', visible: true },  
 // { key: 'maintenanceId', label: '维保单位', visible: true },  
 // { key: 'isDrum', label: '是否鼓式制动机', visible: true },  
 { key: 'status', label: '状态', visible: true },  
 // { key: 'remark', label: '备注', visible: true },  
 ],  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 电梯管理表格数据  
 liftList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示增添和修改弹出层  
 open: false,  
 // 进行查看设备详细信息时，各组件disabled属性总控  
 isDisabled: false,  
 // 是否展示增改界面的确定与取消键  
 isShowSubmission: true,  
 // 是否展示检验类型  
 isShowCheckType: false,  
 // 是否展示placeholder属性  
 isShowPlaceholder: true,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 location: null,  
 administrator: null,  
 liftName: null,  
 checkType: null,  
 maintenanceId: null,  
 status: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 deptId: [  
 { required: true, message: "分布单位不能为空", trigger: "change" }  
 ],  
 location: [  
 { required: true, message: "使用地点不能为空", trigger: "blur" }  
 ],  
 status: [  
 { required: true, message: "状态不能为空", trigger: "change" }  
 ],  
 administrator: [  
 { required: true, message: "输入格式：管理人员1，管理人员2", trigger: ["change", "blur"] }  
 ],  
 phone: [  
 { required: true, message: "输入格式：联系方式1/联系方式2，联系方式3/联系方式4", trigger: ["change", "blur"] }  
 ]  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 其他方法 \*/  
 // 是否展示placeholder  
 getPlaceholder(input, isShowPlaceholder) {  
 // 通过isShowPlaceholder来控制是否展示placeholder  
 return isShowPlaceholder ? (input): '';  
 },  
 // 是否显隐列  
 isShowTableColumns(keyToFind) {  
 const column = this.tableColumns.find(column => column.key === keyToFind);  
 return column.visible;  
 },  
 /\*\* 查询电梯管理列表 \*/  
 getList() {  
 this.loading = true;  
 listLift(this.queryParams).then(response => {  
 this.liftList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 liftId: null,  
 deptId: null,  
 location: null,  
 regcode: null,  
 code: null,  
 administrator: null,  
 phone: null,  
 number: null,  
 liftName: null,  
 buyTime: null,  
 ratedLoad: null,  
 ratedSpeed: null,  
 floor: null,  
 checkType: null,  
 checkTime: null,  
 maintenanceId: null,  
 isDrum: null,  
 status: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.liftId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "添加电梯信息";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const liftId = row.liftId || this.ids  
 getLift(liftId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "修改电梯信息";  
 });  
 },  
 /\*\* 详情按钮操作 \*/  
 handleDetails(row) {  
 this.reset();  
 const liftId = row.liftId || this.ids  
 getLift(liftId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = true;  
 this.isShowSubmission = false;  
 this.isShowCheckType = true;  
 this.isShowPlaceholder = false;  
 this.title = "查看电梯详细信息";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.liftId != null) {  
 updateLift(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addLift(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const liftIds = row.liftId || this.ids;  
 this.$modal.confirm('是否确认删除电梯管理编号为"' + liftIds + '"的数据项？').then(function() {  
 return delLift(liftIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('equipment/lift/export', {  
 ...this.queryParams  
 }, `lift\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

===================================index.Vue(check)==============================

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="对应年份" prop="checkYear">  
 <el-input  
 v-model="queryParams.checkYear"  
 placeholder="请输入对应年份"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="检验类型" prop="type">  
 <el-select v-model="queryParams.type" placeholder="请选择检验类型" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_check\_type"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['equipment:check:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['equipment:check:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['equipment:check:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['equipment:check:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="checkList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
<!-- <el-table-column label="序列ID" align="center" prop="checkId" />-->  
 <el-table-column label="对应年份" align="center" prop="checkYear" />  
 <el-table-column label="检验类型" align="center" prop="type">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_check\_type" :value="scope.row.type"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['equipment:check:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['equipment:check:remove']"  
 >删除</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加或修改检验类型管理对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px">  
 <el-form-item label="对应年份" prop="checkYear">  
 <el-input v-model="form.checkYear" placeholder="请输入对应年份" />  
 </el-form-item>  
 <el-form-item label="检验类型" prop="type">  
 <el-select v-model="form.type" placeholder="请选择检验类型">  
 <el-option  
 v-for="dict in dict.type.equipment\_check\_type"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" placeholder="请输入备注" />  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listCheck, getCheck, delCheck, addCheck, updateCheck } from "@/api/equipment/check";  
  
export default {  
 name: "Check",  
 dicts: ['equipment\_check\_type'],  
 data() {  
 return {  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 检验类型管理表格数据  
 checkList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 checkYear: null,  
 type: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 checkYear: [  
 { required: true, message: "对应年份不能为空", trigger: "blur" }  
 ],  
 type: [  
 { required: true, message: "检验类型不能为空", trigger: "change" }  
 ],  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 查询检验类型管理列表 \*/  
 getList() {  
 this.loading = true;  
 listCheck(this.queryParams).then(response => {  
 this.checkList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 checkId: null,  
 checkYear: null,  
 type: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.checkId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.title = "添加检验类型管理";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const checkId = row.checkId || this.ids  
 getCheck(checkId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.title = "修改检验类型管理";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.checkId != null) {  
 updateCheck(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addCheck(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const checkIds = row.checkId || this.ids;  
 this.$modal.confirm('是否确认删除检验类型管理编号为"' + checkIds + '"的数据项？').then(function() {  
 return delCheck(checkIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('equipment/check/export', {  
 ...this.queryParams  
 }, `check\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

===================================index.Vue(cost)===============================

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="申报人员" prop="user">  
 <el-input  
 v-model="queryParams.user"  
 placeholder="请输入申报人员名"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="报销类型" prop="costType">  
 <el-select v-model="queryParams.costType" placeholder="请选择报销类型" clearable>  
 <el-option  
 v-for="dict in dict.type.cost"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="上传时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="queryParams.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 placeholder="请选择上传时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['image:cost:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['image:cost:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['image:cost:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['image:cost:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="costList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="报销序列号" align="center" prop="costId" />  
 <el-table-column label="申报人员" align="center" prop="user" />  
 <el-table-column label="报销类型" align="center" prop="costType">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.cost" :value="scope.row.costType"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="报销凭证" align="center" prop="url" width="100">  
 <template slot-scope="scope">  
 <image-preview :src="scope.row.url" :width="50" :height="50"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="上传时间" align="center" prop="checkTime" width="180">  
 <template slot-scope="scope">  
 <span>{{ parseTime(scope.row.checkTime, '{y}-{m}-{d}') }}</span>  
 </template>  
 </el-table-column>  
 <el-table-column label="备注" align="center" prop="remark" />  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['image:cost:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['image:cost:remove']"  
 >删除</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加或修改报销凭证对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px">  
 <el-form-item label="报销类型" prop="costType">  
 <el-select v-model="form.costType" placeholder="请选择报销类型">  
 <el-option  
 v-for="dict in dict.type.cost"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="报销凭证" prop="url">  
 <image-upload v-model="form.url"/>  
 </el-form-item>  
 <el-form-item label="上传时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="form.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 placeholder="请选择上传时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" placeholder="请输入备注" />  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listCost, getCost, delCost, addCost, updateCost } from "@/api/image/cost";  
  
export default {  
 name: "Cost",  
 dicts: ['cost'],  
 data() {  
 return {  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 报销凭证表格数据  
 costList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 user: null,  
 costType: null,  
 checkTime: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 user: [  
 { required: true, message: "申报人员不能为空", trigger: "blur" }  
 ],  
 costType: [  
 { required: true, message: "报销类型不能为空", trigger: "change" }  
 ],  
 url: [  
 { required: true, message: "报销凭证不能为空", trigger: "blur" }  
 ],  
 checkTime: [  
 { required: true, message: "上传时间不能为空", trigger: "blur" }  
 ],  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 查询报销凭证列表 \*/  
 getList() {  
 this.loading = true;  
 listCost(this.queryParams).then(response => {  
 this.costList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 costId: null,  
 user: null,  
 costType: null,  
 url: null,  
 checkTime: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null,  
 userId: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.costId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.title = "添加报销凭证";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const costId = row.costId || this.ids  
 getCost(costId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.title = "修改报销凭证";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.costId != null) {  
 updateCost(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addCost(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const costIds = row.costId || this.ids;  
 this.$modal.confirm('是否确认删除报销凭证编号为"' + costIds + '"的数据项？').then(function() {  
 return delCost(costIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('image/cost/export', {  
 ...this.queryParams  
 }, `cost\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

===================================index.Vue(image)==============================<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="维修人员" prop="user">  
 <el-input  
 v-model="queryParams.user"  
 placeholder="请输入维修人员名"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="设备类型" prop="equipmentType">  
 <el-select v-model="queryParams.equipmentType" placeholder="请选择设备类型" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_type"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="维修位置" prop="location">  
 <el-select v-model="queryParams.location" placeholder="请选择维修位置" clearable>  
 <el-option  
 v-for="dict in dict.type.location"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="登记时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="queryParams.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 placeholder="请选择登记时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['image:image:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['image:image:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['image:image:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['image:image:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="imageList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="日志序列号" align="center" prop="logId" />  
 <el-table-column label="维修人员" align="center" prop="user" />  
 <el-table-column label="设备类型" align="center" prop="equipmentType">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_type" :value="scope.row.equipmentType"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="维修位置" align="center" prop="location">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.location" :value="scope.row.location"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="维修凭证" align="center" prop="url" width="100">  
 <template slot-scope="scope">  
 <image-preview :src="scope.row.url" :width="50" :height="50"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="登记时间" align="center" prop="checkTime" width="180">  
 <template slot-scope="scope">  
 <span>{{ parseTime(scope.row.checkTime, '{y}-{m}-{d}') }}</span>  
 </template>  
 </el-table-column>  
 <el-table-column label="备注" align="center" prop="remark" />  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['image:image:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['image:image:remove']"  
 >删除</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加或修改维修凭证对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px">  
 <el-form-item label="设备类型" prop="equipmentType">  
 <el-select v-model="form.equipmentType" placeholder="请选择设备类型">  
 <el-option  
 v-for="dict in dict.type.equipment\_type"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="维修位置" prop="location">  
 <el-select v-model="form.location" placeholder="请选择维修位置">  
 <el-option  
 v-for="dict in dict.type.location"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="维修凭证" prop="url">  
 <image-upload v-model="form.url"/>  
 </el-form-item>  
 <el-form-item label="登记时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="form.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 placeholder="请选择登记时间">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" placeholder="请输入备注" />  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listImage, getImage, delImage, addImage, updateImage } from "@/api/image/image";  
  
export default {  
 name: "Image",  
 dicts: ['equipment\_type', 'location'],  
 data() {  
 return {  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 维修凭证表格数据  
 imageList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 user: null,  
 equipmentType: null,  
 location: null,  
 checkTime: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 equipmentType: [  
 { required: true, message: "设备类型不能为空", trigger: "change" }  
 ],  
 location: [  
 { required: true, message: "维修位置不能为空", trigger: "change" }  
 ],  
 url: [  
 { required: true, message: "维修凭证不能为空", trigger: "blur" }  
 ],  
 checkTime: [  
 { required: true, message: "登记时间不能为空", trigger: "blur" }  
 ],  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 查询维修凭证列表 \*/  
 getList() {  
 this.loading = true;  
 listImage(this.queryParams).then(response => {  
 this.imageList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 logId: null,  
 user: null,  
 equipmentType: null,  
 location: null,  
 url: null,  
 checkTime: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null,  
 userId: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.logId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.title = "添加维修凭证";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const logId = row.logId || this.ids  
 getImage(logId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.title = "修改维修凭证";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.logId != null) {  
 updateImage(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addImage(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const logIds = row.logId || this.ids;  
 this.$modal.confirm('是否确认删除维修凭证编号为"' + logIds + '"的数据项？').then(function() {  
 return delImage(logIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('image/image/export', {  
 ...this.queryParams  
 }, `image\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

===================================index.Vue(crane)==============================

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="queryParams.deptId" placeholder="请选择分布单位" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="使用地点" prop="location">  
 <el-input  
 v-model="queryParams.location"  
 placeholder="请输入使用地点"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input  
 v-model="queryParams.administrator"  
 placeholder="请输入管理人员"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="设备名称" prop="craneName">  
 <el-input  
 v-model="queryParams.craneName"  
 placeholder="请输入设备名称"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
<!-- <el-form-item label="维保单位" prop="maintenanceId">-->  
<!-- <el-select v-model="queryParams.maintenanceId" placeholder="请选择维保单位" clearable>-->  
<!-- <el-option-->  
<!-- v-for="dict in dict.type.equipment\_maintenance\_crane"-->  
<!-- :key="dict.value"-->  
<!-- :label="dict.label"-->  
<!-- :value="dict.value"-->  
<!-- />-->  
<!-- </el-select>-->  
<!-- </el-form-item>-->  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="queryParams.status" placeholder="请选择状态" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['equipment:crane:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['equipment:crane:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['equipment:crane:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['equipment:crane:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" :columns="tableColumns" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="craneList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="序号ID" align="center" prop="craneId" v-if="isShowTableColumns('craneId')"/>  
 <el-table-column label="分布单位" align="center" prop="deptId" v-if="isShowTableColumns('deptId')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_dept\_id" :value="scope.row.deptId"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="使用地点" align="center" prop="location" v-if="isShowTableColumns('location')"/>  
<!-- <el-table-column label="注册代码" align="center" prop="regcode" v-if="isShowTableColumns('regcode')"/>-->  
<!-- <el-table-column label="设备代码" align="center" prop="code" v-if="isShowTableColumns('code')"/>-->  
 <el-table-column label="管理人员" align="center" prop="administrator" v-if="isShowTableColumns('administrator')"/>  
 <el-table-column label="联系方式" align="center" prop="phone" v-if="isShowTableColumns('phone')"/>  
 <el-table-column label="设备编码" align="center" prop="number" v-if="isShowTableColumns('number')"/>  
 <el-table-column label="设备名称" align="center" prop="craneName" v-if="isShowTableColumns('craneName')"/>  
<!-- <el-table-column label="购置日期" align="center" prop="buyTime" width="180" v-if="isShowTableColumns('buyTime')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <span>{{ parseTime(scope.row.buyTime, '{y}-{m}-{d}') }}</span>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
<!-- <el-table-column label="规格(T)" align="center" prop="weight" v-if="isShowTableColumns('weight')"/>-->  
<!-- <el-table-column label="生产厂家" align="center" prop="factory" v-if="isShowTableColumns('factory')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <dict-tag :options="dict.type.equipment\_maintenance\_crane" :value="scope.row.factory"/>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
<!-- <el-table-column label="型号规格" align="center" prop="type" v-if="isShowTableColumns('type')"/>-->  
<!-- <el-table-column label="起升高度(m)" align="center" prop="liftHeight" v-if="isShowTableColumns('liftHeight')"/>-->  
<!-- <el-table-column label="跨度(m)" align="center" prop="span" v-if="isShowTableColumns('span')"/>-->  
<!-- <el-table-column label="起升速度(m/min)" align="center" prop="liftSpeed" v-if="isShowTableColumns('liftSpeed')"/>-->  
 <el-table-column label="年检类型" align="center" prop="checkType" v-if="isShowTableColumns('checkType')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_check\_type\_crane" :value="scope.row.checkType"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="下次年检时间" align="center" prop="checkTime" width="180" v-if="isShowTableColumns('checkTime')">  
 <template slot-scope="scope">  
 <span>{{ parseTime(scope.row.checkTime, '{y}-{m}-{d}') }}</span>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="维保单位" align="center" prop="maintenanceId" v-if="isShowTableColumns('maintenanceId')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <dict-tag :options="dict.type.equipment\_maintenance\_crane" :value="scope.row.maintenanceId"/>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
 <el-table-column label="状态" align="center" prop="status" v-if="isShowTableColumns('status')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_status" :value="scope.row.status"/>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="备注" align="center" prop="remark" v-if="isShowTableColumns('remark')"/>-->  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['equipment:crane:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['equipment:crane:remove']"  
 >删除</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-search"  
 @click="handleDetails(scope.row)"  
 v-hasPermi="['equipment:crane:edit']"  
 >详细信息</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加、修改和查看起重机信息对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px" :disabled="isDisabled">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="form.deptId" :placeholder="getPlaceholder('请选择分布单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="使用地点" prop="location">  
 <el-input v-model="form.location" :placeholder="getPlaceholder('请输入使用地点', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="注册代码" prop="regcode">  
 <el-input v-model="form.regcode" :placeholder="getPlaceholder('请输入注册代码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备代码" prop="code">  
 <el-input v-model="form.code" :placeholder="getPlaceholder('请输入设备代码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input v-model="form.administrator" :placeholder="getPlaceholder('请输入管理人员', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="联系方式" prop="phone">  
 <el-input v-model="form.phone" :placeholder="getPlaceholder('请输入联系方式', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备编码" prop="number">  
 <el-input v-model="form.number" :placeholder="getPlaceholder('请输入设备编码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备名称" prop="craneName">  
 <el-input v-model="form.craneName" :placeholder="getPlaceholder('请输入设备名称', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="购置日期" prop="buyTime">  
 <el-date-picker clearable  
 v-model="form.buyTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择购置日期', isShowPlaceholder)">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="规格(T)" prop="weight">  
 <el-input v-model="form.weight" :placeholder="getPlaceholder('请输入规格', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="生产厂家" prop="factory">  
 <el-select v-model="form.factory" :placeholder="getPlaceholder('请选择生产厂家', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_maintenance\_crane"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="型号规格" prop="type">  
 <el-input v-model="form.type" :placeholder="getPlaceholder('请输入型号规格', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="起升高度(m)" prop="liftHeight">  
 <el-input v-model="form.liftHeight" :placeholder="getPlaceholder('请输入起升高度', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="跨度(m)" prop="span">  
 <el-input v-model="form.span" :placeholder="getPlaceholder('请输入跨度', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="起升速度(m/min)" prop="liftSpeed">  
 <el-input v-model="form.liftSpeed" :placeholder="getPlaceholder('请输入起升速度', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="年检类型" prop="checkType" v-if="isShowCheckType">  
 <el-select v-model="form.checkType" :placeholder="getPlaceholder('请选择年检类型', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_check\_type\_crane"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="下次年检时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="form.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择下次年检时间', isShowPlaceholder)">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="维保单位" prop="maintenanceId">  
 <el-select v-model="form.maintenanceId" :placeholder="getPlaceholder('请选择维保单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_maintenance\_crane"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="form.status" :placeholder="getPlaceholder('请选择状态', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" type="textarea" :placeholder="getPlaceholder('请输入备注', isShowPlaceholder)"/>  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listCrane, getCrane, delCrane, addCrane, updateCrane } from "@/api/equipment/crane";  
  
export default {  
 name: "Crane",  
 dicts: ['equipment\_status', 'equipment\_check\_type\_crane', 'equipment\_dept\_id', 'equipment\_maintenance\_crane'],  
 data() {  
 return {  
 // 显隐列  
 tableColumns: [  
 { key: 'craneId', label: '序号ID', visible: true },  
 { key: 'deptId', label: '分布单元', visible: true },  
 { key: 'location', label: '使用地点', visible: true },  
 // { key: 'regcode', label: '注册代码', visible: true },  
 // { key: 'code', label: '设备代码', visible: true },  
 { key: 'administrator', label: '管理人员', visible: true },  
 { key: 'phone', label: '联系方式', visible: true },  
 { key: 'number', label: '设备编码', visible: true },  
 { key: 'craneName', label: '设备名称', visible: true },  
 // { key: 'buyTime', label: '购置日期', visible: true },  
 // { key: 'weight', label: '规格', visible: true },  
 // { key: 'factory', label: '生产厂家', visible: true },  
 // { key: 'type', label: '型号规格', visible: true },  
 // { key: 'liftHeight', label: '起升高度', visible: true },  
 // { key: 'span', label: '跨度', visible: true },  
 // { key: 'liftSpeed', label: '起升速度', visible: true },  
 { key: 'checkType', label: '年检类型', visible: true },  
 { key: 'checkTime', label: '下次年检时间', visible: true },  
 // { key: 'maintenanceId', label: '维保单位', visible: true },  
 { key: 'status', label: '状态', visible: true },  
 // { key: 'remark', label: '备注', visible: true },  
 ],  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 起重机管理表格数据  
 craneList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 进行查看设备详细信息时，各组件disabled属性总控  
 isDisabled: false,  
 // 是否展示增改界面的确定与取消键  
 isShowSubmission: true,  
 // 是否展示检验类型  
 isShowCheckType: false,  
 // 是否展示placeholder属性  
 isShowPlaceholder: true,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 deptId: null,  
 location: null,  
 administrator: null,  
 craneName: null,  
 maintenanceId: null,  
 status: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 deptId: [  
 { required: true, message: "分布单位不能为空", trigger: "change" }  
 ],  
 location: [  
 { required: true, message: "使用地点不能为空", trigger: "blur" }  
 ],  
 status: [  
 { required: true, message: "状态不能为空", trigger: "change" }  
 ],  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 其他方法 \*/  
 // 是否展示placeholder  
 getPlaceholder(input, isShowPlaceholder) {  
 // 通过isShowPlaceholder来控制是否展示placeholder  
 return isShowPlaceholder ? (input): '';  
 },  
 // 是否显隐列  
 isShowTableColumns(keyToFind) {  
 const column = this.tableColumns.find(column => column.key === keyToFind);  
 return column.visible;  
 },  
 /\*\* 查询起重机管理列表 \*/  
 getList() {  
 this.loading = true;  
 listCrane(this.queryParams).then(response => {  
 this.craneList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 craneId: null,  
 deptId: null,  
 location: null,  
 regcode: null,  
 code: null,  
 administrator: null,  
 phone: null,  
 number: null,  
 craneName: null,  
 buyTime: null,  
 weight: null,  
 factory: null,  
 type: null,  
 liftHeight: null,  
 span: null,  
 liftSpeed: null,  
 checkType: null,  
 checkTime: null,  
 maintenanceId: null,  
 status: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.craneId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "添加起重机信息";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const craneId = row.craneId || this.ids  
 getCrane(craneId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "修改起重机信息";  
 });  
 },  
 /\*\* 详情按钮操作 \*/  
 handleDetails(row) {  
 this.reset();  
 const craneId = row.craneId || this.ids  
 getCrane(craneId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = true;  
 this.isShowSubmission = false;  
 this.isShowCheckType = true;  
 this.isShowPlaceholder = false;  
 this.title = "查看起重机详细信息";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.craneId != null) {  
 updateCrane(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addCrane(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const craneIds = row.craneId || this.ids;  
 this.$modal.confirm('是否确认删除起重机管理编号为"' + craneIds + '"的数据项？').then(function() {  
 return delCrane(craneIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('equipment/crane/export', {  
 ...this.queryParams  
 }, `crane\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

===================================index.Vue(vessel)==============================

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="queryParams.deptId" placeholder="请选择分布单位" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="设备详细地址" prop="address">  
 <el-input  
 v-model="queryParams.address"  
 placeholder="请输入设备详细地址"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input  
 v-model="queryParams.administrator"  
 placeholder="请输入管理人员"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="设备名称" prop="craneName">  
 <el-input  
 v-model="queryParams.craneName"  
 placeholder="请输入设备名称"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
<!-- <el-form-item label="年检类型" prop="checkType">-->  
<!-- <el-select v-model="queryParams.checkType" placeholder="请选择年检类型" clearable>-->  
<!-- <el-option-->  
<!-- v-for="dict in dict.type.equipment\_check\_type\_vessel"-->  
<!-- :key="dict.value"-->  
<!-- :label="dict.label"-->  
<!-- :value="dict.value"-->  
<!-- />-->  
<!-- </el-select>-->  
<!-- </el-form-item>-->  
<!-- <el-form-item label="维保单位" prop="maintenanceId">-->  
<!-- <el-select v-model="queryParams.maintenanceId" placeholder="请选择维保单位" clearable>-->  
<!-- <el-option-->  
<!-- v-for="dict in dict.type.equipment\_maintenance\_id"-->  
<!-- :key="dict.value"-->  
<!-- :label="dict.label"-->  
<!-- :value="dict.value"-->  
<!-- />-->  
<!-- </el-select>-->  
<!-- </el-form-item>-->  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="queryParams.status" placeholder="请选择状态" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item>  
 <el-button type="primary" icon="el-icon-search" size="mini" @click="handleQuery">搜索</el-button>  
 <el-button icon="el-icon-refresh" size="mini" @click="resetQuery">重置</el-button>  
 </el-form-item>  
 </el-form>  
  
 <el-row :gutter="10" class="mb8">  
 <el-col :span="1.5">  
 <el-button  
 type="primary"  
 plain  
 icon="el-icon-plus"  
 size="mini"  
 @click="handleAdd"  
 v-hasPermi="['equipment:vessel:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['equipment:vessel:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['equipment:vessel:remove']"  
 >删除</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="warning"  
 plain  
 icon="el-icon-download"  
 size="mini"  
 @click="handleExport"  
 v-hasPermi="['equipment:vessel:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" :columns="tableColumns" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="vesselList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="序号ID" align="center" prop="vesselId" v-if="isShowTableColumns('vesselId')"/>  
 <el-table-column label="分布单位" align="center" prop="deptId" v-if="isShowTableColumns('deptId')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_dept\_id" :value="scope.row.deptId"/>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="设备所在地区" align="center" prop="area" v-if="isShowTableColumns('area')"/>-->  
 <el-table-column label="设备详细地址" align="center" prop="address" v-if="isShowTableColumns('address')"/>  
 <el-table-column label="设备代码" align="center" prop="code" v-if="isShowTableColumns('code')"/>  
 <el-table-column label="管理人员" align="center" prop="administrator" v-if="isShowTableColumns('administrator')"/>  
 <el-table-column label="联系方式" align="center" prop="phone" v-if="isShowTableColumns('phone')"/>  
<!-- <el-table-column label="出厂编码" align="center" prop="factoryNumber" v-if="isShowTableColumns('factoryNumber')"/>-->  
<!-- <el-table-column label="注册编码" align="center" prop="regcode" v-if="isShowTableColumns('regcode')"/>-->  
<!-- <el-table-column label="注册状态" align="center" prop="regStatus" v-if="isShowTableColumns('regStatus')"/>-->  
 <el-table-column label="设备名称" align="center" prop="craneName" v-if="isShowTableColumns('craneName')"/>  
<!-- <el-table-column label="购置日期" align="center" prop="buyTime" width="180" v-if="isShowTableColumns('buyTime')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <span>{{ parseTime(scope.row.buyTime, '{y}-{m}-{d}') }}</span>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
<!-- <el-table-column label="使用证编号" align="center" prop="license" v-if="isShowTableColumns('license')"/>-->  
<!-- <el-table-column label="设备类型" align="center" prop="type" v-if="isShowTableColumns('type')"/>-->  
<!-- <el-table-column label="设备型号" align="center" prop="typeNumber" v-if="isShowTableColumns('typeNumber')"/>-->  
<!-- <el-table-column label="设备属性" align="center" prop="attribute" v-if="isShowTableColumns('attribute')"/>-->  
 <el-table-column label="年检类型" align="center" prop="checkType" v-if="isShowTableColumns('checkType')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_check\_type\_vessel" :value="scope.row.checkType"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="下次年检时间" align="center" prop="checkTime" width="180" v-if="isShowTableColumns('checkTime')">  
 <template slot-scope="scope">  
 <span>{{ parseTime(scope.row.checkTime, '{y}-{m}-{d}') }}</span>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="维保单位" align="center" prop="maintenanceId" v-if="isShowTableColumns('maintenanceId')">-->  
<!-- <template slot-scope="scope">-->  
<!-- <dict-tag :options="dict.type.equipment\_maintenance\_id" :value="scope.row.maintenanceId"/>-->  
<!-- </template>-->  
<!-- </el-table-column>-->  
 <el-table-column label="状态" align="center" prop="status" v-if="isShowTableColumns('status')">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.equipment\_status" :value="scope.row.status"/>  
 </template>  
 </el-table-column>  
<!-- <el-table-column label="备注" align="center" prop="remark" v-if="isShowTableColumns('remark')"/>-->  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['equipment:vessel:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['equipment:vessel:remove']"  
 >删除</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-search"  
 @click="handleDetails(scope.row)"  
 v-hasPermi="['equipment:vessel:edit']"  
 >详细信息</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
  
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加、修改和查看压力容器信息对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px" :disabled="isDisabled">  
 <el-form-item label="分布单位" prop="deptId">  
 <el-select v-model="form.deptId" :placeholder="getPlaceholder('请选择分布单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_dept\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="设备所在地区" prop="area">  
 <el-input v-model="form.area" :placeholder="getPlaceholder('请输入设备所在地区', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备详细地址" prop="address">  
 <el-input v-model="form.address" :placeholder="getPlaceholder('请输入设备详细地址', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备代码" prop="code">  
 <el-input v-model="form.code" :placeholder="getPlaceholder('请输入设备代码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="管理人员" prop="administrator">  
 <el-input v-model="form.administrator" :placeholder="getPlaceholder('请输入管理人员', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="联系方式" prop="phone">  
 <el-input v-model="form.phone" :placeholder="getPlaceholder('请输入联系方式', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="出厂编码" prop="factoryNumber">  
 <el-input v-model="form.factoryNumber" :placeholder="getPlaceholder('请输入出厂编码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="注册编码" prop="regcode">  
 <el-input v-model="form.regcode" :placeholder="getPlaceholder('请输入注册编码', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="注册状态" prop="regStatus">  
 <el-input v-model="form.regStatus" :placeholder="getPlaceholder('请输入注册状态', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备名称" prop="craneName">  
 <el-input v-model="form.craneName" :placeholder="getPlaceholder('请输入设备名称', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="购置日期" prop="buyTime">  
 <el-date-picker clearable  
 v-model="form.buyTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择购置日期', isShowPlaceholder)">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="使用证编号" prop="license">  
 <el-input v-model="form.license" :placeholder="getPlaceholder('请输入使用证编号', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备类型" prop="type">  
 <el-input v-model="form.type" :placeholder="getPlaceholder('请输入设备类型', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备型号" prop="typeNumber">  
 <el-input v-model="form.typeNumber" :placeholder="getPlaceholder('请输入设备型号', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="设备属性" prop="attribute">  
 <el-input v-model="form.attribute" :placeholder="getPlaceholder('请输入设备属性', isShowPlaceholder)"/>  
 </el-form-item>  
 <el-form-item label="年检类型" prop="checkType" v-if="isShowCheckType" clearable>  
 <el-select v-model="form.checkType" :placeholder="getPlaceholder('请选择年检类型', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_check\_type\_vessel"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="下次年检时间" prop="checkTime">  
 <el-date-picker clearable  
 v-model="form.checkTime"  
 type="date"  
 value-format="yyyy-MM-dd"  
 :placeholder="getPlaceholder('请选择下次年检时间', isShowPlaceholder)">  
 </el-date-picker>  
 </el-form-item>  
 <el-form-item label="维保单位" prop="maintenanceId">  
 <el-select v-model="form.maintenanceId" :placeholder="getPlaceholder('请选择维保单位', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_maintenance\_id"  
 :key="dict.value"  
 :label="dict.label"  
 :value="parseInt(dict.value)"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="状态" prop="status">  
 <el-select v-model="form.status" :placeholder="getPlaceholder('请选择状态', isShowPlaceholder)" clearable>  
 <el-option  
 v-for="dict in dict.type.equipment\_status"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 ></el-option>  
 </el-select>  
 </el-form-item>  
 <el-form-item label="备注" prop="remark">  
 <el-input v-model="form.remark" type="textarea" :placeholder="getPlaceholder('请输入内容', isShowPlaceholder)"/>  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer" v-if="isShowSubmission">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listVessel, getVessel, delVessel, addVessel, updateVessel } from "@/api/equipment/vessel";  
  
export default {  
 name: "Vessel",  
 dicts: ['equipment\_check\_type\_vessel', 'equipment\_maintenance\_id', 'equipment\_dept\_id', 'equipment\_status'],  
 data() {  
 return {  
 // 显隐列  
 tableColumns: [  
 { key: 'vesselId', label: '序号ID', visible: true },  
 { key: 'deptId', label: '分布单元', visible: true },  
 // { key: 'area', label: '设备所在地区', visible: true },  
 { key: 'address', label: '设备详细地址', visible: true },  
 { key: 'code', label: '设备代码', visible: true },  
 { key: 'administrator', label: '管理人员', visible: true },  
 { key: 'phone', label: '联系方式', visible: true },  
 // { key: 'factoryNumber', label: '出厂编号', visible: true },  
 // { key: 'regcode', label: '注册代码', visible: true },  
 // { key: 'regStatus', label: '注册状态', visible: true },  
 { key: 'craneName', label: '设备名称', visible: true },  
 // { key: 'buyTime', label: '购置日期', visible: true },  
 // { key: 'license', label: '使用证编号', visible: true },  
 // { key: 'type', label: '设备类型', visible: true },  
 // { key: 'typeNumber', label: '设备型号', visible: true },  
 // { key: 'attribute', label: '设备属性', visible: true },  
 { key: 'checkType', label: '年检类型', visible: true },  
 { key: 'checkTime', label: '下次年检时间', visible: true },  
 // { key: 'maintenanceId', label: '维保单位', visible: true },  
 { key: 'status', label: '状态', visible: true },  
 // { key: 'remark', label: '备注', visible: true },  
 ],  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 压力容器管理表格数据  
 vesselList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 进行查看设备详细信息时，各组件disabled属性总控  
 isDisabled: false,  
 // 是否展示增改界面的确定与取消键  
 isShowSubmission: true,  
 // 是否展示检验类型  
 isShowCheckType: false,  
 // 是否展示placeholder属性  
 isShowPlaceholder: true,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 deptId: null,  
 address: null,  
 code: null,  
 administrator: null,  
 craneName: null,  
 checkType: null,  
 maintenanceId: null,  
 status: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 area: [  
 { required: true, message: "设备所在地区不能为空", trigger: "blur" }  
 ],  
 address: [  
 { required: true, message: "设备详细地址不能为空", trigger: "blur" }  
 ],  
 status: [  
 { required: true, message: "状态不能为空", trigger: "change" }  
 ],  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 其他方法 \*/  
 // 是否展示placeholder  
 getPlaceholder(input, isShowPlaceholder) {  
 // 通过isShowPlaceholder来控制是否展示placeholder  
 return isShowPlaceholder ? (input): '';  
 },  
 // 是否显隐列  
 isShowTableColumns(keyToFind) {  
 const column = this.tableColumns.find(column => column.key === keyToFind);  
 return column.visible;  
 },  
 /\*\* 查询压力容器管理列表 \*/  
 getList() {  
 this.loading = true;  
 listVessel(this.queryParams).then(response => {  
 this.vesselList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 vesselId: null,  
 deptId: null,  
 area: null,  
 address: null,  
 code: null,  
 administrator: null,  
 phone: null,  
 factoryNumber: null,  
 regcode: null,  
 regStatus: null,  
 craneName: null,  
 buyTime: null,  
 license: null,  
 type: null,  
 typeNumber: null,  
 attribute: null,  
 checkType: null,  
 checkTime: null,  
 maintenanceId: null,  
 status: null,  
 createBy: null,  
 createTime: null,  
 updateBy: null,  
 updateTime: null,  
 remark: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.vesselId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "添加压力容器信息";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const vesselId = row.vesselId || this.ids  
 getVessel(vesselId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = false;  
 this.isShowSubmission = true;  
 this.isShowCheckType = false;  
 this.isShowPlaceholder = true;  
 this.title = "修改压力容器信息";  
 });  
 },  
 /\*\* 详情按钮操作 \*/  
 handleDetails(row) {  
 this.reset();  
 const vesselId = row.vesselId || this.ids  
 getVessel(vesselId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.isDisabled = true;  
 this.isShowSubmission = false;  
 this.isShowCheckType = true;  
 this.isShowPlaceholder = false;  
 this.title = "查看压力容器信息";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.vesselId != null) {  
 updateVessel(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addVessel(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const vesselIds = row.vesselId || this.ids;  
 this.$modal.confirm('是否确认删除压力容器管理编号为"' + vesselIds + '"的数据项？').then(function() {  
 return delVessel(vesselIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('equipment/vessel/export', {  
 ...this.queryParams  
 }, `vessel\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

==================================SysUserController.java==========================

package com.ruoyi.web.controller.system;  
  
import java.util.List;  
import java.util.stream.Collectors;  
import javax.servlet.http.HttpServletResponse;  
import org.apache.commons.lang3.ArrayUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.validation.annotation.Validated;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.multipart.MultipartFile;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.domain.entity.SysDept;  
import com.ruoyi.common.core.domain.entity.SysRole;  
import com.ruoyi.common.core.domain.entity.SysUser;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.SecurityUtils;  
import com.ruoyi.common.utils.StringUtils;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.system.service.ISysDeptService;  
import com.ruoyi.system.service.ISysPostService;  
import com.ruoyi.system.service.ISysRoleService;  
import com.ruoyi.system.service.ISysUserService;  
  
@RestController  
@RequestMapping("/system/user")  
public class SysUserController extends BaseController  
{  
 @Autowired  
 private ISysUserService userService;  
  
 @Autowired  
 private ISysRoleService roleService;  
  
 @Autowired  
 private ISysDeptService deptService;  
  
 @Autowired  
 private ISysPostService postService;  
  
 /\*\*  
 \* 获取用户列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysUser user)  
 {  
 startPage();  
 List<SysUser> list = userService.selectUserList(user);  
 return getDataTable(list);  
 }  
  
 @Log(title = "用户管理", businessType = BusinessType.EXPORT)  
 @PreAuthorize("@ss.hasPermi('system:user:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysUser user)  
 {  
 List<SysUser> list = userService.selectUserList(user);  
 ExcelUtil<SysUser> util = new ExcelUtil<SysUser>(SysUser.class);  
 util.exportExcel(response, list, "用户数据");  
 }  
  
 @Log(title = "用户管理", businessType = BusinessType.IMPORT)  
 @PreAuthorize("@ss.hasPermi('system:user:import')")  
 @PostMapping("/importData")  
 public AjaxResult importData(MultipartFile file, boolean updateSupport) throws Exception  
 {  
 ExcelUtil<SysUser> util = new ExcelUtil<SysUser>(SysUser.class);  
 List<SysUser> userList = util.importExcel(file.getInputStream());  
 String operName = getUsername();  
 String message = userService.importUser(userList, updateSupport, operName);  
 return success(message);  
 }  
  
 @PostMapping("/importTemplate")  
 public void importTemplate(HttpServletResponse response)  
 {  
 ExcelUtil<SysUser> util = new ExcelUtil<SysUser>(SysUser.class);  
 util.importTemplateExcel(response, "用户数据");  
 }  
  
 /\*\*  
 \* 根据用户编号获取详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:query')")  
 @GetMapping(value = { "/", "/{userId}" })  
 public AjaxResult getInfo(@PathVariable(value = "userId", required = false) Long userId)  
 {  
 userService.checkUserDataScope(userId);  
 AjaxResult ajax = AjaxResult.success();  
 List<SysRole> roles = roleService.selectRoleAll();  
 ajax.put("roles", SysUser.isAdmin(userId) ? roles : roles.stream().filter(r -> !r.isAdmin()).collect(Collectors.toList()));  
 ajax.put("posts", postService.selectPostAll());  
 if (StringUtils.isNotNull(userId))  
 {  
 SysUser sysUser = userService.selectUserById(userId);  
 ajax.put(AjaxResult.DATA\_TAG, sysUser);  
 ajax.put("postIds", postService.selectPostListByUserId(userId));  
 ajax.put("roleIds", sysUser.getRoles().stream().map(SysRole::getRoleId).collect(Collectors.toList()));  
 }  
 return ajax;  
 }  
  
 /\*\*  
 \* 新增用户  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:add')")  
 @Log(title = "用户管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@Validated @RequestBody SysUser user)  
 {  
 if (!userService.checkUserNameUnique(user))  
 {  
 return error("新增用户'" + user.getUserName() + "'失败，登录账号已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getPhonenumber()) && !userService.checkPhoneUnique(user))  
 {  
 return error("新增用户'" + user.getUserName() + "'失败，手机号码已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getEmail()) && !userService.checkEmailUnique(user))  
 {  
 return error("新增用户'" + user.getUserName() + "'失败，邮箱账号已存在");  
 }  
 user.setCreateBy(getUsername());  
 user.setPassword(SecurityUtils.encryptPassword(user.getPassword()));  
 return toAjax(userService.insertUser(user));  
 }  
  
 /\*\*  
 \* 修改用户  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:edit')")  
 @Log(title = "用户管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@Validated @RequestBody SysUser user)  
 {  
 userService.checkUserAllowed(user);  
 userService.checkUserDataScope(user.getUserId());  
 if (!userService.checkUserNameUnique(user))  
 {  
 return error("修改用户'" + user.getUserName() + "'失败，登录账号已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getPhonenumber()) && !userService.checkPhoneUnique(user))  
 {  
 return error("修改用户'" + user.getUserName() + "'失败，手机号码已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getEmail()) && !userService.checkEmailUnique(user))  
 {  
 return error("修改用户'" + user.getUserName() + "'失败，邮箱账号已存在");  
 }  
 user.setUpdateBy(getUsername());  
 return toAjax(userService.updateUser(user));  
 }  
  
 /\*\*  
 \* 删除用户  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:remove')")  
 @Log(title = "用户管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{userIds}")  
 public AjaxResult remove(@PathVariable Long[] userIds)  
 {  
 if (ArrayUtils.contains(userIds, getUserId()))  
 {  
 return error("当前用户不能删除");  
 }  
 return toAjax(userService.deleteUserByIds(userIds));  
 }  
  
 /\*\*  
 \* 重置密码  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:resetPwd')")  
 @Log(title = "用户管理", businessType = BusinessType.UPDATE)  
 @PutMapping("/resetPwd")  
 public AjaxResult resetPwd(@RequestBody SysUser user)  
 {  
 userService.checkUserAllowed(user);  
 userService.checkUserDataScope(user.getUserId());  
 user.setPassword(SecurityUtils.encryptPassword(user.getPassword()));  
 user.setUpdateBy(getUsername());  
 return toAjax(userService.resetPwd(user));  
 }  
  
 /\*\*  
 \* 状态修改  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:edit')")  
 @Log(title = "用户管理", businessType = BusinessType.UPDATE)  
 @PutMapping("/changeStatus")  
 public AjaxResult changeStatus(@RequestBody SysUser user)  
 {  
 userService.checkUserAllowed(user);  
 userService.checkUserDataScope(user.getUserId());  
 user.setUpdateBy(getUsername());  
 return toAjax(userService.updateUserStatus(user));  
 }  
  
 /\*\*  
 \* 根据用户编号获取授权角色  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:query')")  
 @GetMapping("/authRole/{userId}")  
 public AjaxResult authRole(@PathVariable("userId") Long userId)  
 {  
 AjaxResult ajax = AjaxResult.success();  
 SysUser user = userService.selectUserById(userId);  
 List<SysRole> roles = roleService.selectRolesByUserId(userId);  
 ajax.put("user", user);  
 ajax.put("roles", SysUser.isAdmin(userId) ? roles : roles.stream().filter(r -> !r.isAdmin()).collect(Collectors.toList()));  
 return ajax;  
 }  
  
 /\*\*  
 \* 用户授权角色  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:edit')")  
 @Log(title = "用户管理", businessType = BusinessType.GRANT)  
 @PutMapping("/authRole")  
 public AjaxResult insertAuthRole(Long userId, Long[] roleIds)  
 {  
 userService.checkUserDataScope(userId);  
 userService.insertUserAuth(userId, roleIds);  
 return success();  
 }  
  
 /\*\*  
 \* 获取部门树列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:user:list')")  
 @GetMapping("/deptTree")  
 public AjaxResult deptTree(SysDept dept)  
 {  
 return success(deptService.selectDeptTreeList(dept));  
 }  
}

===================================SysLoginController.java========================

package com.ruoyi.web.controller.system;  
  
import java.util.List;  
import java.util.Set;  
  
import com.ruoyi.common.core.domain.model.LoginBody;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.constant.Constants;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.domain.entity.SysMenu;  
import com.ruoyi.common.core.domain.entity.SysUser;  
import com.ruoyi.common.utils.SecurityUtils;  
import com.ruoyi.framework.web.service.SysLoginService;  
import com.ruoyi.framework.web.service.SysPermissionService;  
import com.ruoyi.system.service.ISysMenuService;  
  
@RestController  
public class SysLoginController  
{  
 @Autowired  
 private SysLoginService loginService;  
  
 @Autowired  
 private ISysMenuService menuService;  
  
 @Autowired  
 private SysPermissionService permissionService;  
  
 /\*\*  
 \* 登录方法  
 \*   
 \* @param loginBody 登录信息  
 \* @return 结果  
 \*/  
 @PostMapping("/login")  
 public AjaxResult login(@RequestBody LoginBody loginBody)  
 {  
 AjaxResult ajax = AjaxResult.success();  
 // 生成令牌  
 String token = loginService.login(loginBody.getUsername(), loginBody.getPassword(), loginBody.getCode(),  
 loginBody.getUuid());  
 ajax.put(Constants.TOKEN, token);  
 return ajax;  
 }  
  
 /\*\*  
 \* 获取用户信息  
 \*   
 \* @return 用户信息  
 \*/  
 @GetMapping("getInfo")  
 public AjaxResult getInfo()  
 {  
 SysUser user = SecurityUtils.getLoginUser().getUser();  
 // 角色集合  
 Set<String> roles = permissionService.getRolePermission(user);  
 // 权限集合  
 Set<String> permissions = permissionService.getMenuPermission(user);  
 AjaxResult ajax = AjaxResult.success();  
 ajax.put("user", user);  
 ajax.put("roles", roles);  
 ajax.put("permissions", permissions);  
 return ajax;  
 }  
  
 /\*\*  
 \* 获取路由信息  
 \*   
 \* @return 路由信息  
 \*/  
 @GetMapping("getRouters")  
 public AjaxResult getRouters()  
 {  
 Long userId = SecurityUtils.getUserId();  
 List<SysMenu> menus = menuService.selectMenuTreeByUserId(userId);  
 return AjaxResult.success(menuService.buildMenus(menus));  
 }  
}

===================================SysRegisterController.java=======================

package com.ruoyi.web.controller.system;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.domain.model.RegisterBody;  
import com.ruoyi.common.utils.StringUtils;  
import com.ruoyi.framework.web.service.SysRegisterService;  
import com.ruoyi.system.service.ISysConfigService;  
  
@RestController  
public class SysRegisterController extends BaseController  
{  
 @Autowired  
 private SysRegisterService registerService;  
  
 @Autowired  
 private ISysConfigService configService;  
  
 @PostMapping("/register")  
 public AjaxResult register(@RequestBody RegisterBody user)  
 {  
 if (!("true".equals(configService.selectConfigByKey("sys.account.registerUser"))))  
 {  
 return error("当前系统没有开启注册功能！");  
 }  
 String msg = registerService.register(user);  
 return StringUtils.isEmpty(msg) ? success() : error(msg);  
 }  
}

================================SysRoleController.java=============================

package com.ruoyi.web.controller.system;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.validation.annotation.Validated;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.domain.entity.SysDept;  
import com.ruoyi.common.core.domain.entity.SysRole;  
import com.ruoyi.common.core.domain.entity.SysUser;  
import com.ruoyi.common.core.domain.model.LoginUser;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.StringUtils;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.framework.web.service.SysPermissionService;  
import com.ruoyi.framework.web.service.TokenService;  
import com.ruoyi.system.domain.SysUserRole;  
import com.ruoyi.system.service.ISysDeptService;  
import com.ruoyi.system.service.ISysRoleService;  
import com.ruoyi.system.service.ISysUserService;  
  
@RestController  
@RequestMapping("/system/role")  
public class SysRoleController extends BaseController  
{  
 @Autowired  
 private ISysRoleService roleService;  
  
 @Autowired  
 private TokenService tokenService;  
  
 @Autowired  
 private SysPermissionService permissionService;  
  
 @Autowired  
 private ISysUserService userService;  
  
 @Autowired  
 private ISysDeptService deptService;  
  
 @PreAuthorize("@ss.hasPermi('system:role:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysRole role)  
 {  
 startPage();  
 List<SysRole> list = roleService.selectRoleList(role);  
 return getDataTable(list);  
 }  
  
 @Log(title = "角色管理", businessType = BusinessType.EXPORT)  
 @PreAuthorize("@ss.hasPermi('system:role:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysRole role)  
 {  
 List<SysRole> list = roleService.selectRoleList(role);  
 ExcelUtil<SysRole> util = new ExcelUtil<SysRole>(SysRole.class);  
 util.exportExcel(response, list, "角色数据");  
 }  
  
 /\*\*  
 \* 根据角色编号获取详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:query')")  
 @GetMapping(value = "/{roleId}")  
 public AjaxResult getInfo(@PathVariable Long roleId)  
 {  
 roleService.checkRoleDataScope(roleId);  
 return success(roleService.selectRoleById(roleId));  
 }  
  
 /\*\*  
 \* 新增角色  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:add')")  
 @Log(title = "角色管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@Validated @RequestBody SysRole role)  
 {  
 if (!roleService.checkRoleNameUnique(role))  
 {  
 return error("新增角色'" + role.getRoleName() + "'失败，角色名称已存在");  
 }  
 else if (!roleService.checkRoleKeyUnique(role))  
 {  
 return error("新增角色'" + role.getRoleName() + "'失败，角色权限已存在");  
 }  
 role.setCreateBy(getUsername());  
 return toAjax(roleService.insertRole(role));  
  
 }  
  
 /\*\*  
 \* 修改保存角色  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@Validated @RequestBody SysRole role)  
 {  
 roleService.checkRoleAllowed(role);  
 roleService.checkRoleDataScope(role.getRoleId());  
 if (!roleService.checkRoleNameUnique(role))  
 {  
 return error("修改角色'" + role.getRoleName() + "'失败，角色名称已存在");  
 }  
 else if (!roleService.checkRoleKeyUnique(role))  
 {  
 return error("修改角色'" + role.getRoleName() + "'失败，角色权限已存在");  
 }  
 role.setUpdateBy(getUsername());  
   
 if (roleService.updateRole(role) > 0)  
 {  
 // 更新缓存用户权限  
 LoginUser loginUser = getLoginUser();  
 if (StringUtils.isNotNull(loginUser.getUser()) && !loginUser.getUser().isAdmin())  
 {  
 loginUser.setPermissions(permissionService.getMenuPermission(loginUser.getUser()));  
 loginUser.setUser(userService.selectUserByUserName(loginUser.getUser().getUserName()));  
 tokenService.setLoginUser(loginUser);  
 }  
 return success();  
 }  
 return error("修改角色'" + role.getRoleName() + "'失败，请联系管理员");  
 }  
  
 /\*\*  
 \* 修改保存数据权限  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.UPDATE)  
 @PutMapping("/dataScope")  
 public AjaxResult dataScope(@RequestBody SysRole role)  
 {  
 roleService.checkRoleAllowed(role);  
 roleService.checkRoleDataScope(role.getRoleId());  
 return toAjax(roleService.authDataScope(role));  
 }  
  
 /\*\*  
 \* 状态修改  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.UPDATE)  
 @PutMapping("/changeStatus")  
 public AjaxResult changeStatus(@RequestBody SysRole role)  
 {  
 roleService.checkRoleAllowed(role);  
 roleService.checkRoleDataScope(role.getRoleId());  
 role.setUpdateBy(getUsername());  
 return toAjax(roleService.updateRoleStatus(role));  
 }  
  
 /\*\*  
 \* 删除角色  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:remove')")  
 @Log(title = "角色管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{roleIds}")  
 public AjaxResult remove(@PathVariable Long[] roleIds)  
 {  
 return toAjax(roleService.deleteRoleByIds(roleIds));  
 }  
  
 /\*\*  
 \* 获取角色选择框列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:query')")  
 @GetMapping("/optionselect")  
 public AjaxResult optionselect()  
 {  
 return success(roleService.selectRoleAll());  
 }  
  
 /\*\*  
 \* 查询已分配用户角色列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:list')")  
 @GetMapping("/authUser/allocatedList")  
 public TableDataInfo allocatedList(SysUser user)  
 {  
 startPage();  
 List<SysUser> list = userService.selectAllocatedList(user);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 查询未分配用户角色列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:list')")  
 @GetMapping("/authUser/unallocatedList")  
 public TableDataInfo unallocatedList(SysUser user)  
 {  
 startPage();  
 List<SysUser> list = userService.selectUnallocatedList(user);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 取消授权用户  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.GRANT)  
 @PutMapping("/authUser/cancel")  
 public AjaxResult cancelAuthUser(@RequestBody SysUserRole userRole)  
 {  
 return toAjax(roleService.deleteAuthUser(userRole));  
 }  
  
 /\*\*  
 \* 批量取消授权用户  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.GRANT)  
 @PutMapping("/authUser/cancelAll")  
 public AjaxResult cancelAuthUserAll(Long roleId, Long[] userIds)  
 {  
 return toAjax(roleService.deleteAuthUsers(roleId, userIds));  
 }  
  
 /\*\*  
 \* 批量选择用户授权  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:edit')")  
 @Log(title = "角色管理", businessType = BusinessType.GRANT)  
 @PutMapping("/authUser/selectAll")  
 public AjaxResult selectAuthUserAll(Long roleId, Long[] userIds)  
 {  
 roleService.checkRoleDataScope(roleId);  
 return toAjax(roleService.insertAuthUsers(roleId, userIds));  
 }  
  
 /\*\*  
 \* 获取对应角色部门树列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:role:query')")  
 @GetMapping(value = "/deptTree/{roleId}")  
 public AjaxResult deptTree(@PathVariable("roleId") Long roleId)  
 {  
 AjaxResult ajax = AjaxResult.success();  
 ajax.put("checkedKeys", deptService.selectDeptListByRoleId(roleId));  
 ajax.put("depts", deptService.selectDeptTreeList(new SysDept()));  
 return ajax;  
 }  
}

===============================SysDeptController.java==============================

package com.ruoyi.web.controller.system;  
  
import java.util.List;  
import org.apache.commons.lang3.ArrayUtils;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.validation.annotation.Validated;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.constant.UserConstants;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.domain.entity.SysDept;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.StringUtils;  
import com.ruoyi.system.service.ISysDeptService;  
  
@RestController  
@RequestMapping("/system/dept")  
public class SysDeptController extends BaseController  
{  
 @Autowired  
 private ISysDeptService deptService;  
  
 /\*\*  
 \* 获取部门列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:list')")  
 @GetMapping("/list")  
 public AjaxResult list(SysDept dept)  
 {  
 List<SysDept> depts = deptService.selectDeptList(dept);  
 return success(depts);  
 }  
  
 /\*\*  
 \* 查询部门列表（排除节点）  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:list')")  
 @GetMapping("/list/exclude/{deptId}")  
 public AjaxResult excludeChild(@PathVariable(value = "deptId", required = false) Long deptId)  
 {  
 List<SysDept> depts = deptService.selectDeptList(new SysDept());  
 depts.removeIf(d -> d.getDeptId().intValue() == deptId || ArrayUtils.contains(StringUtils.split(d.getAncestors(), ","), deptId + ""));  
 return success(depts);  
 }  
  
 /\*\*  
 \* 根据部门编号获取详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:query')")  
 @GetMapping(value = "/{deptId}")  
 public AjaxResult getInfo(@PathVariable Long deptId)  
 {  
 deptService.checkDeptDataScope(deptId);  
 return success(deptService.selectDeptById(deptId));  
 }  
  
 /\*\*  
 \* 新增部门  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:add')")  
 @Log(title = "部门管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@Validated @RequestBody SysDept dept)  
 {  
 if (!deptService.checkDeptNameUnique(dept))  
 {  
 return error("新增部门'" + dept.getDeptName() + "'失败，部门名称已存在");  
 }  
 dept.setCreateBy(getUsername());  
 return toAjax(deptService.insertDept(dept));  
 }  
  
 /\*\*  
 \* 修改部门  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:edit')")  
 @Log(title = "部门管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@Validated @RequestBody SysDept dept)  
 {  
 Long deptId = dept.getDeptId();  
 deptService.checkDeptDataScope(deptId);  
 if (!deptService.checkDeptNameUnique(dept))  
 {  
 return error("修改部门'" + dept.getDeptName() + "'失败，部门名称已存在");  
 }  
 else if (dept.getParentId().equals(deptId))  
 {  
 return error("修改部门'" + dept.getDeptName() + "'失败，上级部门不能是自己");  
 }  
 else if (StringUtils.equals(UserConstants.DEPT\_DISABLE, dept.getStatus()) && deptService.selectNormalChildrenDeptById(deptId) > 0)  
 {  
 return error("该部门包含未停用的子部门！");  
 }  
 dept.setUpdateBy(getUsername());  
 return toAjax(deptService.updateDept(dept));  
 }  
  
 /\*\*  
 \* 删除部门  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:dept:remove')")  
 @Log(title = "部门管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{deptId}")  
 public AjaxResult remove(@PathVariable Long deptId)  
 {  
 if (deptService.hasChildByDeptId(deptId))  
 {  
 return warn("存在下级部门,不允许删除");  
 }  
 if (deptService.checkDeptExistUser(deptId))  
 {  
 return warn("部门存在用户,不允许删除");  
 }  
 deptService.checkDeptDataScope(deptId);  
 return toAjax(deptService.deleteDeptById(deptId));  
 }  
}

==================================SysPostController.java===========================package com.ruoyi.web.controller.system;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.validation.annotation.Validated;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.system.domain.SysPost;  
import com.ruoyi.system.service.ISysPostService;  
  
@RestController  
@RequestMapping("/system/post")  
public class SysPostController extends BaseController  
{  
 @Autowired  
 private ISysPostService postService;  
  
 /\*\*  
 \* 获取岗位列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:post:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysPost post)  
 {  
 startPage();  
 List<SysPost> list = postService.selectPostList(post);  
 return getDataTable(list);  
 }  
   
 @Log(title = "岗位管理", businessType = BusinessType.EXPORT)  
 @PreAuthorize("@ss.hasPermi('system:post:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysPost post)  
 {  
 List<SysPost> list = postService.selectPostList(post);  
 ExcelUtil<SysPost> util = new ExcelUtil<SysPost>(SysPost.class);  
 util.exportExcel(response, list, "岗位数据");  
 }  
  
 /\*\*  
 \* 根据岗位编号获取详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:post:query')")  
 @GetMapping(value = "/{postId}")  
 public AjaxResult getInfo(@PathVariable Long postId)  
 {  
 return success(postService.selectPostById(postId));  
 }  
  
 /\*\*  
 \* 新增岗位  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:post:add')")  
 @Log(title = "岗位管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@Validated @RequestBody SysPost post)  
 {  
 if (!postService.checkPostNameUnique(post))  
 {  
 return error("新增岗位'" + post.getPostName() + "'失败，岗位名称已存在");  
 }  
 else if (!postService.checkPostCodeUnique(post))  
 {  
 return error("新增岗位'" + post.getPostName() + "'失败，岗位编码已存在");  
 }  
 post.setCreateBy(getUsername());  
 return toAjax(postService.insertPost(post));  
 }  
  
 /\*\*  
 \* 修改岗位  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:post:edit')")  
 @Log(title = "岗位管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@Validated @RequestBody SysPost post)  
 {  
 if (!postService.checkPostNameUnique(post))  
 {  
 return error("修改岗位'" + post.getPostName() + "'失败，岗位名称已存在");  
 }  
 else if (!postService.checkPostCodeUnique(post))  
 {  
 return error("修改岗位'" + post.getPostName() + "'失败，岗位编码已存在");  
 }  
 post.setUpdateBy(getUsername());  
 return toAjax(postService.updatePost(post));  
 }  
  
 /\*\*  
 \* 删除岗位  
 \*/  
 @PreAuthorize("@ss.hasPermi('system:post:remove')")  
 @Log(title = "岗位管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{postIds}")  
 public AjaxResult remove(@PathVariable Long[] postIds)  
 {  
 return toAjax(postService.deletePostByIds(postIds));  
 }  
  
 /\*\*  
 \* 获取岗位选择框列表  
 \*/  
 @GetMapping("/optionselect")  
 public AjaxResult optionselect()  
 {  
 List<SysPost> posts = postService.selectPostAll();  
 return success(posts);  
 }  
}

==============================EquipmentLiftController.java==========================package com.ruoyi.web.controller.equipment;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.equipment.domain.EquipmentLift;  
import com.ruoyi.equipment.service.IEquipmentLiftService;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.common.core.page.TableDataInfo;  
  
/\*\*  
 \* 电梯管理Controller  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
@RestController  
@RequestMapping("/equipment/lift")  
public class EquipmentLiftController extends BaseController  
{  
 @Autowired  
 private IEquipmentLiftService equipmentLiftService;  
  
 /\*\*  
 \* 查询电梯管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(EquipmentLift equipmentLift)  
 {  
 startPage();  
 List<EquipmentLift> list = equipmentLiftService.selectEquipmentLiftList(equipmentLift);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出电梯管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:export')")  
 @Log(title = "电梯管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, EquipmentLift equipmentLift)  
 {  
 List<EquipmentLift> list = equipmentLiftService.selectEquipmentLiftList(equipmentLift);  
 ExcelUtil<EquipmentLift> util = new ExcelUtil<EquipmentLift>(EquipmentLift.class);  
 util.exportExcel(response, list, "电梯管理数据");  
 }  
  
 /\*\*  
 \* 获取电梯管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:query')")  
 @GetMapping(value = "/{liftId}")  
 public AjaxResult getInfo(@PathVariable("liftId") Long liftId)  
 {  
 return success(equipmentLiftService.selectEquipmentLiftByLiftId(liftId));  
 }  
  
 /\*\*  
 \* 新增电梯管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:add')")  
 @Log(title = "电梯管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody EquipmentLift equipmentLift)  
 {  
 return toAjax(equipmentLiftService.insertEquipmentLift(equipmentLift));  
 }  
  
 /\*\*  
 \* 修改电梯管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:edit')")  
 @Log(title = "电梯管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody EquipmentLift equipmentLift)  
 {  
 return toAjax(equipmentLiftService.updateEquipmentLift(equipmentLift));  
 }  
  
 /\*\*  
 \* 删除电梯管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:lift:remove')")  
 @Log(title = "电梯管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{liftIds}")  
 public AjaxResult remove(@PathVariable Long[] liftIds)  
 {  
 return toAjax(equipmentLiftService.deleteEquipmentLiftByLiftIds(liftIds));  
 }  
}

==============================EquipmentCraneController.java========================package com.ruoyi.web.controller.equipment;  
  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.equipment.domain.EquipmentCrane;  
import com.ruoyi.equipment.service.IEquipmentCraneService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.\*;  
  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
/\*\*  
 \* 起重机管理Controller  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
@RestController  
@RequestMapping("/equipment/crane")  
public class EquipmentCraneController extends BaseController  
{  
 @Autowired  
 private IEquipmentCraneService equipmentCraneService;  
  
 /\*\*  
 \* 查询起重机管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(EquipmentCrane equipmentCrane)  
 {  
 startPage();  
 List<EquipmentCrane> list = equipmentCraneService.selectEquipmentCraneList(equipmentCrane);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出起重机管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:export')")  
 @Log(title = "起重机管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, EquipmentCrane equipmentCrane)  
 {  
 List<EquipmentCrane> list = equipmentCraneService.selectEquipmentCraneList(equipmentCrane);  
 ExcelUtil<EquipmentCrane> util = new ExcelUtil<EquipmentCrane>(EquipmentCrane.class);  
 util.exportExcel(response, list, "起重机管理数据");  
 }  
  
 /\*\*  
 \* 获取起重机管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:query')")  
 @GetMapping(value = "/{craneId}")  
 public AjaxResult getInfo(@PathVariable("craneId") Long craneId)  
 {  
 return success(equipmentCraneService.selectEquipmentCraneByCraneId(craneId));  
 }  
  
 /\*\*  
 \* 新增起重机管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:add')")  
 @Log(title = "起重机管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody EquipmentCrane equipmentCrane)  
 {  
 return toAjax(equipmentCraneService.insertEquipmentCrane(equipmentCrane));  
 }  
  
 /\*\*  
 \* 修改起重机管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:edit')")  
 @Log(title = "起重机管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody EquipmentCrane equipmentCrane)  
 {  
 return toAjax(equipmentCraneService.updateEquipmentCrane(equipmentCrane));  
 }  
  
 /\*\*  
 \* 删除起重机管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:crane:remove')")  
 @Log(title = "起重机管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{craneIds}")  
 public AjaxResult remove(@PathVariable Long[] craneIds)  
 {  
 return toAjax(equipmentCraneService.deleteEquipmentCraneByCraneIds(craneIds));  
 }  
}

=============================EquipmentVesselController.java========================package com.ruoyi.web.controller.equipment;  
  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.equipment.domain.EquipmentVessel;  
import com.ruoyi.equipment.service.IEquipmentVesselService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.\*;  
  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
/\*\*  
 \* 压力容器管理Controller  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
@RestController  
@RequestMapping("/equipment/vessel")  
public class EquipmentVesselController extends BaseController  
{  
 @Autowired  
 private IEquipmentVesselService equipmentVesselService;  
  
 /\*\*  
 \* 查询压力容器管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(EquipmentVessel equipmentVessel)  
 {  
 startPage();  
 List<EquipmentVessel> list = equipmentVesselService.selectEquipmentVesselList(equipmentVessel);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出压力容器管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:export')")  
 @Log(title = "压力容器管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, EquipmentVessel equipmentVessel)  
 {  
 List<EquipmentVessel> list = equipmentVesselService.selectEquipmentVesselList(equipmentVessel);  
 ExcelUtil<EquipmentVessel> util = new ExcelUtil<EquipmentVessel>(EquipmentVessel.class);  
 util.exportExcel(response, list, "压力容器管理数据");  
 }  
  
 /\*\*  
 \* 获取压力容器管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:query')")  
 @GetMapping(value = "/{vesselId}")  
 public AjaxResult getInfo(@PathVariable("vesselId") Long vesselId)  
 {  
 return success(equipmentVesselService.selectEquipmentVesselByVesselId(vesselId));  
 }  
  
 /\*\*  
 \* 新增压力容器管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:add')")  
 @Log(title = "压力容器管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody EquipmentVessel equipmentVessel)  
 {  
 return toAjax(equipmentVesselService.insertEquipmentVessel(equipmentVessel));  
 }  
  
 /\*\*  
 \* 修改压力容器管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:edit')")  
 @Log(title = "压力容器管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody EquipmentVessel equipmentVessel)  
 {  
 return toAjax(equipmentVesselService.updateEquipmentVessel(equipmentVessel));  
 }  
  
 /\*\*  
 \* 删除压力容器管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:vessel:remove')")  
 @Log(title = "压力容器管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{vesselIds}")  
 public AjaxResult remove(@PathVariable Long[] vesselIds)  
 {  
 return toAjax(equipmentVesselService.deleteEquipmentVesselByVesselIds(vesselIds));  
 }  
}

=============================EquipmentCheckController.java========================package com.ruoyi.web.controller.equipment;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.PostMapping;  
import org.springframework.web.bind.annotation.PutMapping;  
import org.springframework.web.bind.annotation.DeleteMapping;  
import org.springframework.web.bind.annotation.PathVariable;  
import org.springframework.web.bind.annotation.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.equipment.domain.EquipmentCheck;  
import com.ruoyi.equipment.service.IEquipmentCheckService;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.common.core.page.TableDataInfo;  
  
/\*\*  
 \* 检验类型管理Controller  
 \*   
 \* @author cqq  
 \* @date 2023-09-12  
 \*/  
@RestController  
@RequestMapping("/equipment/check")  
public class EquipmentCheckController extends BaseController  
{  
 @Autowired  
 private IEquipmentCheckService equipmentCheckService;  
  
 /\*\*  
 \* 查询检验类型管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(EquipmentCheck equipmentCheck)  
 {  
 startPage();  
 List<EquipmentCheck> list = equipmentCheckService.selectEquipmentCheckList(equipmentCheck);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出检验类型管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:export')")  
 @Log(title = "检验类型管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, EquipmentCheck equipmentCheck)  
 {  
 List<EquipmentCheck> list = equipmentCheckService.selectEquipmentCheckList(equipmentCheck);  
 ExcelUtil<EquipmentCheck> util = new ExcelUtil<EquipmentCheck>(EquipmentCheck.class);  
 util.exportExcel(response, list, "检验类型管理数据");  
 }  
  
 /\*\*  
 \* 获取检验类型管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:query')")  
 @GetMapping(value = "/{checkId}")  
 public AjaxResult getInfo(@PathVariable("checkId") Integer checkId)  
 {  
 return success(equipmentCheckService.selectEquipmentCheckByCheckId(checkId));  
 }  
  
 /\*\*  
 \* 新增检验类型管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:add')")  
 @Log(title = "检验类型管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody EquipmentCheck equipmentCheck)  
 {  
 return toAjax(equipmentCheckService.insertEquipmentCheck(equipmentCheck));  
 }  
  
 /\*\*  
 \* 修改检验类型管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:edit')")  
 @Log(title = "检验类型管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody EquipmentCheck equipmentCheck)  
 {  
 return toAjax(equipmentCheckService.updateEquipmentCheck(equipmentCheck));  
 }  
  
 /\*\*  
 \* 删除检验类型管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('equipment:check:remove')")  
 @Log(title = "检验类型管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{checkIds}")  
 public AjaxResult remove(@PathVariable Integer[] checkIds)  
 {  
 return toAjax(equipmentCheckService.deleteEquipmentCheckByCheckIds(checkIds));  
 }  
}

=================================CostController.java===============================

package com.ruoyi.web.controller.image;  
  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.SecurityUtils;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.image.domain.Cost;  
import com.ruoyi.image.service.ICostService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.\*;  
  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
/\*\*  
 \* 报销凭证管理Controller  
 \*   
 \* @author dzh  
 \* @date 2024-04-14  
 \*/  
@RestController  
@RequestMapping("/image/cost")  
public class CostController extends BaseController  
{  
 @Autowired  
 private ICostService costService;  
  
 /\*\*  
 \* 查询报销凭证管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(Cost cost)  
 {  
 startPage();  
 List<Cost> list = costService.selectCostList(cost);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出报销凭证管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:export')")  
 @Log(title = "报销凭证管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, Cost cost)  
 {  
 List<Cost> list = costService.selectCostList(cost);  
 ExcelUtil<Cost> util = new ExcelUtil<Cost>(Cost.class);  
 util.exportExcel(response, list, "报销凭证管理数据");  
 }  
  
 /\*\*  
 \* 获取报销凭证管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:query')")  
 @GetMapping(value = "/{costId}")  
 public AjaxResult getInfo(@PathVariable("costId") Long costId)  
 {  
 return success(costService.selectCostByCostId(costId));  
 }  
  
 /\*\*  
 \* 新增报销凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:add')")  
 @Log(title = "报销凭证管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody Cost cost)  
 {  
 cost.setUserId(getUserId());  
 String createAccount = SecurityUtils.getUsername();  
 cost.setUser(createAccount);  
 return toAjax(costService.insertCost(cost));  
 }  
  
 /\*\*  
 \* 修改报销凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:edit')")  
 @Log(title = "报销凭证管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody Cost cost)  
 {  
 return toAjax(costService.updateCost(cost));  
 }  
  
 /\*\*  
 \* 删除报销凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:cost:remove')")  
 @Log(title = "报销凭证管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{costIds}")  
 public AjaxResult remove(@PathVariable Long[] costIds)  
 {  
 return toAjax(costService.deleteCostByCostIds(costIds));  
 }  
}

================================ImageController.java==============================

package com.ruoyi.web.controller.image;  
  
import com.ruoyi.common.annotation.Log;  
import com.ruoyi.common.core.controller.BaseController;  
import com.ruoyi.common.core.domain.AjaxResult;  
import com.ruoyi.common.core.page.TableDataInfo;  
import com.ruoyi.common.enums.BusinessType;  
import com.ruoyi.common.utils.SecurityUtils;  
import com.ruoyi.common.utils.poi.ExcelUtil;  
import com.ruoyi.image.domain.Image;  
import com.ruoyi.image.service.IImageService;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.\*;  
  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
/\*\*  
 \* 维修凭证管理Controller  
 \*   
 \* @author dzh  
 \* @date 2024-04-16  
 \*/  
@RestController  
@RequestMapping("/image/image")  
public class ImageController extends BaseController  
{  
 @Autowired  
 private IImageService imageService;  
  
 /\*\*  
 \* 查询维修凭证管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(Image image)  
 {  
 startPage();  
 List<Image> list = imageService.selectImageList(image);  
 return getDataTable(list);  
 }  
  
 /\*\*  
 \* 导出维修凭证管理列表  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:export')")  
 @Log(title = "维修凭证管理", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, Image image)  
 {  
 List<Image> list = imageService.selectImageList(image);  
 ExcelUtil<Image> util = new ExcelUtil<Image>(Image.class);  
 util.exportExcel(response, list, "维修凭证管理数据");  
 }  
  
 /\*\*  
 \* 获取维修凭证管理详细信息  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:query')")  
 @GetMapping(value = "/{logId}")  
 public AjaxResult getInfo(@PathVariable("logId") Long logId)  
 {  
 return success(imageService.selectImageByLogId(logId));  
 }  
  
 /\*\*  
 \* 新增维修凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:add')")  
 @Log(title = "维修凭证管理", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody Image image)  
 {  
 image.setUserId(getUserId());  
 String createAccount = SecurityUtils.getUsername();  
 image.setUser(createAccount);  
// image.setUser(image.getUser());  
 return toAjax(imageService.insertImage(image));  
 }  
  
 /\*\*  
 \* 修改维修凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:edit')")  
 @Log(title = "维修凭证管理", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody Image image)  
 {  
 return toAjax(imageService.updateImage(image));  
 }  
  
 /\*\*  
 \* 删除维修凭证管理  
 \*/  
 @PreAuthorize("@ss.hasPermi('image:image:remove')")  
 @Log(title = "维修凭证管理", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{logIds}")  
 public AjaxResult remove(@PathVariable Long[] logIds)  
 {  
 return toAjax(imageService.deleteImageByLogIds(logIds));  
 }  
}

============================IEquipmentLiftService.java=============================package com.ruoyi.equipment.service;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentLift;  
  
/\*\*  
 \* 电梯管理Service接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface IEquipmentLiftService   
{  
 /\*\*  
 \* 查询电梯管理  
 \*   
 \* @param liftId 电梯管理主键  
 \* @return 电梯管理  
 \*/  
 public EquipmentLift selectEquipmentLiftByLiftId(Long liftId);  
  
 /\*\*  
 \* 查询电梯管理列表  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 电梯管理集合  
 \*/  
 public List<EquipmentLift> selectEquipmentLiftList(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 新增电梯管理  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentLift(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 修改电梯管理  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentLift(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 批量删除电梯管理  
 \*   
 \* @param liftIds 需要删除的电梯管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentLiftByLiftIds(Long[] liftIds);  
  
 /\*\*  
 \* 删除电梯管理信息  
 \*   
 \* @param liftId 电梯管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentLiftByLiftId(Long liftId);  
}

==========================IEquipmentCheckService.java=============================package com.ruoyi.equipment.service;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentCheck;  
  
/\*\*  
 \* 检验类型管理Service接口  
 \*   
 \* @author cqq  
 \* @date 2023-09-12  
 \*/  
public interface IEquipmentCheckService   
{  
 /\*\*  
 \* 查询检验类型管理  
 \*   
 \* @param checkId 检验类型管理主键  
 \* @return 检验类型管理  
 \*/  
 public EquipmentCheck selectEquipmentCheckByCheckId(Integer checkId);  
  
 /\*\*  
 \* 查询检验类型管理列表  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 检验类型管理集合  
 \*/  
 public List<EquipmentCheck> selectEquipmentCheckList(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 新增检验类型管理  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentCheck(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 修改检验类型管理  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentCheck(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 批量删除检验类型管理  
 \*   
 \* @param checkIds 需要删除的检验类型管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCheckByCheckIds(Integer[] checkIds);  
  
 /\*\*  
 \* 删除检验类型管理信息  
 \*   
 \* @param checkId 检验类型管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCheckByCheckId(Integer checkId);  
}

==========================IEquipmentCraneService.java=============================package com.ruoyi.equipment.service;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentCrane;  
  
/\*\*  
 \* 起重机管理Service接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface IEquipmentCraneService   
{  
 /\*\*  
 \* 查询起重机管理  
 \*   
 \* @param craneId 起重机管理主键  
 \* @return 起重机管理  
 \*/  
 public EquipmentCrane selectEquipmentCraneByCraneId(Long craneId);  
  
 /\*\*  
 \* 查询起重机管理列表  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 起重机管理集合  
 \*/  
 public List<EquipmentCrane> selectEquipmentCraneList(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 新增起重机管理  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentCrane(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 修改起重机管理  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentCrane(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 批量删除起重机管理  
 \*   
 \* @param craneIds 需要删除的起重机管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCraneByCraneIds(Long[] craneIds);  
  
 /\*\*  
 \* 删除起重机管理信息  
 \*   
 \* @param craneId 起重机管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCraneByCraneId(Long craneId);  
}

==========================IEquipmentVesselService.java=============================package com.ruoyi.equipment.service;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentVessel;  
  
/\*\*  
 \* 压力容器管理Service接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface IEquipmentVesselService   
{  
 /\*\*  
 \* 查询压力容器管理  
 \*   
 \* @param vesselId 压力容器管理主键  
 \* @return 压力容器管理  
 \*/  
 public EquipmentVessel selectEquipmentVesselByVesselId(Long vesselId);  
  
 /\*\*  
 \* 查询压力容器管理列表  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 压力容器管理集合  
 \*/  
 public List<EquipmentVessel> selectEquipmentVesselList(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 新增压力容器管理  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentVessel(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 修改压力容器管理  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentVessel(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 批量删除压力容器管理  
 \*   
 \* @param vesselIds 需要删除的压力容器管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentVesselByVesselIds(Long[] vesselIds);  
  
 /\*\*  
 \* 删除压力容器管理信息  
 \*   
 \* @param vesselId 压力容器管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentVesselByVesselId(Long vesselId);  
}

================================ICostService.java===============================

package com.ruoyi.image.service;  
  
import com.ruoyi.image.domain.Cost;  
  
import java.util.List;  
  
/\*\*  
 \* 报销凭证管理Service接口  
 \*   
 \* @author dzh  
 \* @date 2024-04-14  
 \*/  
public interface ICostService   
{  
 /\*\*  
 \* 查询报销凭证管理  
 \*   
 \* @param costId 报销凭证管理主键  
 \* @return 报销凭证管理  
 \*/  
 public Cost selectCostByCostId(Long costId);  
  
 /\*\*  
 \* 查询报销凭证管理列表  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 报销凭证管理集合  
 \*/  
 public List<Cost> selectCostList(Cost cost);  
  
 /\*\*  
 \* 新增报销凭证管理  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 结果  
 \*/  
 public int insertCost(Cost cost);  
  
 /\*\*  
 \* 修改报销凭证管理  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 结果  
 \*/  
 public int updateCost(Cost cost);  
  
 /\*\*  
 \* 批量删除报销凭证管理  
 \*   
 \* @param costIds 需要删除的报销凭证管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteCostByCostIds(Long[] costIds);  
  
 /\*\*  
 \* 删除报销凭证管理信息  
 \*   
 \* @param costId 报销凭证管理主键  
 \* @return 结果  
 \*/  
 public int deleteCostByCostId(Long costId);  
}

================================IImageService.java===============================

package com.ruoyi.image.service;  
  
import com.ruoyi.image.domain.Image;  
  
import java.util.List;  
  
/\*\*  
 \* 维修凭证管理Service接口  
 \*   
 \* @author dzh  
 \* @date 2024-04-16  
 \*/  
public interface IImageService   
{  
 /\*\*  
 \* 查询维修凭证管理  
 \*   
 \* @param logId 维修凭证管理主键  
 \* @return 维修凭证管理  
 \*/  
 public Image selectImageByLogId(Long logId);  
  
 /\*\*  
 \* 查询维修凭证管理列表  
 \*   
 \* @param image 维修凭证管理  
 \* @return 维修凭证管理集合  
 \*/  
 public List<Image> selectImageList(Image image);  
  
 /\*\*  
 \* 新增维修凭证管理  
 \*   
 \* @param image 维修凭证管理  
 \* @return 结果  
 \*/  
 public int insertImage(Image image);  
  
 /\*\*  
 \* 修改维修凭证管理  
 \*   
 \* @param image 维修凭证管理  
 \* @return 结果  
 \*/  
 public int updateImage(Image image);  
  
 /\*\*  
 \* 批量删除维修凭证管理  
 \*   
 \* @param logIds 需要删除的维修凭证管理主键集合  
 \* @return 结果  
 \*/  
 public int deleteImageByLogIds(Long[] logIds);  
  
 /\*\*  
 \* 删除维修凭证管理信息  
 \*   
 \* @param logId 维修凭证管理主键  
 \* @return 结果  
 \*/  
 public int deleteImageByLogId(Long logId);  
}

=============================EquipmentLiftMapper.java============================package com.ruoyi.equipment.mapper;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentLift;  
  
/\*\*  
 \* 电梯管理Mapper接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface EquipmentLiftMapper   
{  
 /\*\*  
 \* 查询电梯管理  
 \*   
 \* @param liftId 电梯管理主键  
 \* @return 电梯管理  
 \*/  
 public EquipmentLift selectEquipmentLiftByLiftId(Long liftId);  
  
 /\*\*  
 \* 查询电梯管理列表  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 电梯管理集合  
 \*/  
 public List<EquipmentLift> selectEquipmentLiftList(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 新增电梯管理  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentLift(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 修改电梯管理  
 \*   
 \* @param equipmentLift 电梯管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentLift(EquipmentLift equipmentLift);  
  
 /\*\*  
 \* 删除电梯管理  
 \*   
 \* @param liftId 电梯管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentLiftByLiftId(Long liftId);  
  
 /\*\*  
 \* 批量删除电梯管理  
 \*   
 \* @param liftIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentLiftByLiftIds(Long[] liftIds);  
}

===========================EquipmentCraneMapper.java============================

package com.ruoyi.equipment.mapper;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentCrane;  
  
/\*\*  
 \* 起重机管理Mapper接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface EquipmentCraneMapper   
{  
 /\*\*  
 \* 查询起重机管理  
 \*   
 \* @param craneId 起重机管理主键  
 \* @return 起重机管理  
 \*/  
 public EquipmentCrane selectEquipmentCraneByCraneId(Long craneId);  
  
 /\*\*  
 \* 查询起重机管理列表  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 起重机管理集合  
 \*/  
 public List<EquipmentCrane> selectEquipmentCraneList(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 新增起重机管理  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentCrane(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 修改起重机管理  
 \*   
 \* @param equipmentCrane 起重机管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentCrane(EquipmentCrane equipmentCrane);  
  
 /\*\*  
 \* 删除起重机管理  
 \*   
 \* @param craneId 起重机管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCraneByCraneId(Long craneId);  
  
 /\*\*  
 \* 批量删除起重机管理  
 \*   
 \* @param craneIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCraneByCraneIds(Long[] craneIds);  
}

===========================EquipmentVesselMapper.java============================

package com.ruoyi.equipment.mapper;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentVessel;  
  
/\*\*  
 \* 压力容器管理Mapper接口  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public interface EquipmentVesselMapper   
{  
 /\*\*  
 \* 查询压力容器管理  
 \*   
 \* @param vesselId 压力容器管理主键  
 \* @return 压力容器管理  
 \*/  
 public EquipmentVessel selectEquipmentVesselByVesselId(Long vesselId);  
  
 /\*\*  
 \* 查询压力容器管理列表  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 压力容器管理集合  
 \*/  
 public List<EquipmentVessel> selectEquipmentVesselList(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 新增压力容器管理  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentVessel(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 修改压力容器管理  
 \*   
 \* @param equipmentVessel 压力容器管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentVessel(EquipmentVessel equipmentVessel);  
  
 /\*\*  
 \* 删除压力容器管理  
 \*   
 \* @param vesselId 压力容器管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentVesselByVesselId(Long vesselId);  
  
 /\*\*  
 \* 批量删除压力容器管理  
 \*   
 \* @param vesselIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentVesselByVesselIds(Long[] vesselIds);  
}

===========================EquipmentCheckMapper.java============================

package com.ruoyi.equipment.mapper;  
  
import java.util.List;  
import com.ruoyi.equipment.domain.EquipmentCheck;  
  
/\*\*  
 \* 检验类型管理Mapper接口  
 \*   
 \* @author cqq  
 \* @date 2023-09-12  
 \*/  
public interface EquipmentCheckMapper   
{  
 /\*\*  
 \* 查询检验类型管理  
 \*   
 \* @param checkId 检验类型管理主键  
 \* @return 检验类型管理  
 \*/  
 public EquipmentCheck selectEquipmentCheckByCheckId(Integer checkId);  
  
 /\*\*  
 \* 查询检验类型管理列表  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 检验类型管理集合  
 \*/  
 public List<EquipmentCheck> selectEquipmentCheckList(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 新增检验类型管理  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 结果  
 \*/  
 public int insertEquipmentCheck(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 修改检验类型管理  
 \*   
 \* @param equipmentCheck 检验类型管理  
 \* @return 结果  
 \*/  
 public int updateEquipmentCheck(EquipmentCheck equipmentCheck);  
  
 /\*\*  
 \* 删除检验类型管理  
 \*   
 \* @param checkId 检验类型管理主键  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCheckByCheckId(Integer checkId);  
  
 /\*\*  
 \* 批量删除检验类型管理  
 \*   
 \* @param checkIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteEquipmentCheckByCheckIds(Integer[] checkIds);  
}

==============================CostMapper.java===================================

package com.ruoyi.image.mapper;  
  
import com.ruoyi.image.domain.Cost;  
  
import java.util.List;  
  
/\*\*  
 \* 报销凭证管理Mapper接口  
 \*   
 \* @author dzh  
 \* @date 2024-04-14  
 \*/  
public interface CostMapper   
{  
 /\*\*  
 \* 查询报销凭证管理  
 \*   
 \* @param costId 报销凭证管理主键  
 \* @return 报销凭证管理  
 \*/  
 public Cost selectCostByCostId(Long costId);  
  
 /\*\*  
 \* 查询报销凭证管理列表  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 报销凭证管理集合  
 \*/  
 public List<Cost> selectCostList(Cost cost);  
  
 /\*\*  
 \* 新增报销凭证管理  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 结果  
 \*/  
 public int insertCost(Cost cost);  
  
 /\*\*  
 \* 修改报销凭证管理  
 \*   
 \* @param cost 报销凭证管理  
 \* @return 结果  
 \*/  
 public int updateCost(Cost cost);  
  
 /\*\*  
 \* 删除报销凭证管理  
 \*   
 \* @param costId 报销凭证管理主键  
 \* @return 结果  
 \*/  
 public int deleteCostByCostId(Long costId);  
  
 /\*\*  
 \* 批量删除报销凭证管理  
 \*   
 \* @param costIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteCostByCostIds(Long[] costIds);  
}

=============================ImageMapper.java===================================

package com.ruoyi.image.mapper;  
  
import com.ruoyi.image.domain.Image;  
  
import java.util.List;  
  
/\*\*  
 \* 维修凭证管理Mapper接口  
 \*   
 \* @author dzh  
 \* @date 2024-04-16  
 \*/  
public interface ImageMapper   
{  
 /\*\*  
 \* 查询维修凭证管理  
 \*   
 \* @param logId 维修凭证管理主键  
 \* @return 维修凭证管理  
 \*/  
 public Image selectImageByLogId(Long logId);  
  
 /\*\*  
 \* 查询维修凭证管理列表  
 \*   
 \* @param image 维修凭证管理  
 \* @return 维修凭证管理集合  
 \*/  
 public List<Image> selectImageList(Image image);  
  
 /\*\*  
 \* 新增维修凭证管理  
 \*   
 \* @param image 维修凭证管理  
 \* @return 结果  
 \*/  
 public int insertImage(Image image);  
  
 /\*\*  
 \* 修改维修凭证管理  
 \*   
 \* @param image 维修凭证管理  
 \* @return 结果  
 \*/  
 public int updateImage(Image image);  
  
 /\*\*  
 \* 删除维修凭证管理  
 \*   
 \* @param logId 维修凭证管理主键  
 \* @return 结果  
 \*/  
 public int deleteImageByLogId(Long logId);  
  
 /\*\*  
 \* 批量删除维修凭证管理  
 \*   
 \* @param logIds 需要删除的数据主键集合  
 \* @return 结果  
 \*/  
 public int deleteImageByLogIds(Long[] logIds);  
}

================================EquipmentLiftMapper.xml==========================<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.equipment.mapper.EquipmentLiftMapper">  
   
 <resultMap type="EquipmentLift" id="EquipmentLiftResult">  
 <result property="liftId" column="lift\_id" />  
 <result property="deptId" column="dept\_id" />  
 <result property="location" column="location" />  
 <result property="regcode" column="regcode" />  
 <result property="code" column="code" />  
 <result property="administrator" column="administrator" />  
 <result property="phone" column="phone" />  
 <result property="number" column="number" />  
 <result property="liftName" column="lift\_name" />  
 <result property="buyTime" column="buy\_time" />  
 <result property="ratedLoad" column="rated\_load" />  
 <result property="ratedSpeed" column="rated\_speed" />  
 <result property="floor" column="floor" />  
 <result property="checkType" column="check\_type" />  
 <result property="checkTime" column="check\_time" />  
 <result property="maintenanceId" column="maintenance\_id" />  
 <result property="isDrum" column="is\_drum" />  
 <result property="status" column="status" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 </resultMap>  
  
<!-- <sql id="selectEquipmentLiftVo">-->  
<!-- select lift\_id, dept\_id, location, regcode, code, administrator, phone, number, lift\_name, buy\_time, rated\_load, rated\_speed, floor, check\_type, check\_time, maintenance\_id, is\_drum, status, create\_by, create\_time, update\_by, update\_time, remark from equipment\_lift-->  
<!-- </sql>-->  
  
 <sql id="selectEquipmentLiftVo">  
 select lift\_id, dept\_id, location, regcode, code, administrator, phone, number, lift\_name, buy\_time, rated\_load, rated\_speed, floor,  
 (select type from equipment\_check where check\_year=year(check\_time)-year(buy\_time)) as check\_type,  
 check\_time, maintenance\_id, is\_drum, status, create\_by, create\_time, update\_by, update\_time, remark from equipment\_lift  
 </sql>  
  
 <select id="selectEquipmentLiftList" parameterType="EquipmentLift" resultMap="EquipmentLiftResult">  
 <include refid="selectEquipmentLiftVo"/>  
 <where>   
 <if test="location != null and location != ''"> and location like concat('%', #{location}, '%')</if>  
 <if test="administrator != null and administrator != ''"> and administrator like concat('%', #{administrator}, '%')</if>  
 <if test="liftName != null and liftName != ''"> and lift\_name like concat('%', #{liftName}, '%')</if>  
 <if test="checkType != null "> and check\_type = #{checkType}</if>  
 <if test="maintenanceId != null "> and maintenance\_id = #{maintenanceId}</if>  
 <if test="status != null and status != ''"> and status = #{status}</if>  
 </where>  
 </select>  
   
 <select id="selectEquipmentLiftByLiftId" parameterType="Long" resultMap="EquipmentLiftResult">  
 <include refid="selectEquipmentLiftVo"/>  
 where lift\_id = #{liftId}  
 </select>  
   
 <insert id="insertEquipmentLift" parameterType="EquipmentLift" useGeneratedKeys="true" keyProperty="liftId">  
 insert into equipment\_lift  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">dept\_id,</if>  
 <if test="location != null and location != ''">location,</if>  
 <if test="regcode != null">regcode,</if>  
 <if test="code != null">code,</if>  
 <if test="administrator != null">administrator,</if>  
 <if test="phone != null">phone,</if>  
 <if test="number != null">number,</if>  
 <if test="liftName != null">lift\_name,</if>  
 <if test="buyTime != null">buy\_time,</if>  
 <if test="ratedLoad != null">rated\_load,</if>  
 <if test="ratedSpeed != null">rated\_speed,</if>  
 <if test="floor != null">floor,</if>  
 <if test="checkType != null">check\_type,</if>  
 <if test="checkTime != null">check\_time,</if>  
 <if test="maintenanceId != null">maintenance\_id,</if>  
 <if test="isDrum != null">is\_drum,</if>  
 <if test="status != null">status,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">#{deptId},</if>  
 <if test="location != null and location != ''">#{location},</if>  
 <if test="regcode != null">#{regcode},</if>  
 <if test="code != null">#{code},</if>  
 <if test="administrator != null">#{administrator},</if>  
 <if test="phone != null">#{phone},</if>  
 <if test="number != null">#{number},</if>  
 <if test="liftName != null">#{liftName},</if>  
 <if test="buyTime != null">#{buyTime},</if>  
 <if test="ratedLoad != null">#{ratedLoad},</if>  
 <if test="ratedSpeed != null">#{ratedSpeed},</if>  
 <if test="floor != null">#{floor},</if>  
 <if test="checkType != null">#{checkType},</if>  
 <if test="checkTime != null">#{checkTime},</if>  
 <if test="maintenanceId != null">#{maintenanceId},</if>  
 <if test="isDrum != null">#{isDrum},</if>  
 <if test="status != null">#{status},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 </trim>  
 </insert>  
  
 <update id="updateEquipmentLift" parameterType="EquipmentLift">  
 update equipment\_lift  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="deptId != null">dept\_id = #{deptId},</if>  
 <if test="location != null and location != ''">location = #{location},</if>  
 <if test="regcode != null">regcode = #{regcode},</if>  
 <if test="code != null">code = #{code},</if>  
 <if test="administrator != null">administrator = #{administrator},</if>  
 <if test="phone != null">phone = #{phone},</if>  
 <if test="number != null">number = #{number},</if>  
 <if test="liftName != null">lift\_name = #{liftName},</if>  
 <if test="buyTime != null">buy\_time = #{buyTime},</if>  
 <if test="ratedLoad != null">rated\_load = #{ratedLoad},</if>  
 <if test="ratedSpeed != null">rated\_speed = #{ratedSpeed},</if>  
 <if test="floor != null">floor = #{floor},</if>  
 <if test="checkType != null">check\_type = #{checkType},</if>  
 <if test="checkTime != null">check\_time = #{checkTime},</if>  
 <if test="maintenanceId != null">maintenance\_id = #{maintenanceId},</if>  
 <if test="isDrum != null">is\_drum = #{isDrum},</if>  
 <if test="status != null">status = #{status},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 </trim>  
 where lift\_id = #{liftId}  
 </update>  
  
 <delete id="deleteEquipmentLiftByLiftId" parameterType="Long">  
 delete from equipment\_lift where lift\_id = #{liftId}  
 </delete>  
  
 <delete id="deleteEquipmentLiftByLiftIds" parameterType="String">  
 delete from equipment\_lift where lift\_id in   
 <foreach item="liftId" collection="array" open="(" separator="," close=")">  
 #{liftId}  
 </foreach>  
 </delete>  
</mapper>

==============================EquipmentCraneMapper.xml==========================<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.equipment.mapper.EquipmentCraneMapper">  
   
 <resultMap type="EquipmentCrane" id="EquipmentCraneResult">  
 <result property="craneId" column="crane\_id" />  
 <result property="deptId" column="dept\_id" />  
 <result property="location" column="location" />  
 <result property="regcode" column="regcode" />  
 <result property="code" column="code" />  
 <result property="administrator" column="administrator" />  
 <result property="phone" column="phone" />  
 <result property="number" column="number" />  
 <result property="craneName" column="crane\_name" />  
 <result property="buyTime" column="buy\_time" />  
 <result property="weight" column="weight" />  
 <result property="factory" column="factory" />  
 <result property="type" column="type" />  
 <result property="liftHeight" column="lift\_height" />  
 <result property="span" column="span" />  
 <result property="liftSpeed" column="lift\_speed" />  
 <result property="checkType" column="check\_type" />  
 <result property="checkTime" column="check\_time" />  
 <result property="maintenanceId" column="maintenance\_id" />  
 <result property="status" column="status" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 </resultMap>  
  
<!-- <sql id="selectEquipmentCraneVo">-->  
<!-- select crane\_id, dept\_id, location, regcode, code, administrator, phone, number, crane\_name, buy\_time, weight, factory, type, lift\_height, span, lift\_speed, check\_type, check\_time, maintenance\_id, status, create\_by, create\_time, update\_by, update\_time, remark from equipment\_crane-->  
<!-- </sql>-->  
  
 <sql id="selectEquipmentCraneVo">  
 select crane\_id, dept\_id, location, regcode, code, administrator, phone, number, crane\_name, buy\_time, weight, factory, type, lift\_height, span, lift\_speed, 1 as check\_type, check\_time, maintenance\_id, status, create\_by, create\_time, update\_by, update\_time, remark from equipment\_crane  
 </sql>  
  
 <select id="selectEquipmentCraneList" parameterType="EquipmentCrane" resultMap="EquipmentCraneResult">  
 <include refid="selectEquipmentCraneVo"/>  
 <where>   
 <if test="deptId != null "> and dept\_id = #{deptId}</if>  
 <if test="location != null and location != ''"> and location like concat('%', #{location}, '%')</if>  
 <if test="administrator != null and administrator != ''"> and administrator like concat('%', #{administrator}, '%')</if>  
 <if test="craneName != null and craneName != ''"> and crane\_name like concat('%', #{craneName}, '%')</if>  
 <if test="status != null and status != ''"> and status = #{status}</if>  
 </where>  
 </select>  
   
 <select id="selectEquipmentCraneByCraneId" parameterType="Long" resultMap="EquipmentCraneResult">  
 <include refid="selectEquipmentCraneVo"/>  
 where crane\_id = #{craneId}  
 </select>  
   
 <insert id="insertEquipmentCrane" parameterType="EquipmentCrane" useGeneratedKeys="true" keyProperty="craneId">  
 insert into equipment\_crane  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">dept\_id,</if>  
 <if test="location != null">location,</if>  
 <if test="regcode != null">regcode,</if>  
 <if test="code != null">code,</if>  
 <if test="administrator != null">administrator,</if>  
 <if test="phone != null">phone,</if>  
 <if test="number != null">number,</if>  
 <if test="craneName != null">crane\_name,</if>  
 <if test="buyTime != null">buy\_time,</if>  
 <if test="weight != null">weight,</if>  
 <if test="factory != null">factory,</if>  
 <if test="type != null">type,</if>  
 <if test="liftHeight != null">lift\_height,</if>  
 <if test="span != null">span,</if>  
 <if test="liftSpeed != null">lift\_speed,</if>  
 <if test="checkType != null">check\_type,</if>  
 <if test="checkTime != null">check\_time,</if>  
 <if test="maintenanceId != null">maintenance\_id,</if>  
 <if test="status != null">status,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">#{deptId},</if>  
 <if test="location != null">#{location},</if>  
 <if test="regcode != null">#{regcode},</if>  
 <if test="code != null">#{code},</if>  
 <if test="administrator != null">#{administrator},</if>  
 <if test="phone != null">#{phone},</if>  
 <if test="number != null">#{number},</if>  
 <if test="craneName != null">#{craneName},</if>  
 <if test="buyTime != null">#{buyTime},</if>  
 <if test="weight != null">#{weight},</if>  
 <if test="factory != null">#{factory},</if>  
 <if test="type != null">#{type},</if>  
 <if test="liftHeight != null">#{liftHeight},</if>  
 <if test="span != null">#{span},</if>  
 <if test="liftSpeed != null">#{liftSpeed},</if>  
 <if test="checkType != null">#{checkType},</if>  
 <if test="checkTime != null">#{checkTime},</if>  
 <if test="maintenanceId != null">#{maintenanceId},</if>  
 <if test="status != null">#{status},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 </trim>  
 </insert>  
  
 <update id="updateEquipmentCrane" parameterType="EquipmentCrane">  
 update equipment\_crane  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="deptId != null">dept\_id = #{deptId},</if>  
 <if test="location != null">location = #{location},</if>  
 <if test="regcode != null">regcode = #{regcode},</if>  
 <if test="code != null">code = #{code},</if>  
 <if test="administrator != null">administrator = #{administrator},</if>  
 <if test="phone != null">phone = #{phone},</if>  
 <if test="number != null">number = #{number},</if>  
 <if test="craneName != null">crane\_name = #{craneName},</if>  
 <if test="buyTime != null">buy\_time = #{buyTime},</if>  
 <if test="weight != null">weight = #{weight},</if>  
 <if test="factory != null">factory = #{factory},</if>  
 <if test="type != null">type = #{type},</if>  
 <if test="liftHeight != null">lift\_height = #{liftHeight},</if>  
 <if test="span != null">span = #{span},</if>  
 <if test="liftSpeed != null">lift\_speed = #{liftSpeed},</if>  
 <if test="checkType != null">check\_type = #{checkType},</if>  
 <if test="checkTime != null">check\_time = #{checkTime},</if>  
 <if test="maintenanceId != null">maintenance\_id = #{maintenanceId},</if>  
 <if test="status != null">status = #{status},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 </trim>  
 where crane\_id = #{craneId}  
 </update>  
  
 <delete id="deleteEquipmentCraneByCraneId" parameterType="Long">  
 delete from equipment\_crane where crane\_id = #{craneId}  
 </delete>  
  
 <delete id="deleteEquipmentCraneByCraneIds" parameterType="String">  
 delete from equipment\_crane where crane\_id in   
 <foreach item="craneId" collection="array" open="(" separator="," close=")">  
 #{craneId}  
 </foreach>  
 </delete>  
</mapper>

=============================EquipmentVesselMapper.xml==========================

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.equipment.mapper.EquipmentVesselMapper">  
   
 <resultMap type="EquipmentVessel" id="EquipmentVesselResult">  
 <result property="vesselId" column="vessel\_id" />  
 <result property="deptId" column="dept\_id" />  
 <result property="area" column="area" />  
 <result property="address" column="address" />  
 <result property="code" column="code" />  
 <result property="administrator" column="administrator" />  
 <result property="phone" column="phone" />  
 <result property="factoryNumber" column="factory\_number" />  
 <result property="regcode" column="regcode" />  
 <result property="regStatus" column="reg\_status" />  
 <result property="vesselName" column="vessel\_name" />  
 <result property="buyTime" column="buy\_time" />  
 <result property="license" column="license" />  
 <result property="type" column="type" />  
 <result property="typeNumber" column="type\_number" />  
 <result property="attribute" column="attribute" />  
 <result property="checkType" column="check\_type" />  
 <result property="checkTime" column="check\_time" />  
 <result property="maintenanceId" column="maintenance\_id" />  
 <result property="status" column="status" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 </resultMap>  
  
 <sql id="selectEquipmentVesselVo">  
 select vessel\_id, dept\_id, area, address, code, administrator, phone, factory\_number, regcode, reg\_status, vessel\_name, buy\_time, license, type, type\_number, attribute, check\_type, check\_time, maintenance\_id, status, create\_by, create\_time, update\_by, update\_time, remark from equipment\_vessel  
 </sql>  
  
 <select id="selectEquipmentVesselList" parameterType="EquipmentVessel" resultMap="EquipmentVesselResult">  
 <include refid="selectEquipmentVesselVo"/>  
 <where>   
 <if test="deptId != null "> and dept\_id = #{deptId}</if>  
 <if test="address != null and address != ''"> and address like concat('%', #{address}, '%')</if>  
 <if test="code != null "> and code = #{code}</if>  
 <if test="administrator != null and administrator != ''"> and administrator like concat('%', #{administrator}, '%')</if>  
 <if test="vesselName != null and vesselName != ''"> and vessel\_name like concat('%', #{vesselName}, '%')</if>  
 <if test="checkType != null "> and check\_type = #{checkType}</if>  
 <if test="maintenanceId != null "> and maintenance\_id = #{maintenanceId}</if>  
 <if test="status != null and status != ''"> and status = #{status}</if>  
 </where>  
 </select>  
   
 <select id="selectEquipmentVesselByVesselId" parameterType="Long" resultMap="EquipmentVesselResult">  
 <include refid="selectEquipmentVesselVo"/>  
 where vessel\_id = #{vesselId}  
 </select>  
   
 <insert id="insertEquipmentVessel" parameterType="EquipmentVessel" useGeneratedKeys="true" keyProperty="vesselId">  
 insert into equipment\_vessel  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">dept\_id,</if>  
 <if test="area != null and area != ''">area,</if>  
 <if test="address != null and address != ''">address,</if>  
 <if test="code != null and code != ''">code,</if>  
 <if test="administrator != null">administrator,</if>  
 <if test="phone != null">phone,</if>  
 <if test="factoryNumber != null">factory\_number,</if>  
 <if test="regcode != null">regcode,</if>  
 <if test="regStatus != null">reg\_status,</if>  
 <if test="vesselName != null">vessel\_name,</if>  
 <if test="buyTime != null">buy\_time,</if>  
 <if test="license != null">license,</if>  
 <if test="type != null">type,</if>  
 <if test="typeNumber != null">type\_number,</if>  
 <if test="attribute != null">attribute,</if>  
 <if test="checkType != null">check\_type,</if>  
 <if test="checkTime != null">check\_time,</if>  
 <if test="maintenanceId != null">maintenance\_id,</if>  
 <if test="status != null">status,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="deptId != null">#{deptId},</if>  
 <if test="area != null and area != ''">#{area},</if>  
 <if test="address != null and address != ''">#{address},</if>  
 <if test="code != null and code != ''">#{code},</if>  
 <if test="administrator != null">#{administrator},</if>  
 <if test="phone != null">#{phone},</if>  
 <if test="factoryNumber != null">#{factoryNumber},</if>  
 <if test="regcode != null">#{regcode},</if>  
 <if test="regStatus != null">#{regStatus},</if>  
 <if test="vesselName != null">#{vesselName},</if>  
 <if test="buyTime != null">#{buyTime},</if>  
 <if test="license != null">#{license},</if>  
 <if test="type != null">#{type},</if>  
 <if test="typeNumber != null">#{typeNumber},</if>  
 <if test="attribute != null">#{attribute},</if>  
 <if test="checkType != null">#{checkType},</if>  
 <if test="checkTime != null">#{checkTime},</if>  
 <if test="maintenanceId != null">#{maintenanceId},</if>  
 <if test="status != null">#{status},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 </trim>  
 </insert>  
  
 <update id="updateEquipmentVessel" parameterType="EquipmentVessel">  
 update equipment\_vessel  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="deptId != null">dept\_id = #{deptId},</if>  
 <if test="area != null and area != ''">area = #{area},</if>  
 <if test="address != null and address != ''">address = #{address},</if>  
 <if test="code != null and code != ''">code = #{code},</if>  
 <if test="administrator != null">administrator = #{administrator},</if>  
 <if test="phone != null">phone = #{phone},</if>  
 <if test="factoryNumber != null">factory\_number = #{factoryNumber},</if>  
 <if test="regcode != null">regcode = #{regcode},</if>  
 <if test="regStatus != null">reg\_status = #{regStatus},</if>  
 <if test="vesselName != null">vessel\_name = #{vesselName},</if>  
 <if test="buyTime != null">buy\_time = #{buyTime},</if>  
 <if test="license != null">license = #{license},</if>  
 <if test="type != null">type = #{type},</if>  
 <if test="typeNumber != null">type\_number = #{typeNumber},</if>  
 <if test="attribute != null">attribute = #{attribute},</if>  
 <if test="checkType != null">check\_type = #{checkType},</if>  
 <if test="checkTime != null">check\_time = #{checkTime},</if>  
 <if test="maintenanceId != null">maintenance\_id = #{maintenanceId},</if>  
 <if test="status != null">status = #{status},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 </trim>  
 where vessel\_id = #{vesselId}  
 </update>  
  
 <delete id="deleteEquipmentVesselByVesselId" parameterType="Long">  
 delete from equipment\_vessel where vessel\_id = #{vesselId}  
 </delete>  
  
 <delete id="deleteEquipmentVesselByVesselIds" parameterType="String">  
 delete from equipment\_vessel where vessel\_id in   
 <foreach item="vesselId" collection="array" open="(" separator="," close=")">  
 #{vesselId}  
 </foreach>  
 </delete>  
</mapper>

=============================EquipmentCheckMapper.xml==========================

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.equipment.mapper.EquipmentCheckMapper">  
   
 <resultMap type="EquipmentCheck" id="EquipmentCheckResult">  
 <result property="checkId" column="check\_id" />  
 <result property="checkYear" column="check\_year" />  
 <result property="type" column="type" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 </resultMap>  
  
 <sql id="selectEquipmentCheckVo">  
 select check\_id, check\_year, type, create\_by, create\_time, update\_by, update\_time, remark from equipment\_check  
 </sql>  
  
 <select id="selectEquipmentCheckList" parameterType="EquipmentCheck" resultMap="EquipmentCheckResult">  
 <include refid="selectEquipmentCheckVo"/>  
 <where>   
 <if test="checkYear != null "> and check\_year = #{checkYear}</if>  
 <if test="type != null "> and type = #{type}</if>  
 </where>  
 </select>  
   
 <select id="selectEquipmentCheckByCheckId" parameterType="Integer" resultMap="EquipmentCheckResult">  
 <include refid="selectEquipmentCheckVo"/>  
 where check\_id = #{checkId}  
 </select>  
   
 <insert id="insertEquipmentCheck" parameterType="EquipmentCheck">  
 insert into equipment\_check  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="checkId != null">check\_id,</if>  
 <if test="checkYear != null">check\_year,</if>  
 <if test="type != null">type,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="checkId != null">#{checkId},</if>  
 <if test="checkYear != null">#{checkYear},</if>  
 <if test="type != null">#{type},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 </trim>  
 </insert>  
  
 <update id="updateEquipmentCheck" parameterType="EquipmentCheck">  
 update equipment\_check  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="checkYear != null">check\_year = #{checkYear},</if>  
 <if test="type != null">type = #{type},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 </trim>  
 where check\_id = #{checkId}  
 </update>  
  
 <delete id="deleteEquipmentCheckByCheckId" parameterType="Integer">  
 delete from equipment\_check where check\_id = #{checkId}  
 </delete>  
  
 <delete id="deleteEquipmentCheckByCheckIds" parameterType="String">  
 delete from equipment\_check where check\_id in   
 <foreach item="checkId" collection="array" open="(" separator="," close=")">  
 #{checkId}  
 </foreach>  
 </delete>  
</mapper>

=================================CostMapper.xml=================================

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.image.mapper.CostMapper">  
   
 <resultMap type="Cost" id="CostResult">  
 <result property="costId" column="cost\_id" />  
 <result property="user" column="user" />  
 <result property="costType" column="cost\_type" />  
 <result property="url" column="url" />  
 <result property="checkTime" column="check\_time" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 <result property="userId" column="user\_id" />  
 </resultMap>  
  
 <sql id="selectCostVo">  
 select cost\_id, user, cost\_type, url, check\_time, create\_by, create\_time, update\_by, update\_time, remark, user\_id from cost costTable  
 </sql>  
  
 <select id="selectCostList" parameterType="Cost" resultMap="CostResult">  
 <include refid="selectCostVo"/>  
 <where>   
 <if test="user != null and user != ''"> and user = #{user}</if>  
 <if test="costType != null and costType != ''"> and cost\_type = #{costType}</if>  
 <if test="url != null and url != ''"> and url = #{url}</if>  
 <if test="checkTime != null "> and check\_time = #{checkTime}</if>  
 ${params.dataScope}  
 </where>  
 </select>  
   
 <select id="selectCostByCostId" parameterType="Long" resultMap="CostResult">  
 <include refid="selectCostVo"/>  
 where cost\_id = #{costId}  
 </select>  
   
 <insert id="insertCost" parameterType="Cost">  
 insert into cost  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="costId != null">cost\_id,</if>  
 <if test="user != null and user != ''">user,</if>  
 <if test="costType != null and costType != ''">cost\_type,</if>  
 <if test="url != null and url != ''">url,</if>  
 <if test="checkTime != null">check\_time,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 <if test="userId != null">user\_id,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="costId != null">#{costId},</if>  
 <if test="user != null and user != ''">#{user},</if>  
 <if test="costType != null and costType != ''">#{costType},</if>  
 <if test="url != null and url != ''">#{url},</if>  
 <if test="checkTime != null">#{checkTime},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 <if test="userId != null">#{userId},</if>  
 </trim>  
 </insert>  
  
 <update id="updateCost" parameterType="Cost">  
 update cost  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="user != null and user != ''">user = #{user},</if>  
 <if test="costType != null and costType != ''">cost\_type = #{costType},</if>  
 <if test="url != null and url != ''">url = #{url},</if>  
 <if test="checkTime != null">check\_time = #{checkTime},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 <if test="userId != null">user\_id = #{userId},</if>  
 </trim>  
 where cost\_id = #{costId}  
 </update>  
  
 <delete id="deleteCostByCostId" parameterType="Long">  
 delete from cost where cost\_id = #{costId}  
 </delete>  
  
 <delete id="deleteCostByCostIds" parameterType="String">  
 delete from cost where cost\_id in   
 <foreach item="costId" collection="array" open="(" separator="," close=")">  
 #{costId}  
 </foreach>  
 </delete>  
</mapper>

=================================ImageMapper.xml===============================

<?xml version="1.0" encoding="UTF-8" ?>  
<!DOCTYPE mapper  
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"  
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">  
<mapper namespace="com.ruoyi.image.mapper.ImageMapper">  
   
 <resultMap type="Image" id="ImageResult">  
 <result property="logId" column="log\_id" />  
 <result property="user" column="user" />  
 <result property="equipmentType" column="equipment\_type" />  
 <result property="location" column="location" />  
 <result property="url" column="url" />  
 <result property="checkTime" column="check\_time" />  
 <result property="createBy" column="create\_by" />  
 <result property="createTime" column="create\_time" />  
 <result property="updateBy" column="update\_by" />  
 <result property="updateTime" column="update\_time" />  
 <result property="remark" column="remark" />  
 <result property="userId" column="user\_id" />  
 </resultMap>  
  
 <sql id="selectImageVo">  
 select log\_id, user, equipment\_type, location, url, check\_time, create\_by, create\_time, update\_by, update\_time, remark, user\_id from image t  
 </sql>  
  
 <select id="selectImageList" parameterType="Image" resultMap="ImageResult">  
 <include refid="selectImageVo"/>  
 <where>   
 <if test="user != null and user != ''"> and user = #{user}</if>  
 <if test="equipmentType != null "> and equipment\_type = #{equipmentType}</if>  
 <if test="location != null "> and location = #{location}</if>  
 <if test="url != null and url != ''"> and url = #{url}</if>  
 <if test="checkTime != null "> and check\_time = #{checkTime}</if>  
 <if test="userId != null "> and user\_id = #{userId}</if>  
 ${params.dataScope}  
 </where>  
  
 </select>  
   
 <select id="selectImageByLogId" parameterType="Long" resultMap="ImageResult">  
 <include refid="selectImageVo"/>  
 where log\_id = #{logId}  
 </select>  
   
 <insert id="insertImage" parameterType="Image" useGeneratedKeys="true" keyProperty="logId">  
 insert into image  
 <trim prefix="(" suffix=")" suffixOverrides=",">  
 <if test="user != null">user,</if>  
 <if test="equipmentType != null">equipment\_type,</if>  
 <if test="location != null">location,</if>  
 <if test="url != null and url != ''">url,</if>  
 <if test="checkTime != null">check\_time,</if>  
 <if test="createBy != null">create\_by,</if>  
 <if test="createTime != null">create\_time,</if>  
 <if test="updateBy != null">update\_by,</if>  
 <if test="updateTime != null">update\_time,</if>  
 <if test="remark != null">remark,</if>  
 <if test="userId != null">user\_id,</if>  
 </trim>  
 <trim prefix="values (" suffix=")" suffixOverrides=",">  
 <if test="user != null">#{user},</if>  
 <if test="equipmentType != null">#{equipmentType},</if>  
 <if test="location != null">#{location},</if>  
 <if test="url != null and url != ''">#{url},</if>  
 <if test="checkTime != null">#{checkTime},</if>  
 <if test="createBy != null">#{createBy},</if>  
 <if test="createTime != null">#{createTime},</if>  
 <if test="updateBy != null">#{updateBy},</if>  
 <if test="updateTime != null">#{updateTime},</if>  
 <if test="remark != null">#{remark},</if>  
 <if test="userId != null">#{userId},</if>  
 </trim>  
 </insert>  
  
 <update id="updateImage" parameterType="Image">  
 update image  
 <trim prefix="SET" suffixOverrides=",">  
 <if test="user != null">user = #{user},</if>  
 <if test="equipmentType != null">equipment\_type = #{equipmentType},</if>  
 <if test="location != null">location = #{location},</if>  
 <if test="url != null and url != ''">url = #{url},</if>  
 <if test="checkTime != null">check\_time = #{checkTime},</if>  
 <if test="createBy != null">create\_by = #{createBy},</if>  
 <if test="createTime != null">create\_time = #{createTime},</if>  
 <if test="updateBy != null">update\_by = #{updateBy},</if>  
 <if test="updateTime != null">update\_time = #{updateTime},</if>  
 <if test="remark != null">remark = #{remark},</if>  
 <if test="userId != null">user\_id = #{userId},</if>  
 </trim>  
 where log\_id = #{logId}  
 </update>  
  
 <delete id="deleteImageByLogId" parameterType="Long">  
 delete from image where log\_id = #{logId}  
 </delete>  
  
 <delete id="deleteImageByLogIds" parameterType="String">  
 delete from image where log\_id in   
 <foreach item="logId" collection="array" open="(" separator="," close=")">  
 #{logId}  
 </foreach>  
 </delete>  
</mapper>

=================================EquipmentLift.java===============================

package com.ruoyi.equipment.domain;  
  
import java.util.Date;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
  
/\*\*  
 \* 电梯管理对象 equipment\_lift  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public class EquipmentLift extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 序号ID \*/  
 private Long liftId;  
  
 /\*\* 分布单位 \*/  
 @Excel(name = "分布单位")  
 private Long deptId;  
  
 /\*\* 使用地点 \*/  
 @Excel(name = "使用地点")  
 private String location;  
  
 /\*\* 注册代码 \*/  
 @Excel(name = "注册代码")  
 private String regcode;  
  
 /\*\* 设备代码 \*/  
 @Excel(name = "设备代码")  
 private String code;  
  
 /\*\* 管理人员 \*/  
 @Excel(name = "管理人员")  
 private String administrator;  
  
 /\*\* 联系方式 \*/  
 @Excel(name = "联系方式")  
 private String phone;  
  
 /\*\* 设备编码 \*/  
 @Excel(name = "设备编码")  
 private String number;  
  
 /\*\* 设备名称 \*/  
 @Excel(name = "设备名称")  
 private String liftName;  
  
 /\*\* 购置日期 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "购置日期", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date buyTime;  
  
 /\*\* 额定载程 \*/  
 @Excel(name = "额定载程")  
 private String ratedLoad;  
  
 /\*\* 额定速度 \*/  
 @Excel(name = "额定速度")  
 private String ratedSpeed;  
  
 /\*\* 层站 \*/  
 @Excel(name = "层站")  
 private String floor;  
  
 /\*\* 年检类型（1检验，2检测，3检验与检测） \*/  
 @Excel(name = "年检类型", readConverterExp = "1=检验，2检测，3检验与检测")  
 private Integer checkType;  
  
 /\*\* 下次年检时间 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "下次年检时间", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date checkTime;  
  
 /\*\* 维保单位 \*/  
 @Excel(name = "维保单位")  
 private Long maintenanceId;  
  
 /\*\* 是否鼓式制动机（0不是，1是） \*/  
 @Excel(name = "是否鼓式制动机", readConverterExp = "0=不是，1是")  
 private String isDrum;  
  
 /\*\* 状态（0正常，1停用，2其他状态） \*/  
 @Excel(name = "状态", readConverterExp = "0=正常，1停用，2其他状态")  
 private String status;  
  
 public void setLiftId(Long liftId)   
 {  
 this.liftId = liftId;  
 }  
  
 public Long getLiftId()   
 {  
 return liftId;  
 }  
 public void setDeptId(Long deptId)   
 {  
 this.deptId = deptId;  
 }  
  
 public Long getDeptId()   
 {  
 return deptId;  
 }  
 public void setLocation(String location)   
 {  
 this.location = location;  
 }  
  
 public String getLocation()   
 {  
 return location;  
 }  
 public void setRegcode(String regcode)   
 {  
 this.regcode = regcode;  
 }  
  
 public String getRegcode()   
 {  
 return regcode;  
 }  
 public void setCode(String code)   
 {  
 this.code = code;  
 }  
  
 public String getCode()   
 {  
 return code;  
 }  
 public void setAdministrator(String administrator)   
 {  
 this.administrator = administrator;  
 }  
  
 public String getAdministrator()   
 {  
 return administrator;  
 }  
 public void setPhone(String phone)   
 {  
 this.phone = phone;  
 }  
  
 public String getPhone()   
 {  
 return phone;  
 }  
 public void setNumber(String number)   
 {  
 this.number = number;  
 }  
  
 public String getNumber()   
 {  
 return number;  
 }  
 public void setLiftName(String liftName)   
 {  
 this.liftName = liftName;  
 }  
  
 public String getLiftName()   
 {  
 return liftName;  
 }  
 public void setBuyTime(Date buyTime)   
 {  
 this.buyTime = buyTime;  
 }  
  
 public Date getBuyTime()   
 {  
 return buyTime;  
 }  
 public void setRatedLoad(String ratedLoad)   
 {  
 this.ratedLoad = ratedLoad;  
 }  
  
 public String getRatedLoad()   
 {  
 return ratedLoad;  
 }  
 public void setRatedSpeed(String ratedSpeed)   
 {  
 this.ratedSpeed = ratedSpeed;  
 }  
  
 public String getRatedSpeed()   
 {  
 return ratedSpeed;  
 }  
 public void setFloor(String floor)   
 {  
 this.floor = floor;  
 }  
  
 public String getFloor()   
 {  
 return floor;  
 }  
 public void setCheckType(Integer checkType)   
 {  
 this.checkType = checkType;  
 }  
  
 public Integer getCheckType()   
 {  
 return checkType;  
 }  
 public void setCheckTime(Date checkTime)   
 {  
 this.checkTime = checkTime;  
 }  
  
 public Date getCheckTime()   
 {  
 return checkTime;  
 }  
 public void setMaintenanceId(Long maintenanceId)   
 {  
 this.maintenanceId = maintenanceId;  
 }  
  
 public Long getMaintenanceId()   
 {  
 return maintenanceId;  
 }  
 public void setIsDrum(String isDrum)   
 {  
 this.isDrum = isDrum;  
 }  
  
 public String getIsDrum()   
 {  
 return isDrum;  
 }  
 public void setStatus(String status)   
 {  
 this.status = status;  
 }  
  
 public String getStatus()   
 {  
 return status;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("liftId", getLiftId())  
 .append("deptId", getDeptId())  
 .append("location", getLocation())  
 .append("regcode", getRegcode())  
 .append("code", getCode())  
 .append("administrator", getAdministrator())  
 .append("phone", getPhone())  
 .append("number", getNumber())  
 .append("liftName", getLiftName())  
 .append("buyTime", getBuyTime())  
 .append("ratedLoad", getRatedLoad())  
 .append("ratedSpeed", getRatedSpeed())  
 .append("floor", getFloor())  
 .append("checkType", getCheckType())  
 .append("checkTime", getCheckTime())  
 .append("maintenanceId", getMaintenanceId())  
 .append("isDrum", getIsDrum())  
 .append("status", getStatus())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .toString();  
 }  
}

===============================EquipmentCrane.java===============================

package com.ruoyi.equipment.domain;  
  
import java.util.Date;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
  
/\*\*  
 \* 起重机管理对象 equipment\_crane  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public class EquipmentCrane extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 序号ID \*/  
 private Long craneId;  
  
 /\*\* 分布单位 \*/  
 @Excel(name = "分布单位")  
 private Long deptId;  
  
 /\*\* 使用地点 \*/  
 @Excel(name = "使用地点")  
 private String location;  
  
 /\*\* 注册代码 \*/  
 @Excel(name = "注册代码")  
 private String regcode;  
  
 /\*\* 设备代码 \*/  
 @Excel(name = "设备代码")  
 private String code;  
  
 /\*\* 管理人员 \*/  
 @Excel(name = "管理人员")  
 private String administrator;  
  
 /\*\* 联系方式 \*/  
 @Excel(name = "联系方式")  
 private String phone;  
  
 /\*\* 设备编码 \*/  
 @Excel(name = "设备编码")  
 private String number;  
  
 /\*\* 设备名称 \*/  
 @Excel(name = "设备名称")  
 private String craneName;  
  
 /\*\* 购置日期 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "购置日期", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date buyTime;  
  
 /\*\* 规格 \*/  
 @Excel(name = "规格")  
 private String weight;  
  
 /\*\* 生产厂家 \*/  
 @Excel(name = "生产厂家")  
 private String factory;  
  
 /\*\* 型号规格 \*/  
 @Excel(name = "型号规格")  
 private String type;  
  
 /\*\* 起升高度 \*/  
 @Excel(name = "起升高度")  
 private String liftHeight;  
  
 /\*\* 跨度 \*/  
 @Excel(name = "跨度")  
 private String span;  
  
 /\*\* 起升速度 \*/  
 @Excel(name = "起升速度")  
 private String liftSpeed;  
  
 /\*\* 年检类型（1检验，2检测，3检验与检测） \*/  
 @Excel(name = "年检类型", readConverterExp = "1=检验，2检测，3检验与检测")  
 private Integer checkType;  
  
 /\*\* 下次年检时间 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "下次年检时间", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date checkTime;  
  
 /\*\* 维保单位 \*/  
 @Excel(name = "维保单位")  
 private Long maintenanceId;  
  
 /\*\* 状态（0正常，1停用，2其他状态） \*/  
 @Excel(name = "状态", readConverterExp = "0=正常，1停用，2其他状态")  
 private String status;  
  
 public void setCraneId(Long craneId)   
 {  
 this.craneId = craneId;  
 }  
  
 public Long getCraneId()   
 {  
 return craneId;  
 }  
 public void setDeptId(Long deptId)   
 {  
 this.deptId = deptId;  
 }  
  
 public Long getDeptId()   
 {  
 return deptId;  
 }  
 public void setLocation(String location)   
 {  
 this.location = location;  
 }  
  
 public String getLocation()   
 {  
 return location;  
 }  
 public void setRegcode(String regcode)   
 {  
 this.regcode = regcode;  
 }  
  
 public String getRegcode()   
 {  
 return regcode;  
 }  
 public void setCode(String code)   
 {  
 this.code = code;  
 }  
  
 public String getCode()   
 {  
 return code;  
 }  
 public void setAdministrator(String administrator)   
 {  
 this.administrator = administrator;  
 }  
  
 public String getAdministrator()   
 {  
 return administrator;  
 }  
 public void setPhone(String phone)   
 {  
 this.phone = phone;  
 }  
  
 public String getPhone()   
 {  
 return phone;  
 }  
 public void setNumber(String number)   
 {  
 this.number = number;  
 }  
  
 public String getNumber()   
 {  
 return number;  
 }  
 public void setCraneName(String craneName)   
 {  
 this.craneName = craneName;  
 }  
  
 public String getCraneName()   
 {  
 return craneName;  
 }  
 public void setBuyTime(Date buyTime)   
 {  
 this.buyTime = buyTime;  
 }  
  
 public Date getBuyTime()   
 {  
 return buyTime;  
 }  
 public void setWeight(String weight)   
 {  
 this.weight = weight;  
 }  
  
 public String getWeight()   
 {  
 return weight;  
 }  
 public void setFactory(String factory)   
 {  
 this.factory = factory;  
 }  
  
 public String getFactory()   
 {  
 return factory;  
 }  
 public void setType(String type)   
 {  
 this.type = type;  
 }  
  
 public String getType()   
 {  
 return type;  
 }  
 public void setLiftHeight(String liftHeight)   
 {  
 this.liftHeight = liftHeight;  
 }  
  
 public String getLiftHeight()   
 {  
 return liftHeight;  
 }  
 public void setSpan(String span)   
 {  
 this.span = span;  
 }  
  
 public String getSpan()   
 {  
 return span;  
 }  
 public void setLiftSpeed(String liftSpeed)   
 {  
 this.liftSpeed = liftSpeed;  
 }  
  
 public String getLiftSpeed()   
 {  
 return liftSpeed;  
 }  
 public void setCheckType(Integer checkType)   
 {  
 this.checkType = checkType;  
 }  
  
 public Integer getCheckType()   
 {  
 return checkType;  
 }  
 public void setCheckTime(Date checkTime)   
 {  
 this.checkTime = checkTime;  
 }  
  
 public Date getCheckTime()   
 {  
 return checkTime;  
 }  
 public void setMaintenanceId(Long maintenanceId)   
 {  
 this.maintenanceId = maintenanceId;  
 }  
  
 public Long getMaintenanceId()   
 {  
 return maintenanceId;  
 }  
 public void setStatus(String status)   
 {  
 this.status = status;  
 }  
  
 public String getStatus()   
 {  
 return status;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("craneId", getCraneId())  
 .append("deptId", getDeptId())  
 .append("location", getLocation())  
 .append("regcode", getRegcode())  
 .append("code", getCode())  
 .append("administrator", getAdministrator())  
 .append("phone", getPhone())  
 .append("number", getNumber())  
 .append("craneName", getCraneName())  
 .append("buyTime", getBuyTime())  
 .append("weight", getWeight())  
 .append("factory", getFactory())  
 .append("type", getType())  
 .append("liftHeight", getLiftHeight())  
 .append("span", getSpan())  
 .append("liftSpeed", getLiftSpeed())  
 .append("checkType", getCheckType())  
 .append("checkTime", getCheckTime())  
 .append("maintenanceId", getMaintenanceId())  
 .append("status", getStatus())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .toString();  
 }  
}

===============================EquipmentVessel.java==============================

package com.ruoyi.equipment.domain;  
  
import java.util.Date;  
import com.fasterxml.jackson.annotation.JsonFormat;  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
  
/\*\*  
 \* 压力容器管理对象 equipment\_vessel  
 \*   
 \* @author cqq  
 \* @date 2023-12-20  
 \*/  
public class EquipmentVessel extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 序号ID \*/  
 private Long vesselId;  
  
 /\*\* 分布单位 \*/  
 @Excel(name = "分布单位")  
 private Long deptId;  
  
 /\*\* 设备所在地区 \*/  
 @Excel(name = "设备所在地区")  
 private String area;  
  
 /\*\* 设备详细地址 \*/  
 @Excel(name = "设备详细地址")  
 private String address;  
  
 /\*\* 设备代码 \*/  
 @Excel(name = "设备代码")  
 private String code;  
  
 /\*\* 管理人员 \*/  
 @Excel(name = "管理人员")  
 private String administrator;  
  
 /\*\* 联系方式 \*/  
 @Excel(name = "联系方式")  
 private String phone;  
  
 /\*\* 出厂编码 \*/  
 @Excel(name = "出厂编码")  
 private String factoryNumber;  
  
 /\*\* 注册编码 \*/  
 @Excel(name = "注册编码")  
 private String regcode;  
  
 /\*\* 注册状态 \*/  
 @Excel(name = "注册状态")  
 private String regStatus;  
  
 /\*\* 设备名称 \*/  
 @Excel(name = "设备名称")  
 private String vesselName;  
  
 /\*\* 购置日期 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "购置日期", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date buyTime;  
  
 /\*\* 使用证编号 \*/  
 @Excel(name = "使用证编号")  
 private String license;  
  
 /\*\* 设备类型 \*/  
 @Excel(name = "设备类型")  
 private String type;  
  
 /\*\* 设备型号 \*/  
 @Excel(name = "设备型号")  
 private String typeNumber;  
  
 /\*\* 设备属性 \*/  
 @Excel(name = "设备属性")  
 private String attribute;  
  
 /\*\* 年检类型（1检验，2检测，3检验与检测） \*/  
 @Excel(name = "年检类型", readConverterExp = "1=检验，2检测，3检验与检测")  
 private Integer checkType;  
  
 /\*\* 下次年检时间 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "下次年检时间", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date checkTime;  
  
 /\*\* 维保单位 \*/  
 @Excel(name = "维保单位")  
 private Long maintenanceId;  
  
 /\*\* 状态（0正常，1停用，2其他状态，例如：安装中） \*/  
 @Excel(name = "状态", readConverterExp = "0=正常，1停用，2其他状态，例如：安装中")  
 private String status;  
  
 public void setVesselId(Long vesselId)   
 {  
 this.vesselId = vesselId;  
 }  
  
 public Long getVesselId()   
 {  
 return vesselId;  
 }  
 public void setDeptId(Long deptId)   
 {  
 this.deptId = deptId;  
 }  
  
 public Long getDeptId()   
 {  
 return deptId;  
 }  
 public void setArea(String area)   
 {  
 this.area = area;  
 }  
  
 public String getArea()   
 {  
 return area;  
 }  
 public void setAddress(String address)   
 {  
 this.address = address;  
 }  
  
 public String getAddress()   
 {  
 return address;  
 }  
 public void setAdministrator(String administrator)   
 {  
 this.administrator = administrator;  
 }  
  
 public String getCode() { return code; }  
 public void setCode(String code)  
 {  
 this.code = code;  
 }  
  
 public String getAdministrator()   
 {  
 return administrator;  
 }  
 public void setPhone(String phone)   
 {  
 this.phone = phone;  
 }  
  
 public String getPhone()   
 {  
 return phone;  
 }  
 public void setFactoryNumber(String factoryNumber)   
 {  
 this.factoryNumber = factoryNumber;  
 }  
  
 public String getFactoryNumber()   
 {  
 return factoryNumber;  
 }  
 public void setRegcode(String regcode)   
 {  
 this.regcode = regcode;  
 }  
  
 public String getRegcode()   
 {  
 return regcode;  
 }  
 public void setRegStatus(String regStatus)   
 {  
 this.regStatus = regStatus;  
 }  
  
 public String getRegStatus()   
 {  
 return regStatus;  
 }  
 public void setVesselName(String vesselName)  
 {  
 this.vesselName = vesselName;  
 }  
  
 public String getVesselName()  
 {  
 return vesselName;  
 }  
 public void setBuyTime(Date buyTime)   
 {  
 this.buyTime = buyTime;  
 }  
  
 public Date getBuyTime()   
 {  
 return buyTime;  
 }  
 public void setLicense(String license)   
 {  
 this.license = license;  
 }  
  
 public String getLicense()   
 {  
 return license;  
 }  
 public void setType(String type)   
 {  
 this.type = type;  
 }  
  
 public String getType()   
 {  
 return type;  
 }  
 public void setTypeNumber(String typeNumber)   
 {  
 this.typeNumber = typeNumber;  
 }  
  
 public String getTypeNumber()   
 {  
 return typeNumber;  
 }  
 public void setAttribute(String attribute)   
 {  
 this.attribute = attribute;  
 }  
  
 public String getAttribute()   
 {  
 return attribute;  
 }  
 public void setCheckType(Integer checkType)   
 {  
 this.checkType = checkType;  
 }  
  
 public Integer getCheckType()   
 {  
 return checkType;  
 }  
 public void setCheckTime(Date checkTime)   
 {  
 this.checkTime = checkTime;  
 }  
  
 public Date getCheckTime()   
 {  
 return checkTime;  
 }  
 public void setMaintenanceId(Long maintenanceId)   
 {  
 this.maintenanceId = maintenanceId;  
 }  
  
 public Long getMaintenanceId()   
 {  
 return maintenanceId;  
 }  
 public void setStatus(String status)   
 {  
 this.status = status;  
 }  
  
 public String getStatus()   
 {  
 return status;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("vesselId", getVesselId())  
 .append("deptId", getDeptId())  
 .append("area", getArea())  
 .append("address", getAddress())  
 .append("code", getCode())  
 .append("administrator", getAdministrator())  
 .append("phone", getPhone())  
 .append("factoryNumber", getFactoryNumber())  
 .append("regcode", getRegcode())  
 .append("regStatus", getRegStatus())  
 .append("vesselName", getVesselName())  
 .append("buyTime", getBuyTime())  
 .append("license", getLicense())  
 .append("type", getType())  
 .append("typeNumber", getTypeNumber())  
 .append("attribute", getAttribute())  
 .append("checkType", getCheckType())  
 .append("checkTime", getCheckTime())  
 .append("maintenanceId", getMaintenanceId())  
 .append("status", getStatus())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .toString();  
 }  
}

===============================EquipmentCheck.java==============================

package com.ruoyi.equipment.domain;  
  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
  
/\*\*  
 \* 检验类型管理对象 equipment\_check  
 \*   
 \* @author cqq  
 \* @date 2023-09-12  
 \*/  
public class EquipmentCheck extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 序列ID \*/  
 private Integer checkId;  
  
 /\*\* 对应年份 \*/  
 @Excel(name = "对应年份")  
 private Integer checkYear;  
  
 /\*\* 检验类型 \*/  
 @Excel(name = "检验类型")  
 private Integer type;  
  
 public void setCheckId(Integer checkId)   
 {  
 this.checkId = checkId;  
 }  
  
 public Integer getCheckId()   
 {  
 return checkId;  
 }  
 public void setCheckYear(Integer checkYear)   
 {  
 this.checkYear = checkYear;  
 }  
  
 public Integer getCheckYear()   
 {  
 return checkYear;  
 }  
 public void setType(Integer type)   
 {  
 this.type = type;  
 }  
  
 public Integer getType()   
 {  
 return type;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("checkId", getCheckId())  
 .append("checkYear", getCheckYear())  
 .append("type", getType())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .toString();  
 }  
}

=====================================Cost.java===================================

package com.ruoyi.image.domain;  
  
import com.fasterxml.jackson.annotation.JsonFormat;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
  
import java.util.Date;  
  
/\*\*  
 \* 报销凭证管理对象 cost  
 \*   
 \* @author dzh  
 \* @date 2024-04-14  
 \*/  
public class Cost extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 开销id \*/  
 @Excel(name = "开销id")  
 private Long costId;  
  
 /\*\* 用户名 \*/  
 @Excel(name = "用户名")  
 private String user;  
  
 /\*\* 开销类型 \*/  
 @Excel(name = "开销类型")  
 private String costType;  
  
 /\*\* 图片路径 \*/  
 @Excel(name = "图片路径")  
 private String url;  
  
 /\*\* 凭证时间 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "凭证时间", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date checkTime;  
  
 /\*\* 用户id \*/  
 private Long userId;  
  
 public void setCostId(Long costId)   
 {  
 this.costId = costId;  
 }  
  
 public Long getCostId()   
 {  
 return costId;  
 }  
 public void setUser(String user)   
 {  
 this.user = user;  
 }  
  
 public String getUser()   
 {  
 return user;  
 }  
 public void setCostType(String costType)   
 {  
 this.costType = costType;  
 }  
  
 public String getCostType()   
 {  
 return costType;  
 }  
 public void setUrl(String url)   
 {  
 this.url = url;  
 }  
  
 public String getUrl()   
 {  
 return url;  
 }  
 public void setCheckTime(Date checkTime)   
 {  
 this.checkTime = checkTime;  
 }  
  
 public Date getCheckTime()   
 {  
 return checkTime;  
 }  
 public void setUserId(Long userId)   
 {  
 this.userId = userId;  
 }  
  
 public Long getUserId()   
 {  
 return userId;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("costId", getCostId())  
 .append("user", getUser())  
 .append("costType", getCostType())  
 .append("url", getUrl())  
 .append("checkTime", getCheckTime())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .append("userId", getUserId())  
 .toString();  
 }  
}

====================================Image.java==================================

package com.ruoyi.image.domain;  
  
import com.fasterxml.jackson.annotation.JsonFormat;  
import com.ruoyi.common.annotation.Excel;  
import com.ruoyi.common.core.domain.BaseEntity;  
import org.apache.commons.lang3.builder.ToStringBuilder;  
import org.apache.commons.lang3.builder.ToStringStyle;  
  
import java.util.Date;  
  
/\*\*  
 \* 维修凭证管理对象 image  
 \*   
 \* @author dzh  
 \* @date 2024-04-16  
 \*/  
public class Image extends BaseEntity  
{  
 private static final long serialVersionUID = 1L;  
  
 /\*\* 日志ID \*/  
 private Long logId;  
  
 /\*\* 用户名 \*/  
 @Excel(name = "用户名")  
 private String user;  
  
 /\*\* 设备类型0电梯1起重机2压力容器 \*/  
 @Excel(name = "设备类型0电梯1起重机2压力容器")  
 private Long equipmentType;  
  
 /\*\* 位置 \*/  
 @Excel(name = "位置")  
 private Long location;  
  
 /\*\* 图片位置 \*/  
 @Excel(name = "图片位置")  
 private String url;  
  
 /\*\* 登记时间 \*/  
 @JsonFormat(pattern = "yyyy-MM-dd")  
 @Excel(name = "登记时间", width = 30, dateFormat = "yyyy-MM-dd")  
 private Date checkTime;  
  
 /\*\* 用户id \*/  
 @Excel(name = "用户id")  
 private Long userId;  
  
 public void setLogId(Long logId)   
 {  
 this.logId = logId;  
 }  
  
 public Long getLogId()   
 {  
 return logId;  
 }  
 public void setUser(String user)   
 {  
 this.user = user;  
 }  
  
 public String getUser()   
 {  
 return user;  
 }  
 public void setEquipmentType(Long equipmentType)   
 {  
 this.equipmentType = equipmentType;  
 }  
  
 public Long getEquipmentType()   
 {  
 return equipmentType;  
 }  
 public void setLocation(Long location)   
 {  
 this.location = location;  
 }  
  
 public Long getLocation()   
 {  
 return location;  
 }  
 public void setUrl(String url)   
 {  
 this.url = url;  
 }  
  
 public String getUrl()   
 {  
 return url;  
 }  
 public void setCheckTime(Date checkTime)   
 {  
 this.checkTime = checkTime;  
 }  
  
 public Date getCheckTime()   
 {  
 return checkTime;  
 }  
 public void setUserId(Long userId)   
 {  
 this.userId = userId;  
 }  
  
 public Long getUserId()   
 {  
 return userId;  
 }  
  
 @Override  
 public String toString() {  
 return new ToStringBuilder(this,ToStringStyle.MULTI\_LINE\_STYLE)  
 .append("logId", getLogId())  
 .append("user", getUser())  
 .append("equipmentType", getEquipmentType())  
 .append("location", getLocation())  
 .append("url", getUrl())  
 .append("checkTime", getCheckTime())  
 .append("createBy", getCreateBy())  
 .append("createTime", getCreateTime())  
 .append("updateBy", getUpdateBy())  
 .append("updateTime", getUpdateTime())  
 .append("remark", getRemark())  
 .append("userId", getUserId())  
 .toString();  
 }  
}

================================SecurityConfig.java===============================

package com.ruoyi.framework.config;  
  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.context.annotation.Bean;  
import org.springframework.http.HttpMethod;  
import org.springframework.security.authentication.AuthenticationManager;  
import org.springframework.security.config.annotation.authentication.builders.AuthenticationManagerBuilder;  
import org.springframework.security.config.annotation.method.configuration.EnableGlobalMethodSecurity;  
import org.springframework.security.config.annotation.web.builders.HttpSecurity;  
import org.springframework.security.config.annotation.web.configuration.WebSecurityConfigurerAdapter;  
import org.springframework.security.config.annotation.web.configurers.ExpressionUrlAuthorizationConfigurer;  
import org.springframework.security.config.http.SessionCreationPolicy;  
import org.springframework.security.core.userdetails.UserDetailsService;  
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;  
import org.springframework.security.web.authentication.UsernamePasswordAuthenticationFilter;  
import org.springframework.security.web.authentication.logout.LogoutFilter;  
import org.springframework.web.filter.CorsFilter;  
import com.ruoyi.framework.config.properties.PermitAllUrlProperties;  
import com.ruoyi.framework.security.filter.JwtAuthenticationTokenFilter;  
import com.ruoyi.framework.security.handle.AuthenticationEntryPointImpl;  
import com.ruoyi.framework.security.handle.LogoutSuccessHandlerImpl;  
  
/\*\*  
 \* spring security配置  
 \*   
 \* @author cqq  
 \*/  
@EnableGlobalMethodSecurity(prePostEnabled = true, securedEnabled = true)  
public class SecurityConfig extends WebSecurityConfigurerAdapter  
{  
 /\*\*  
 \* 自定义用户认证逻辑  
 \*/  
 @Autowired  
 private UserDetailsService userDetailsService;  
   
 /\*\*  
 \* 认证失败处理类  
 \*/  
 @Autowired  
 private AuthenticationEntryPointImpl unauthorizedHandler;  
  
 /\*\*  
 \* 退出处理类  
 \*/  
 @Autowired  
 private LogoutSuccessHandlerImpl logoutSuccessHandler;  
  
 /\*\*  
 \* token认证过滤器  
 \*/  
 @Autowired  
 private JwtAuthenticationTokenFilter authenticationTokenFilter;  
   
 /\*\*  
 \* 跨域过滤器  
 \*/  
 @Autowired  
 private CorsFilter corsFilter;  
  
 /\*\*  
 \* 允许匿名访问的地址  
 \*/  
 @Autowired  
 private PermitAllUrlProperties permitAllUrl;  
  
 /\*\*  
 \* 解决 无法直接注入 AuthenticationManager  
 \*  
 \* @return  
 \* @throws Exception  
 \*/  
 @Bean  
 @Override  
 public AuthenticationManager authenticationManagerBean() throws Exception  
 {  
 return super.authenticationManagerBean();  
 }  
  
 /\*\*  
 \* anyRequest | 匹配所有请求路径  
 \* access | SpringEl表达式结果为true时可以访问  
 \* anonymous | 匿名可以访问  
 \* denyAll | 用户不能访问  
 \* fullyAuthenticated | 用户完全认证可以访问（非remember-me下自动登录）  
 \* hasAnyAuthority | 如果有参数，参数表示权限，则其中任何一个权限可以访问  
 \* hasAnyRole | 如果有参数，参数表示角色，则其中任何一个角色可以访问  
 \* hasAuthority | 如果有参数，参数表示权限，则其权限可以访问  
 \* hasIpAddress | 如果有参数，参数表示IP地址，如果用户IP和参数匹配，则可以访问  
 \* hasRole | 如果有参数，参数表示角色，则其角色可以访问  
 \* permitAll | 用户可以任意访问  
 \* rememberMe | 允许通过remember-me登录的用户访问  
 \* authenticated | 用户登录后可访问  
 \*/  
 @Override  
 protected void configure(HttpSecurity httpSecurity) throws Exception  
 {  
 // 注解标记允许匿名访问的url  
 ExpressionUrlAuthorizationConfigurer<HttpSecurity>.ExpressionInterceptUrlRegistry registry = httpSecurity.authorizeRequests();  
 permitAllUrl.getUrls().forEach(url -> registry.antMatchers(url).permitAll());  
  
 httpSecurity  
 // CSRF禁用，因为不使用session  
 .csrf().disable()  
 // 禁用HTTP响应标头  
 .headers().cacheControl().disable().and()  
 // 认证失败处理类  
 .exceptionHandling().authenticationEntryPoint(unauthorizedHandler).and()  
 // 基于token，所以不需要session  
 .sessionManagement().sessionCreationPolicy(SessionCreationPolicy.STATELESS).and()  
 // 过滤请求  
 .authorizeRequests()  
 // 对于登录login 注册register 验证码captchaImage 允许匿名访问  
 .antMatchers("/login", "/register", "/captchaImage", "/captchaData").permitAll()  
 // 静态资源，可匿名访问  
 .antMatchers(HttpMethod.GET, "/", "/\*.html", "/\*\*/\*.html", "/\*\*/\*.css", "/\*\*/\*.js", "/profile/\*\*").permitAll()  
 .antMatchers("/swagger-ui.html", "/swagger-resources/\*\*", "/webjars/\*\*", "/\*/api-docs", "/druid/\*\*").permitAll()  
 // 除上面外的所有请求全部需要鉴权认证  
 .anyRequest().authenticated()  
 .and()  
 .headers().frameOptions().disable();  
 // 添加Logout filter  
 httpSecurity.logout().logoutUrl("/logout").logoutSuccessHandler(logoutSuccessHandler);  
 // 添加JWT filter  
 httpSecurity.addFilterBefore(authenticationTokenFilter, UsernamePasswordAuthenticationFilter.class);  
 // 添加CORS filter  
 httpSecurity.addFilterBefore(corsFilter, JwtAuthenticationTokenFilter.class);  
 httpSecurity.addFilterBefore(corsFilter, LogoutFilter.class);  
 }  
  
 /\*\*  
 \* 强散列哈希加密实现  
 \*/  
 @Bean  
 public BCryptPasswordEncoder bCryptPasswordEncoder()  
 {  
 return new BCryptPasswordEncoder();  
 }  
  
 /\*\*  
 \* 身份认证接口  
 \*/  
 @Override  
 protected void configure(AuthenticationManagerBuilder auth) throws Exception  
 {  
 auth.userDetailsService(userDetailsService).passwordEncoder(bCryptPasswordEncoder());  
 }  
}

=================================SwaggerConfig.java==============================

package com.ruoyi.web.core.config;  
  
import java.util.ArrayList;  
import java.util.List;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.beans.factory.annotation.Value;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import com.ruoyi.common.config.RuoYiConfig;  
import io.swagger.annotations.ApiOperation;  
import io.swagger.models.auth.In;  
import springfox.documentation.builders.ApiInfoBuilder;  
import springfox.documentation.builders.PathSelectors;  
import springfox.documentation.builders.RequestHandlerSelectors;  
import springfox.documentation.service.ApiInfo;  
import springfox.documentation.service.ApiKey;  
import springfox.documentation.service.AuthorizationScope;  
import springfox.documentation.service.Contact;  
import springfox.documentation.service.SecurityReference;  
import springfox.documentation.service.SecurityScheme;  
import springfox.documentation.spi.DocumentationType;  
import springfox.documentation.spi.service.contexts.SecurityContext;  
import springfox.documentation.spring.web.plugins.Docket;  
  
/\*\*  
 \* Swagger2的接口配置  
 \*   
 \* @author cqq  
 \*/  
@Configuration  
public class SwaggerConfig  
{  
 /\*\* 系统基础配置 \*/  
 @Autowired  
 private RuoYiConfig ruoyiConfig;  
  
 /\*\* 是否开启swagger \*/  
 @Value("${swagger.enabled}")  
 private boolean enabled;  
  
 /\*\* 设置请求的统一前缀 \*/  
 @Value("${swagger.pathMapping}")  
 private String pathMapping;  
  
 /\*\*  
 \* 创建API  
 \*/  
 @Bean  
 public Docket createRestApi()  
 {  
 return new Docket(DocumentationType.OAS\_30)  
 // 是否启用Swagger  
 .enable(enabled)  
 // 用来创建该API的基本信息，展示在文档的页面中（自定义展示的信息）  
 .apiInfo(apiInfo())  
 // 设置哪些接口暴露给Swagger展示  
 .select()  
 // 扫描所有有注解的api，用这种方式更灵活  
 .apis(RequestHandlerSelectors.withMethodAnnotation(ApiOperation.class))  
 // 扫描指定包中的swagger注解  
 // .apis(RequestHandlerSelectors.basePackage("com.ruoyi.project.tool.swagger"))  
 // 扫描所有 .apis(RequestHandlerSelectors.any())  
 .paths(PathSelectors.any())  
 .build()  
 /\* 设置安全模式，swagger可以设置访问token \*/  
 .securitySchemes(securitySchemes())  
 .securityContexts(securityContexts())  
 .pathMapping(pathMapping);  
 }  
  
 /\*\*  
 \* 安全模式，这里指定token通过Authorization头请求头传递  
 \*/  
 private List<SecurityScheme> securitySchemes()  
 {  
 List<SecurityScheme> apiKeyList = new ArrayList<SecurityScheme>();  
 apiKeyList.add(new ApiKey("Authorization", "Authorization", In.HEADER.toValue()));  
 return apiKeyList;  
 }  
  
 /\*\*  
 \* 安全上下文  
 \*/  
 private List<SecurityContext> securityContexts()  
 {  
 List<SecurityContext> securityContexts = new ArrayList<>();  
 securityContexts.add(  
 SecurityContext.builder()  
 .securityReferences(defaultAuth())  
 .operationSelector(o -> o.requestMappingPattern().matches("/.\*"))  
 .build());  
 return securityContexts;  
 }  
  
 /\*\*  
 \* 默认的安全上引用  
 \*/  
 private List<SecurityReference> defaultAuth()  
 {  
 AuthorizationScope authorizationScope = new AuthorizationScope("global", "accessEverything");  
 AuthorizationScope[] authorizationScopes = new AuthorizationScope[1];  
 authorizationScopes[0] = authorizationScope;  
 List<SecurityReference> securityReferences = new ArrayList<>();  
 securityReferences.add(new SecurityReference("Authorization", authorizationScopes));  
 return securityReferences;  
 }  
  
 /\*\*  
 \* 添加摘要信息  
 \*/  
 private ApiInfo apiInfo()  
 {  
 // 用ApiInfoBuilder进行定制  
 return new ApiInfoBuilder()  
 // 设置标题  
 .title("标题：特种设备管理系统\_接口文档")  
 // 描述  
 .description("描述：用于管理集团旗下公司的人员信息,具体包括XXX,XXX模块...")  
 // 作者信息  
 .contact(new Contact(ruoyiConfig.getName(), null, null))  
 // 版本  
 .version("版本号:" + ruoyiConfig.getVersion())  
 .build();  
 }  
}

================================ValidateCodeUtils.java=============================

package com.ruoyi.equipment.util;  
  
import java.util.Random;  
  
/\*\*  
 \* 随机生成验证码工具类  
 \*/  
public class ValidateCodeUtils {  
 /\*\*  
 \* 随机生成验证码  
 \* @param length 长度为4位或者6位  
 \* @return  
 \*/  
 public static Integer generateValidateCode(int length){  
 Integer code =null;  
 if(length == 4){  
 code = new Random().nextInt(9999);//生成随机数，最大为9999  
 if(code < 1000){  
 code = code + 1000;//保证随机数为4位数字  
 }  
 }else if(length == 6){  
 code = new Random().nextInt(999999);//生成随机数，最大为999999  
 if(code < 100000){  
 code = code + 100000;//保证随机数为6位数字  
 }  
 }else{  
 throw new RuntimeException("只能生成4位或6位数字验证码");  
 }  
 return code;  
 }  
  
 /\*\*  
 \* 随机生成指定长度字符串验证码  
 \* @param length 长度  
 \* @return  
 \*/  
 public static String generateValidateCode4String(int length){  
 Random rdm = new Random();  
 String hash1 = Integer.toHexString(rdm.nextInt());  
 String capstr = hash1.substring(0, length);  
 return capstr;  
 }  
}

=================================RedisConfig.java=================================

package com.ruoyi.framework.config;  
  
import org.springframework.cache.annotation.CachingConfigurerSupport;  
import org.springframework.cache.annotation.EnableCaching;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.data.redis.connection.RedisConnectionFactory;  
import org.springframework.data.redis.core.RedisTemplate;  
import org.springframework.data.redis.core.script.DefaultRedisScript;  
import org.springframework.data.redis.serializer.StringRedisSerializer;  
  
/\*\*  
 \* redis配置  
 \*   
 \* @author   
 \*/  
@Configuration  
@EnableCaching  
public class RedisConfig extends CachingConfigurerSupport  
{  
 @Bean  
 @SuppressWarnings(value = { "unchecked", "rawtypes" })  
 public RedisTemplate<Object, Object> redisTemplate(RedisConnectionFactory connectionFactory)  
 {  
 RedisTemplate<Object, Object> template = new RedisTemplate<>();  
 template.setConnectionFactory(connectionFactory);  
  
 FastJson2JsonRedisSerializer serializer = new FastJson2JsonRedisSerializer(Object.class);  
  
 // 使用StringRedisSerializer来序列化和反序列化redis的key值  
 template.setKeySerializer(new StringRedisSerializer());  
 template.setValueSerializer(serializer);  
  
 // Hash的key也采用StringRedisSerializer的序列化方式  
 template.setHashKeySerializer(new StringRedisSerializer());  
 template.setHashValueSerializer(serializer);  
  
 template.afterPropertiesSet();  
 return template;  
 }  
  
 @Bean  
 public DefaultRedisScript<Long> limitScript()  
 {  
 DefaultRedisScript<Long> redisScript = new DefaultRedisScript<>();  
 redisScript.setScriptText(limitScriptText());  
 redisScript.setResultType(Long.class);  
 return redisScript;  
 }  
  
 /\*\*  
 \* 限流脚本  
 \*/  
 private String limitScriptText()  
 {  
 return "local key = KEYS[1]\n" +  
 "local count = tonumber(ARGV[1])\n" +  
 "local time = tonumber(ARGV[2])\n" +  
 "local current = redis.call('get', key);\n" +  
 "if current and tonumber(current) > count then\n" +  
 " return tonumber(current);\n" +  
 "end\n" +  
 "current = redis.call('incr', key)\n" +  
 "if tonumber(current) == 1 then\n" +  
 " redis.call('expire', key, time)\n" +  
 "end\n" +  
 "return tonumber(current);";  
 }  
}

=============================ApplicationConfig.java================================

package com.ruoyi.framework.config;  
  
import java.util.TimeZone;  
import org.mybatis.spring.annotation.MapperScan;  
import org.springframework.boot.autoconfigure.jackson.Jackson2ObjectMapperBuilderCustomizer;  
import org.springframework.context.annotation.Bean;  
import org.springframework.context.annotation.Configuration;  
import org.springframework.context.annotation.EnableAspectJAutoProxy;  
  
/\*\*  
 \* 程序注解配置  
 \*  
 \* @author  
 \*/  
@Configuration  
// 表示通过aop框架暴露该代理对象,AopContext能够访问  
@EnableAspectJAutoProxy(exposeProxy = true)  
// 指定要扫描的Mapper类的包的路径  
@MapperScan("com.ruoyi.\*\*.mapper")  
public class ApplicationConfig  
{  
 /\*\*  
 \* 时区配置  
 \*/  
 @Bean  
 public Jackson2ObjectMapperBuilderCustomizer jacksonObjectMapperCustomization()  
 {  
 return jacksonObjectMapperBuilder -> jacksonObjectMapperBuilder.timeZone(TimeZone.getDefault());  
 }  
}

=====================================END======================================