石化产品碳足迹计算系统V1.0源代码

北京杰迅鸿翔信息技术有限公司

### Java:

package com.jxhs.web.controller.system;  
  
import java.io.IOException;  
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.RequestBody;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
import org.springframework.web.multipart.MultipartFile;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.config.RuoYiConfig;  
import com.jxhs.common.constant.UserConstants;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.entity.SysUser;  
import com.jxhs.common.core.domain.model.LoginUser;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.SecurityUtils;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.common.utils.file.FileUploadUtils;  
import com.jxhs.framework.web.service.TokenService;  
import com.jxhs.system.service.ISysUserService;  
  
*/\*\*  
 \* 个人信息 业务处理  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/system/user/profile")  
public class SysProfileController extends BaseController  
{  
 @Autowired  
 private ISysUserService userService;  
  
 @Autowired  
 private TokenService tokenService;  
  
 */\*\*  
 \* 个人信息  
 \*/* @GetMapping  
 public AjaxResult profile()  
 {  
 LoginUser loginUser = getLoginUser();  
 SysUser user = loginUser.getUser();  
 AjaxResult ajax = AjaxResult.*success*(user);  
 ajax.put("roleGroup", userService.selectUserRoleGroup(loginUser.getUsername()));  
 ajax.put("postGroup", userService.selectUserPostGroup(loginUser.getUsername()));  
 return ajax;  
 }  
  
 */\*\*  
 \* 修改用户  
 \*/* @Log(title = "个人信息", businessType = BusinessType.*UPDATE*)  
 @PutMapping  
 public AjaxResult updateProfile(@RequestBody SysUser user)  
 {  
 LoginUser loginUser = getLoginUser();  
 SysUser sysUser = loginUser.getUser();  
 user.setUserName(sysUser.getUserName());  
 if (StringUtils.*isNotEmpty*(user.getPhonenumber())  
 && UserConstants.*NOT\_UNIQUE*.equals(userService.checkPhoneUnique(user)))  
 {  
 return AjaxResult.*error*("修改用户'" + user.getUserName() + "'失败，手机号码已存在");  
 }  
 if (StringUtils.*isNotEmpty*(user.getEmail())  
 && UserConstants.*NOT\_UNIQUE*.equals(userService.checkEmailUnique(user)))  
 {  
 return AjaxResult.*error*("修改用户'" + user.getUserName() + "'失败，邮箱账号已存在");  
 }  
 user.setUserId(sysUser.getUserId());  
 user.setPassword(null);  
 if (userService.updateUserProfile(user) > 0)  
 {  
 *// 更新缓存用户信息* sysUser.setNickName(user.getNickName());  
 sysUser.setPhonenumber(user.getPhonenumber());  
 sysUser.setEmail(user.getEmail());  
 sysUser.setSex(user.getSex());  
 tokenService.setLoginUser(loginUser);  
 return AjaxResult.*success*();  
 }  
 return AjaxResult.*error*("修改个人信息异常，请联系管理员");  
 }  
  
 */\*\*  
 \* 重置密码  
 \*/* @Log(title = "个人信息", businessType = BusinessType.*UPDATE*)  
 @PutMapping("/updatePwd")  
 public AjaxResult updatePwd(String oldPassword, String newPassword)  
 {  
 LoginUser loginUser = getLoginUser();  
 String userName = loginUser.getUsername();  
 String password = loginUser.getPassword();  
 if (!SecurityUtils.*matchesPassword*(oldPassword, password))  
 {  
 return AjaxResult.*error*("修改密码失败，旧密码错误");  
 }  
 if (SecurityUtils.*matchesPassword*(newPassword, password))  
 {  
 return AjaxResult.*error*("新密码不能与旧密码相同");  
 }  
 if (userService.resetUserPwd(userName, SecurityUtils.*encryptPassword*(newPassword)) > 0)  
 {  
 *// 更新缓存用户密码* loginUser.getUser().setPassword(SecurityUtils.*encryptPassword*(newPassword));  
 tokenService.setLoginUser(loginUser);  
 return AjaxResult.*success*();  
 }  
 return AjaxResult.*error*("修改密码异常，请联系管理员");  
 }  
  
 */\*\*  
 \* 头像上传  
 \*/* @Log(title = "用户头像", businessType = BusinessType.*UPDATE*)  
 @PostMapping("/avatar")  
 public AjaxResult avatar(@RequestParam("avatarfile") MultipartFile file) throws IOException  
 {  
 if (!file.isEmpty())  
 {  
 LoginUser loginUser = getLoginUser();  
 String avatar = FileUploadUtils.*upload*(RuoYiConfig.*getAvatarPath*(), file);  
 if (userService.updateUserAvatar(loginUser.getUsername(), avatar))  
 {  
 AjaxResult ajax = AjaxResult.*success*();  
 ajax.put("imgUrl", avatar);  
 *// 更新缓存用户头像* loginUser.getUser().setAvatar(avatar);  
 tokenService.setLoginUser(loginUser);  
 return ajax;  
 }  
 }  
 return AjaxResult.*error*("上传图片异常，请联系管理员");  
 }  
}

package com.jxhs.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.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.model.RegisterBody;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.framework.web.service.SysRegisterService;  
import com.jxhs.system.service.ISysConfigService;  
  
*/\*\*  
 \* 注册验证  
 \*   
 \* @author ruoyi  
 \*/*@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);  
 }  
}

package com.jxhs.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.jxhs.common.annotation.Log;  
import com.jxhs.common.constant.UserConstants;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.entity.SysRole;  
import com.jxhs.common.core.domain.entity.SysUser;  
import com.jxhs.common.core.domain.model.LoginUser;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.framework.web.service.SysPermissionService;  
import com.jxhs.framework.web.service.TokenService;  
import com.jxhs.system.domain.SysUserRole;  
import com.jxhs.system.service.ISysRoleService;  
import com.jxhs.system.service.ISysUserService;  
  
*/\*\*  
 \* 角色信息  
 \*   
 \* @author ruoyi  
 \*/*@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;  
  
 @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 AjaxResult.*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 (UserConstants.*NOT\_UNIQUE*.equals(roleService.checkRoleNameUnique(role)))  
 {  
 return AjaxResult.*error*("新增角色'" + role.getRoleName() + "'失败，角色名称已存在");  
 }  
 else if (UserConstants.*NOT\_UNIQUE*.equals(roleService.checkRoleKeyUnique(role)))  
 {  
 return AjaxResult.*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 (UserConstants.*NOT\_UNIQUE*.equals(roleService.checkRoleNameUnique(role)))  
 {  
 return AjaxResult.*error*("修改角色'" + role.getRoleName() + "'失败，角色名称已存在");  
 }  
 else if (UserConstants.*NOT\_UNIQUE*.equals(roleService.checkRoleKeyUnique(role)))  
 {  
 return AjaxResult.*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 AjaxResult.*success*();  
 }  
 return AjaxResult.*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 AjaxResult.*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));  
 }  
}

package com.jxhs.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.jxhs.common.annotation.Log;  
import com.jxhs.common.constant.UserConstants;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.entity.SysRole;  
import com.jxhs.common.core.domain.entity.SysUser;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.SecurityUtils;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.system.service.ISysPostService;  
import com.jxhs.system.service.ISysRoleService;  
import com.jxhs.system.service.ISysUserService;  
  
*/\*\*  
 \* 用户信息  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/system/user")  
public class SysUserController extends BaseController  
{  
 @Autowired  
 private ISysUserService userService;  
  
 @Autowired  
 private ISysRoleService roleService;  
  
 @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 AjaxResult.*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 (UserConstants.NOT\_UNIQUE.equals(userService.checkUserNameUnique(user.getUserName())))  
 {  
 return AjaxResult.error("新增用户'" + user.getUserName() + "'失败，登录账号已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getPhonenumber())  
 && UserConstants.NOT\_UNIQUE.equals(userService.checkPhoneUnique(user)))  
 {  
 return AjaxResult.error("新增用户'" + user.getUserName() + "'失败，手机号码已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getEmail())  
 && UserConstants.NOT\_UNIQUE.equals(userService.checkEmailUnique(user)))  
 {  
 return AjaxResult.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 (StringUtils.isNotEmpty(user.getPhonenumber())  
 && UserConstants.NOT\_UNIQUE.equals(userService.checkPhoneUnique(user)))  
 {  
 return AjaxResult.error("修改用户'" + user.getUserName() + "'失败，手机号码已存在");  
 }  
 else if (StringUtils.isNotEmpty(user.getEmail())  
 && UserConstants.NOT\_UNIQUE.equals(userService.checkEmailUnique(user)))  
 {  
 return AjaxResult.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();  
 }  
}

package com.jxhs.web.controller.tool;  
  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.stereotype.Controller;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import com.jxhs.common.core.controller.BaseController;  
  
*/\*\*  
 \* swagger 接口  
 \*   
 \* @author ruoyi  
 \*/*@Controller  
@RequestMapping("/tool/swagger")  
public class SwaggerController extends BaseController  
{  
 @PreAuthorize("@ss.hasPermi('tool:swagger:view')")  
 @GetMapping()  
 public String index()  
 {  
 return redirect("/swagger-ui.html");  
 }  
}

package com.jxhs.web.controller.tool;  
  
import java.util.ArrayList;  
import java.util.LinkedHashMap;  
import java.util.List;  
import java.util.Map;  
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.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.utils.StringUtils;  
import io.swagger.annotations.Api;  
import io.swagger.annotations.ApiImplicitParam;  
import io.swagger.annotations.ApiImplicitParams;  
import io.swagger.annotations.ApiModel;  
import io.swagger.annotations.ApiModelProperty;  
import io.swagger.annotations.ApiOperation;  
  
*/\*\*  
 \* swagger 用户测试方法  
 \*   
 \* @author ruoyi  
 \*/*@Api("用户信息管理")  
@RestController  
@RequestMapping("/test/user")  
public class TestController extends BaseController  
{  
 private final static Map<Integer, UserEntity> *users* = new LinkedHashMap<Integer, UserEntity>();  
 {  
 *users*.put(1, new UserEntity(1, "admin", "admin123", "15888888888"));  
 *users*.put(2, new UserEntity(2, "ry", "admin123", "15666666666"));  
 }  
  
 @ApiOperation("获取用户列表")  
 @GetMapping("/list")  
 public AjaxResult userList()  
 {  
 List<UserEntity> userList = new ArrayList<UserEntity>(*users*.values());  
 return AjaxResult.*success*(userList);  
 }  
  
 @ApiOperation("获取用户详细")  
 @ApiImplicitParam(name = "userId", value = "用户ID", required = true, dataType = "int", paramType = "path", dataTypeClass = Integer.class)  
 @GetMapping("/{userId}")  
 public AjaxResult getUser(@PathVariable Integer userId)  
 {  
 if (!*users*.isEmpty() && *users*.containsKey(userId))  
 {  
 return AjaxResult.*success*(*users*.get(userId));  
 }  
 else  
 {  
 return error("用户不存在");  
 }  
 }  
  
 @ApiOperation("新增用户")  
 @ApiImplicitParams({  
 @ApiImplicitParam(name = "userId", value = "用户id", dataType = "Integer", dataTypeClass = Integer.class),  
 @ApiImplicitParam(name = "username", value = "用户名称", dataType = "String", dataTypeClass = String.class),  
 @ApiImplicitParam(name = "password", value = "用户密码", dataType = "String", dataTypeClass = String.class),  
 @ApiImplicitParam(name = "mobile", value = "用户手机", dataType = "String", dataTypeClass = String.class)  
 })  
 @PostMapping("/save")  
 public AjaxResult save(UserEntity user)  
 {  
 if (StringUtils.*isNull*(user) || StringUtils.*isNull*(user.getUserId()))  
 {  
 return error("用户ID不能为空");  
 }  
 return AjaxResult.*success*(*users*.put(user.getUserId(), user));  
 }  
  
 @ApiOperation("更新用户")  
 @PutMapping("/update")  
 public AjaxResult update(@RequestBody UserEntity user)  
 {  
 if (StringUtils.*isNull*(user) || StringUtils.*isNull*(user.getUserId()))  
 {  
 return error("用户ID不能为空");  
 }  
 if (*users*.isEmpty() || !*users*.containsKey(user.getUserId()))  
 {  
 return error("用户不存在");  
 }  
 *users*.remove(user.getUserId());  
 return AjaxResult.*success*(*users*.put(user.getUserId(), user));  
 }  
  
 @ApiOperation("删除用户信息")  
 @ApiImplicitParam(name = "userId", value = "用户ID", required = true, dataType = "int", paramType = "path", dataTypeClass = Integer.class)  
 @DeleteMapping("/{userId}")  
 public AjaxResult delete(@PathVariable Integer userId)  
 {  
 if (!*users*.isEmpty() && *users*.containsKey(userId))  
 {  
 *users*.remove(userId);  
 return success();  
 }  
 else  
 {  
 return error("用户不存在");  
 }  
 }  
}  
  
@ApiModel(value = "UserEntity", description = "用户实体")  
class UserEntity  
{  
 @ApiModelProperty("用户ID")  
 private Integer userId;  
  
 @ApiModelProperty("用户名称")  
 private String username;  
  
 @ApiModelProperty("用户密码")  
 private String password;  
  
 @ApiModelProperty("用户手机")  
 private String mobile;  
  
 public UserEntity()  
 {  
  
 }  
  
 public UserEntity(Integer userId, String username, String password, String mobile)  
 {  
 this.userId = userId;  
 this.username = username;  
 this.password = password;  
 this.mobile = mobile;  
 }  
  
 public Integer getUserId()  
 {  
 return userId;  
 }  
  
 public void setUserId(Integer userId)  
 {  
 this.userId = userId;  
 }  
  
 public String getUsername()  
 {  
 return username;  
 }  
  
 public void setUsername(String username)  
 {  
 this.username = username;  
 }  
  
 public String getPassword()  
 {  
 return password;  
 }  
  
 public void setPassword(String password)  
 {  
 this.password = password;  
 }  
  
 public String getMobile()  
 {  
 return mobile;  
 }  
  
 public void setMobile(String mobile)  
 {  
 this.mobile = mobile;  
 }  
}

package com.jxhs.web.controller.monitor;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
import java.util.Map;  
import java.util.Properties;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.data.redis.core.RedisCallback;  
import org.springframework.data.redis.core.RedisTemplate;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.utils.StringUtils;  
  
*/\*\*  
 \* 缓存监控  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/monitor/cache")  
public class CacheController  
{  
 @Autowired  
 private RedisTemplate<String, String> redisTemplate;  
  
 @PreAuthorize("@ss.hasPermi('monitor:cache:list')")  
 @GetMapping()  
 public AjaxResult getInfo() throws Exception  
 {  
 Properties info = (Properties) redisTemplate.execute((RedisCallback<Object>) connection -> connection.info());  
 Properties commandStats = (Properties) redisTemplate.execute((RedisCallback<Object>) connection -> connection.info("commandstats"));  
 Object dbSize = redisTemplate.execute((RedisCallback<Object>) connection -> connection.dbSize());  
  
 Map<String, Object> result = new HashMap<>(3);  
 result.put("info", info);  
 result.put("dbSize", dbSize);  
  
 List<Map<String, String>> pieList = new ArrayList<>();  
 commandStats.stringPropertyNames().forEach(key -> {  
 Map<String, String> data = new HashMap<>(2);  
 String property = commandStats.getProperty(key);  
 data.put("name", StringUtils.*removeStart*(key, "cmdstat\_"));  
 data.put("value", StringUtils.*substringBetween*(property, "calls=", ",usec"));  
 pieList.add(data);  
 });  
 result.put("commandStats", pieList);  
 return AjaxResult.*success*(result);  
 }  
}

package com.jxhs.web.controller.monitor;  
  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.framework.web.domain.Server;  
  
*/\*\*  
 \* 服务器监控  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/monitor/server")  
public class ServerController  
{  
 @PreAuthorize("@ss.hasPermi('monitor:server:list')")  
 @GetMapping()  
 public AjaxResult getInfo() throws Exception  
 {  
 Server server = new Server();  
 server.copyTo();  
 return AjaxResult.*success*(server);  
 }  
}

package com.jxhs.web.controller.monitor;  
  
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.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.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.system.domain.SysLogininfor;  
import com.jxhs.system.service.ISysLogininforService;  
  
*/\*\*  
 \* 系统访问记录  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/monitor/logininfor")  
public class SysLogininforController extends BaseController  
{  
 @Autowired  
 private ISysLogininforService logininforService;  
  
 @PreAuthorize("@ss.hasPermi('monitor:logininfor:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysLogininfor logininfor)  
 {  
 startPage();  
 List<SysLogininfor> list = logininforService.selectLogininforList(logininfor);  
 return getDataTable(list);  
 }  
  
 @Log(title = "登录日志", businessType = BusinessType.*EXPORT*)  
 @PreAuthorize("@ss.hasPermi('monitor:logininfor:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysLogininfor logininfor)  
 {  
 List<SysLogininfor> list = logininforService.selectLogininforList(logininfor);  
 ExcelUtil<SysLogininfor> util = new ExcelUtil<SysLogininfor>(SysLogininfor.class);  
 util.exportExcel(response, list, "登录日志");  
 }  
  
 @PreAuthorize("@ss.hasPermi('monitor:logininfor:remove')")  
 @Log(title = "登录日志", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/{infoIds}")  
 public AjaxResult remove(@PathVariable Long[] infoIds)  
 {  
 return toAjax(logininforService.deleteLogininforByIds(infoIds));  
 }  
  
 @PreAuthorize("@ss.hasPermi('monitor:logininfor:remove')")  
 @Log(title = "登录日志", businessType = BusinessType.*CLEAN*)  
 @DeleteMapping("/clean")  
 public AjaxResult clean()  
 {  
 logininforService.cleanLogininfor();  
 return AjaxResult.*success*();  
 }  
}

package com.jxhs.web.controller.monitor;  
  
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.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.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.system.domain.SysOperLog;  
import com.jxhs.system.service.ISysOperLogService;  
  
*/\*\*  
 \* 操作日志记录  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/monitor/operlog")  
public class SysOperlogController extends BaseController  
{  
 @Autowired  
 private ISysOperLogService operLogService;  
  
 @PreAuthorize("@ss.hasPermi('monitor:operlog:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysOperLog operLog)  
 {  
 startPage();  
 List<SysOperLog> list = operLogService.selectOperLogList(operLog);  
 return getDataTable(list);  
 }  
  
 @Log(title = "操作日志", businessType = BusinessType.*EXPORT*)  
 @PreAuthorize("@ss.hasPermi('monitor:operlog:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysOperLog operLog)  
 {  
 List<SysOperLog> list = operLogService.selectOperLogList(operLog);  
 ExcelUtil<SysOperLog> util = new ExcelUtil<SysOperLog>(SysOperLog.class);  
 util.exportExcel(response, list, "操作日志");  
 }  
  
 @Log(title = "操作日志", businessType = BusinessType.*DELETE*)  
 @PreAuthorize("@ss.hasPermi('monitor:operlog:remove')")  
 @DeleteMapping("/{operIds}")  
 public AjaxResult remove(@PathVariable Long[] operIds)  
 {  
 return toAjax(operLogService.deleteOperLogByIds(operIds));  
 }  
  
 @Log(title = "操作日志", businessType = BusinessType.*CLEAN*)  
 @PreAuthorize("@ss.hasPermi('monitor:operlog:remove')")  
 @DeleteMapping("/clean")  
 public AjaxResult clean()  
 {  
 operLogService.cleanOperLog();  
 return AjaxResult.*success*();  
 }  
}

package com.jxhs.web.controller.monitor;  
  
import java.util.ArrayList;  
import java.util.Collection;  
import java.util.Collections;  
import java.util.List;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.security.access.prepost.PreAuthorize;  
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.RequestMapping;  
import org.springframework.web.bind.annotation.RestController;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.constant.Constants;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.model.LoginUser;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.core.redis.RedisCache;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.system.domain.SysUserOnline;  
import com.jxhs.system.service.ISysUserOnlineService;  
  
*/\*\*  
 \* 在线用户监控  
 \*   
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/monitor/online")  
public class SysUserOnlineController extends BaseController  
{  
 @Autowired  
 private ISysUserOnlineService userOnlineService;  
  
 @Autowired  
 private RedisCache redisCache;  
  
 @PreAuthorize("@ss.hasPermi('monitor:online:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(String ipaddr, String userName)  
 {  
 Collection<String> keys = redisCache.keys(Constants.*LOGIN\_TOKEN\_KEY* + "\*");  
 List<SysUserOnline> userOnlineList = new ArrayList<SysUserOnline>();  
 for (String key : keys)  
 {  
 LoginUser user = redisCache.getCacheObject(key);  
 if (StringUtils.*isNotEmpty*(ipaddr) && StringUtils.*isNotEmpty*(userName))  
 {  
 if (StringUtils.*equals*(ipaddr, user.getIpaddr()) && StringUtils.*equals*(userName, user.getUsername()))  
 {  
 userOnlineList.add(userOnlineService.selectOnlineByInfo(ipaddr, userName, user));  
 }  
 }  
 else if (StringUtils.*isNotEmpty*(ipaddr))  
 {  
 if (StringUtils.*equals*(ipaddr, user.getIpaddr()))  
 {  
 userOnlineList.add(userOnlineService.selectOnlineByIpaddr(ipaddr, user));  
 }  
 }  
 else if (StringUtils.*isNotEmpty*(userName) && StringUtils.*isNotNull*(user.getUser()))  
 {  
 if (StringUtils.*equals*(userName, user.getUsername()))  
 {  
 userOnlineList.add(userOnlineService.selectOnlineByUserName(userName, user));  
 }  
 }  
 else  
 {  
 userOnlineList.add(userOnlineService.loginUserToUserOnline(user));  
 }  
 }  
 Collections.*reverse*(userOnlineList);  
 userOnlineList.removeAll(Collections.*singleton*(null));  
 return getDataTable(userOnlineList);  
 }  
  
 */\*\*  
 \* 强退用户  
 \*/* @PreAuthorize("@ss.hasPermi('monitor:online:forceLogout')")  
 @Log(title = "在线用户", businessType = BusinessType.*FORCE*)  
 @DeleteMapping("/{tokenId}")  
 public AjaxResult forceLogout(@PathVariable String tokenId)  
 {  
 redisCache.deleteObject(Constants.*LOGIN\_TOKEN\_KEY* + tokenId);  
 return AjaxResult.*success*();  
 }  
}

package com.jxhs.asm.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.asm.domain.\*;  
import com.jxhs.asm.domain.pricedb.OilCodeReference;  
  
import com.jxhs.asm.domain.pricedb.OilSystemPrice;  
import com.jxhs.asm.service.\*;  
  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.\*;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
  
*/\*\*  
 \* 模型Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-21  
 \*/*@RestController  
@RequestMapping("/asm/model")  
public class AsmModelController extends BaseController {  
  
  
 @Autowired  
 private IAsmModelService asmModelService;  
  
 @Autowired  
 private IAsmModelDeviceService asmModelDeviceService;  
  
 @Autowired  
 private IAsmModelBlendService asmModelBlendService;  
  
 @Autowired  
 private IAsmModelDeviceInputService asmModelDeviceInputService;  
  
 @Autowired  
 private IAsmModelDeviceOutputService asmModelDeviceOutputService;  
  
 @Autowired  
 private IAsmModelDeviceOsblService asmModelDeviceOsblService;  
  
 @Autowired  
 private IAsmModelMaterialBalanceService asmModelMaterialBalanceService;  
  
 @Autowired  
 private IMesModelDataService mesModelDataService;  
  
 */\*\*  
 \* 查询模型列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmModel asmModel) {  
 startPage();  
 List<AsmModel> list = asmModelService.selectAsmModelList(asmModel);  
 return getDataTable(list);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:cMissionCalculation')")  
 @GetMapping("/cMissionCalculation")  
 public ArrayList<DischargeVo> cMissionCalculation(Long modelId) {  
 AsmModel cModel = asmModelService.selectAsmModelByModelId(modelId);  
 ArrayList<DischargeVo> dischargeVos = asmModelService.cmissionCalculation(cModel);  
 return dischargeVos;  
 }  
  
 *//碳足迹分析物* @PreAuthorize("@ss.hasPermi('asm:model:ProductCarbonEmissions')")  
 @GetMapping("/ProductCarbonEmissions")  
 public ArrayList<AsmModelMaterialBalance> ProductCarbonEmissions(Long modelId) {  
 AsmModel cModel = asmModelService.selectAsmModelByModelId(modelId);  
 ArrayList<AsmModelMaterialBalance> asmModelMaterialBalance = asmModelService.selectCo2z(cModel);  
 return asmModelMaterialBalance;  
 }  
  
 */\*\*  
 \* 导出模型列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:export')")  
 @Log(title = "导出模型", businessType = BusinessType.*EXPORT*)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmModel asmModel) {  
 List<AsmModel> list = asmModelService.selectAsmModelList(asmModel);  
 ExcelUtil<AsmModel> util = new ExcelUtil<AsmModel>(AsmModel.class);  
 util.exportExcel(response, list, "模型数据");  
 }  
  
 */\*\*  
 \* 获取模型详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:query')")  
 @GetMapping(value = "/getInfo")  
 public AjaxResult getInfo(Long modelId) {  
 return AjaxResult.*success*(asmModelService.selectAsmModelByModelId(modelId));  
 }  
  
 */\*\*  
 \* 新增模型  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:add')")  
 @Log(title = "新增模型", businessType = BusinessType.*INSERT*)  
 @PostMapping("/add")  
 public AjaxResult add(@RequestBody AsmModel asmModel) {  
 AsmModel asmModel1 = new AsmModel();  
 asmModel1.setModelName(asmModel.getModelName());  
 if (asmModelService.selectAsmModelList(asmModel1).size() > 0) {  
 return AjaxResult.*error*("此装置名称" + asmModel.getModelName() + "已存在");  
 }  
 return toAjax(asmModelService.insertAsmModel(asmModel));  
 }  
  
 */\*\*  
 \* 修改模型  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "修改模型", businessType = BusinessType.*UPDATE*)  
 @PutMapping("/edit")  
 public AjaxResult edit(@RequestBody AsmModel asmModel) {  
 return toAjax(asmModelService.updateAsmModel(asmModel));  
 }  
  
 */\*\*  
 \* 删除模型  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:remove')")  
 @Log(title = "删除模型", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/remove")  
 public AjaxResult remove(Long[] modelIds) {  
 asmModelBlendService.deleteAsmModelBlendByModelIds(modelIds);  
 asmModelDeviceService.deleteAsmModelDeviceByModelIds(modelIds);  
 asmModelDeviceInputService.deleteAsmModelDeviceInputByModelIds(modelIds);  
 asmModelDeviceOutputService.deleteAsmModelDeviceOutputByModelIds(modelIds);  
 asmModelMaterialBalanceService.deleteAsmModelMaterialBalanceByModelIds(modelIds);  
 asmModelDeviceOsblService.deleteAsmModelDeviceOsblByModelIds(modelIds);  
 return toAjax(asmModelService.deleteAsmModelByModelIds(modelIds));  
 }  
  
 */\*\*  
 \* 导出模型列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:exportExcel')")  
 @Log(title = "导出模型", businessType = BusinessType.*EXPORT*)  
 @GetMapping("/exportExcel")  
 public AjaxResult exportExcel(Long modelId, ModelMap mmap, boolean isPrint, Integer printnum) {  
 AsmModel model = asmModelService.selectAsmModelByModelId(modelId);  
 if (model == null) {  
 return AjaxResult.*error*("不存在的模型ID: " + modelId);  
 }  
 System.*out*.println(isPrint);  
*// isPrint=true;* return asmModelService.exportExcel(model, isPrint, printnum);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "计算数据", businessType = BusinessType.*EXPORT*)  
 @GetMapping("/calc")  
 public AjaxResult calc(Long modelId) {  
 AsmModel model = asmModelService.selectAsmModelByModelId(modelId);  
 if (model == null) {  
 return error("不存在的模型ID: " + modelId);  
 }  
 asmModelService.calc(model);  
 return toAjax(true);  
 }  
  
 @PreAuthorize("@ss.hasPermi('co2e:CModel:edit')")  
 @GetMapping("/calcBatch")  
 @Log(title = "模型管理", businessType = BusinessType.*CALC*)  
 public AjaxResult calcBatch(String ids) {  
 asmModelService.calcBatch(ids);  
 return toAjax(true);  
 }  
  
 */\*\*  
 \* 修改模型管理  
 \*/* @GetMapping("/editgraph")  
 @Log(title = "修改模型", businessType = BusinessType.*DELETE*)  
 public AjaxResult edit\_graph(Long modelId, ModelMap mmap) {  
 AsmModel model = asmModelService.selectAsmModelByModelId(modelId);  
 mmap.put("cModel", model);  
 mmap.put("success", true);  
 if (model == null) {  
 mmap.put("success", false);  
 mmap.put("info", "不存在的模型");  
 return AjaxResult.*error*("不存在的模型", mmap);  
 }  
  
 if (model.getDevices() == null || model.getDevices().size() == 0) {  
 mmap.put("success", false);  
 mmap.put("info", "不存在的模型");  
 return AjaxResult.*error*("不存在的模型", mmap);  
 }  
 ModelMap modelMap = asmModelService.edit\_graph(model, mmap);  
 if (modelMap != null) {  
 return AjaxResult.*success*("success", modelMap);  
 }  
 return AjaxResult.*error*("无法建模");  
 }  
  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "获取干气配置", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/gasconfiguration")  
 public AjaxResult gasconfiguration(Long modelId) {  
 AsmModel model = asmModelService.selectgasbymodelId(modelId);  
 return AjaxResult.*success*(model);  
 }  
  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "修改干气配置", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/updategasconfiguration")  
 public AjaxResult updategasconfiguration(Long inputId, String isCalc) {  
 AsmModelDeviceInput asmModelDeviceInput = asmModelDeviceInputService.selectAsmModelDeviceInputByInputId(inputId);  
 asmModelDeviceInput.setIsCalc(isCalc);  
 AsmModelDeviceOutput asmModelDeviceOutput = asmModelDeviceOutputService.selectAsmModelDeviceOutputByOutputId(asmModelDeviceInput.getSuperOutputId());  
 asmModelDeviceOutputService.updateAsmModelDeviceOutput(asmModelDeviceOutput);  
 asmModelDeviceInputService.updateAsmModelDeviceInput(asmModelDeviceInput);  
 return AjaxResult.*success*();  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "存储装置位置", businessType = BusinessType.*UPDATE*)  
 @PostMapping("/updatedeviceposition")  
 public AjaxResult updatedeviceposition(@RequestBody List<AsmModelDevice> asmModelDeviceList) {  
 System.*out*.println(asmModelDeviceList);  
 int a = asmModelDeviceService.updatedeviceposition(asmModelDeviceList);  
 return AjaxResult.*success*();  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "获取万元产值", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/getthousandoutputvalue")  
 public AjaxResult getthousandoutputvalue(AsmModel asmModel) {  
 System.*out*.println(asmModel);  
 return asmModelService.getthousandoutputvalue(asmModel);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:list')")  
 @Log(title = "获取价格体系", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/getPriceSystem")  
 public AjaxResult getPriceSystem() {  
 return asmModelService.getPriceSystem();  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:query')")  
 @Log(title = "获取价格体系库物料编码", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/getfactorylist")  
 public AjaxResult getfactorylist(AsmModel asmModel) {  
*// startPage();* return asmModelService.getfactorylist(asmModel);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:add')")  
 @Log(title = "新增价格体系库物料编码", businessType = BusinessType.*INSERT*)  
 @PostMapping("/addfactorylist")  
 public AjaxResult addfactorylist(@RequestBody List<OilCodeReference> oilCodeReferenceList) {  
 System.*out*.println(oilCodeReferenceList);  
 return asmModelService.addfactorylist(oilCodeReferenceList);  
 }  
  
 *//获取工厂对应的物料价格* @PreAuthorize("@ss.hasPermi('asm:model:query')")  
 @GetMapping("/getfactorypricelist")  
 public AjaxResult getfactorypricelist(AsmModel asmModel) {  
 return asmModelService.getfactorypricelist(asmModel);  
 }  
  
 *//修改工厂对应的物料价格* @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @PostMapping("/updateFactoryPricelist")  
 public AjaxResult updateFactoryPricelist(@RequestBody List<OilSystemPrice> oilSystemPriceList) {  
 return asmModelService.updateFactoryPricelist(oilSystemPriceList);  
*// return AjaxResult.success();* }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:add')")  
 @Log(title = "新增标准物料编码", businessType = BusinessType.*INSERT*)  
 @PostMapping("/addMaterialList")  
 public AjaxResult addMaterialList(@RequestBody List<OilCodeReference> oilCodeReferenceList) {  
 System.*out*.println(oilCodeReferenceList);  
 return asmModelService.addMaterialList(oilCodeReferenceList);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:edit')")  
 @Log(title = "修改产品是否计算碳", businessType = BusinessType.*UPDATE*)  
 @GetMapping("/updateConfiguration")  
 public AjaxResult updateConfiguration(@RequestParam(value = "inputIds", required = false) List<Long> inputIds, @RequestParam(value = "isCalc", required = false) String isCalc) {  
 int a = 0;  
 for (Long inputId : inputIds) {  
 AsmModelDeviceInput asmModelDeviceInput = asmModelDeviceInputService.selectAsmModelDeviceInputByInputId(inputId);  
 asmModelDeviceInput.setIsCalc(isCalc);  
 AsmModelDeviceOutput asmModelDeviceOutput = asmModelDeviceOutputService.selectAsmModelDeviceOutputByOutputId(asmModelDeviceInput.getSuperOutputId());  
 asmModelDeviceOutput.setIsCalc(isCalc);  
 asmModelDeviceOutputService.updateAsmModelDeviceOutput(asmModelDeviceOutput);  
 asmModelDeviceInputService.updateAsmModelDeviceInput(asmModelDeviceInput);  
 a++;  
 }  
 if (a > 0) {  
 return AjaxResult.*success*(a);  
 } else {  
 return AjaxResult.*error*("未修改");  
 }  
 }  
  
 */\*\*  
 \* 导出双碳平台所需模型列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:model:exportExcel')")  
 @Log(title = "导出模型", businessType = BusinessType.*EXPORT*)  
 @GetMapping("/exportCcplantExcel")  
 public AjaxResult exportCcplantExcel(Long modelId) {  
 AsmModel model = asmModelService.selectAsmModelByModelId(modelId);  
 if (model == null) {  
 return AjaxResult.*error*("不存在的模型ID: " + modelId);  
 }  
 return asmModelService.exportCcplantExcel(model);  
 }  
  
 */\*  
 \* 展示调和方案配置  
 \* \*/* @PreAuthorize("@ss.hasPermi('asm:model:query')")  
 @GetMapping("/ShowSolutionConfig")  
 public AjaxResult ShowSolutionConfig(Long modelId) {  
 AsmModel model = asmModelService.selectReconcileDeviceByModelId(modelId);  
*// asmModelService.selectAsmModelByModelId(modelId);* return AjaxResult.*success*(model);  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:model:list')")  
 @GetMapping("/meslist")  
 public AjaxResult meslist() {  
 return asmModelService.selectMesData();  
 }  
  
}

package com.jxhs.asm.service;  
  
import java.util.ArrayList;  
import java.util.List;  
  
import com.jxhs.asm.domain.AsmModel;  
import com.jxhs.asm.domain.AsmModelDeviceOutput;  
import com.jxhs.asm.domain.AsmModelMaterialBalance;  
import com.jxhs.asm.domain.DischargeVo;  
import com.jxhs.asm.domain.pricedb.OilCodeReference;  
import com.jxhs.asm.domain.pricedb.OilMaterial;  
import com.jxhs.asm.domain.pricedb.OilSystemPrice;  
import com.jxhs.common.core.domain.AjaxResult;  
import org.springframework.ui.ModelMap;  
  
*/\*\*  
 \* 模型Service接口  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-21  
 \*/*public interface IAsmModelService  
{  
 */\*\*  
 \* 查询导出所需模型  
 \*  
 \* @param modelId 模型主键  
 \* @return 模型  
 \*/* public AsmModel selectAsmModelByModelId(Long modelId);  
  
 */\*\*  
 \* 查询计算所需模型  
 \*  
 \* @param modelId 模型主键  
 \* @return 模型  
 \*/* public AsmModel selectexportAsmModelByModelId(Long modelId);  
  
 */\*\*  
 \* 查询模型列表  
 \*  
 \* @param asmModel 模型  
 \* @return 模型集合  
 \*/* public List<AsmModel> selectAsmModelList(AsmModel asmModel);  
  
 */\*\*  
 \* 新增模型  
 \*  
 \* @param asmModel 模型  
 \* @return 结果  
 \*/* public int insertAsmModel(AsmModel asmModel);  
  
 */\*\*  
 \* 修改模型  
 \*  
 \* @param asmModel 模型  
 \* @return 结果  
 \*/* public int updateAsmModel(AsmModel asmModel);  
  
 */\*\*  
 \* 批量删除模型  
 \*  
 \* @param modelIds 需要删除的模型主键集合  
 \* @return 结果  
 \*/* public int deleteAsmModelByModelIds(Long[] modelIds);  
  
 */\*\*  
 \* 删除模型信息  
 \*  
 \* @param modelId 模型主键  
 \* @return 结果  
 \*/* public int deleteAsmModelByModelId(Long modelId);  
  
 public AjaxResult exportExcel(AsmModel model,boolean isPrint,Integer printnum);  
  
 public void calc(AsmModel model);  
  
 public ModelMap edit\_graph(AsmModel model, ModelMap mmap);  
  
 ArrayList<DischargeVo> cmissionCalculation(AsmModel cModel);  
  
 ArrayList<AsmModelMaterialBalance> selectCo2z(AsmModel cModel);  
  
 public void calcBatch(String ids);  
  
 */\*  
 \* 查询投入和产出干气的装置  
 \* \*/* public AsmModel selectgasbymodelId(Long modelId);  
  
 public AjaxResult getthousandoutputvalue(AsmModel asmModel);  
  
 public AjaxResult getPriceSystem();  
  
 public AjaxResult getfactorylist(AsmModel asmModel);  
  
 public AjaxResult addfactorylist(List<OilCodeReference> oilCodeReferenceList);  
  
 public AjaxResult getfactorypricelist(AsmModel asmModel);  
  
 public AjaxResult updateFactoryPricelist(List<OilSystemPrice> oilSystemPriceList);  
  
 public AjaxResult addMaterialList(List<OilCodeReference> oilCodeReferenceList);  
  
 public AjaxResult exportCcplantExcel(AsmModel model);  
  
 public AsmModel selectReconcileDeviceByModelId(Long modelId);  
  
 public AjaxResult selectMesData();  
}

package com.jxhs.asm.controller;  
  
import java.io.InputStream;  
import java.util.ArrayList;  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.common.core.domain.entity.SysParent;  
import org.apache.poi.ss.usermodel.Cell;  
import org.apache.poi.ss.usermodel.Row;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.web.bind.annotation.\*;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.asm.domain.AsmCompany;  
import com.jxhs.asm.service.IAsmCompanyService;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
import org.springframework.web.multipart.MultipartFile;  
  
*/\*\*  
 \* 企业字典Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-03-24  
 \*/*@RestController  
@RequestMapping("/asm/companyDict")  
public class AsmCompanyDictController extends BaseController  
{  
 @Autowired  
 private IAsmCompanyService asmCompanyService;  
  
 */\*\*  
 \* 查询企业字典列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmCompany asmCompany)  
 {  
 startPage();  
 List<AsmCompany> list = asmCompanyService.selectAsmCompanyList(asmCompany);  
 return getDataTable(list);  
 }  
  
 */\*\*  
 \* 导入企业字典列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:IMPORT')")  
 @Log(title = "企业字典", businessType = BusinessType.*IMPORT*)  
 @PostMapping("/IMPORT")  
 public AjaxResult fileUpload( @RequestBody MultipartFile file,boolean isEmpty)  
 {  
 */\*List<AsmCompany> list = asmCompanyService.selectAsmCompanyList(asmCompany);  
 ExcelUtil<AsmCompany> util = new ExcelUtil<AsmCompany>(AsmCompany.class);\*/  
 //判断文件是否为空* if (!file.isEmpty()) {  
 if (isEmpty==true){  
 int rows=asmCompanyService.deleteAll();  
 }  
 try {  
 *//拿到文件 把它转成流的形式 因为读取文件是以流的形式读取的，只有把它转变成流的形式，程序才能辨别* String fileName = file.getOriginalFilename();  
 InputStream is = file.getInputStream();  
  
 List<AsmCompany> listAsm = new ArrayList<AsmCompany>();  
 *//调用Excel工具类里读取文件的方法，此方法是以流的形式读取的* List<Row> list = com.jxhs.common.utils.file.ExcelUtil.*getExcelRead*(fileName,is, true);  
 ExcelUtil<AsmCompany> util=new ExcelUtil<AsmCompany>(AsmCompany.class);  
 List<AsmCompany> sysParentList=util.importExcel(file.getInputStream());  
 *//首先是读取行 也就是一行一行读，然后在取到列，遍历行里面的行，根据行得到列的值  
 /\*int i=0;  
 for (Row row : list) {  
 Cell cell\_1 = row.getCell(i);  
 \*//\*Cell cell\_2 = row.getCell(1);  
 Cell cell\_3 = row.getCell(2);\*//\*  
 //得到列的值，也就是你需要解析的字段的值  
 String company\_name = com.jxhs.common.utils.file.ExcelUtil.getValue(cell\_1);  
 \*//\*String company\_abb = com.jxhs.common.utils.file.ExcelUtil.getValue(cell\_2);  
 int company\_order=Integer.valueOf(com.jxhs.common.utils.file.ExcelUtil.getValue(cell\_3));\*//\*  
 //最后把你得到值装进你需要保存的数据库里，也就是其对应的实体类。由于我这边是读取文件里的两个值，加上页面传过来的，一共是五个，你读取的数据需要与你的数据库匹配，也就是你数据库里有什么值，就读取什么  
 AsmCompany asmCompany=new AsmCompany();  
 \*//\*asmCompany.setCompanyAbb(company\_abb);  
 asmCompany.setCompanyOrder((long) company\_order);\*//\*  
 asmCompany.setCompanyName(company\_name);  
 i++;  
 listAsm.add(asmCompany);  
 }\*/  
 //最后做个批量插入，也就是把你读取出来的数据保存到数据库中* return toAjax(asmCompanyService.insertlistAsm(sysParentList));  
 } catch (Exception e) {  
 return AjaxResult.*error*("插入失败");  
 }  
 }else{  
 return AjaxResult.*error*("获取文件失败");  
 }  
 }  
 */\*\*  
 \* 导出企业字典列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:export')")  
 @Log(title = "企业字典", businessType = BusinessType.*EXPORT*)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmCompany asmCompany)  
 {  
 List<AsmCompany> list = asmCompanyService.selectAsmCompanyList(asmCompany);  
 ExcelUtil<AsmCompany> util = new ExcelUtil<AsmCompany>(AsmCompany.class);  
 util.exportExcel(response, list, "企业字典数据");  
 }  
  
 */\*\*  
 \* 获取企业字典详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:query')")  
 @GetMapping( "/getInfo")  
 public AjaxResult getInfo(String companyId)  
 {  
 return AjaxResult.*success*(asmCompanyService.selectAsmCompanyByCompanyId(companyId));  
 }  
  
 */\*\*  
 \* 新增企业字典  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:add')")  
 @Log(title = "企业字典", businessType = BusinessType.*INSERT*)  
 @PostMapping("/add")  
 public AjaxResult add(@RequestBody AsmCompany asmCompany)  
 {  
 return toAjax(asmCompanyService.insertAsmCompany(asmCompany));  
 }  
  
 */\*\*  
 \* 修改企业字典  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:edit')")  
 @Log(title = "企业字典", businessType = BusinessType.*UPDATE*)  
 @PutMapping("/edit")  
 public AjaxResult edit(@RequestBody AsmCompany asmCompany)  
 {  
 return toAjax(asmCompanyService.updateAsmCompany(asmCompany));  
 }  
  
 */\*\*  
 \* 删除企业字典  
 \*/* @PreAuthorize("@ss.hasPermi('asm:companyDict:remove')")  
 @Log(title = "企业字典", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/remove")  
 public AjaxResult remove(String[] companyIds)  
 {  
 return toAjax(asmCompanyService.deleteAsmCompanyByCompanyIds(companyIds));  
 }  
}

package com.jxhs.asm.controller;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.asm.domain.AsmModelBlend;  
import com.jxhs.asm.service.IAsmModelBlendService;  
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.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
  
*/\*\*  
 \* 油品调和Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-21  
 \*/*@RestController  
@RequestMapping("/system/blend")  
public class AsmModelBlendController extends BaseController  
{  
 @Autowired  
 private IAsmModelBlendService asmModelBlendService;  
  
 */\*\*  
 \* 查询油品调和列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmModelBlend asmModelBlend)  
 {  
 startPage();  
 List<AsmModelBlend> list = asmModelBlendService.selectAsmModelBlendList(asmModelBlend);  
 return getDataTable(list);  
 }  
  
 */\*\*  
 \* 导出油品调和列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:export')")  
 @Log(title = "油品调和", businessType = BusinessType.*EXPORT*)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmModelBlend asmModelBlend)  
 {  
 List<AsmModelBlend> list = asmModelBlendService.selectAsmModelBlendList(asmModelBlend);  
 ExcelUtil<AsmModelBlend> util = new ExcelUtil<AsmModelBlend>(AsmModelBlend.class);  
 util.exportExcel(response, list, "油品调和数据");  
 }  
  
 */\*\*  
 \* 获取油品调和详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:query')")  
 @GetMapping(value = "/{blendId}")  
 public AjaxResult getInfo(@PathVariable("blendId") Long blendId)  
 {  
 return AjaxResult.success(asmModelBlendService.selectAsmModelBlendByBlendId(blendId));  
 }  
  
 */\*\*  
 \* 新增油品调和  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:add')")  
 @Log(title = "油品调和", businessType = BusinessType.INSERT)  
 @PostMapping  
 public AjaxResult add(@RequestBody AsmModelBlend asmModelBlend)  
 {  
 return toAjax(asmModelBlendService.insertAsmModelBlend(asmModelBlend));  
 }  
  
 */\*\*  
 \* 修改油品调和  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:edit')")  
 @Log(title = "油品调和", businessType = BusinessType.UPDATE)  
 @PutMapping  
 public AjaxResult edit(@RequestBody AsmModelBlend asmModelBlend)  
 {  
 return toAjax(asmModelBlendService.updateAsmModelBlend(asmModelBlend));  
 }  
  
 */\*\*  
 \* 删除油品调和  
 \*/* @PreAuthorize("@ss.hasPermi('system:blend:remove')")  
 @Log(title = "油品调和", businessType = BusinessType.DELETE)  
 @DeleteMapping("/{blendIds}")  
 public AjaxResult remove(@PathVariable Long[] blendIds)  
 {  
 return toAjax(asmModelBlendService.deleteAsmModelBlendByBlendIds(blendIds));  
 }  
}

package com.jxhs.asm.controller;  
  
import java.util.ArrayList;  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.asm.domain.AsmModel;  
import com.jxhs.asm.domain.AsmModelDevice;  
import com.jxhs.asm.domain.DischargeVo;  
import com.jxhs.asm.service.IAsmModelDeviceService;  
import org.springframework.security.access.prepost.PreAuthorize;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.ui.ModelMap;  
import org.springframework.web.bind.annotation.\*;  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
  
*/\*\*  
 \* 装置Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-21  
 \*/*@RestController  
@RequestMapping("/system/device")  
public class AsmModelDeviceController extends BaseController  
{  
 @Autowired  
 private IAsmModelDeviceService asmModelDeviceService;  
  
 private String prefix = "co2e/CModelDevice";  
 */\*\*  
 \* 查询装置列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmModelDevice asmModelDevice)  
 {  
 startPage();  
 List<AsmModelDevice> list = asmModelDeviceService.selectAsmModelDeviceList(asmModelDevice);  
 return getDataTable(list);  
 }  
  
 */\*\*  
 \* 导出装置列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:export')")  
 @Log(title = "装置", businessType = BusinessType.*EXPORT*)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmModelDevice asmModelDevice)  
 {  
 List<AsmModelDevice> list = asmModelDeviceService.selectAsmModelDeviceList(asmModelDevice);  
 ExcelUtil<AsmModelDevice> util = new ExcelUtil<AsmModelDevice>(AsmModelDevice.class);  
 util.exportExcel(response, list, "装置数据");  
 }  
  
 */\*\*  
 \* 获取装置详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:query')")  
 @GetMapping(value = "/{deviceId}")  
 public AjaxResult getInfo(@PathVariable("deviceId") Long deviceId)  
 {  
 return AjaxResult.*success*(asmModelDeviceService.selectAsmModelDeviceByDeviceId(deviceId));  
 }  
  
 */\*\*  
 \* 新增装置  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:add')")  
 @Log(title = "装置", businessType = BusinessType.*INSERT*)  
 @PostMapping  
 public AjaxResult add(@RequestBody AsmModelDevice asmModelDevice)  
 {  
 return toAjax(asmModelDeviceService.insertAsmModelDevice(asmModelDevice));  
 }  
  
 */\*\*  
 \* 修改装置  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:edit')")  
 @Log(title = "装置", businessType = BusinessType.*UPDATE*)  
 @PutMapping  
 public AjaxResult edit(@RequestBody AsmModelDevice asmModelDevice)  
 {  
 return toAjax(asmModelDeviceService.updateAsmModelDevice(asmModelDevice));  
 }  
  
 */\*\*  
 \* 删除装置  
 \*/* @PreAuthorize("@ss.hasPermi('system:device:remove')")  
 @Log(title = "装置", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/{deviceIds}")  
 public AjaxResult remove(@PathVariable Long[] deviceIds)  
 {  
 return toAjax(asmModelDeviceService.deleteAsmModelDeviceByDeviceIds(deviceIds));  
 }  
  
 @GetMapping("/view/{deviceId}")  
 public AjaxResult view(@PathVariable("deviceId") Long deviceId, ModelMap mmap){  
 AsmModelDevice cModelDevice = asmModelDeviceService.selectAsmModelDeviceByDeviceId(deviceId);  
 mmap.put("cModelDevice", cModelDevice);  
 return AjaxResult.*success*(mmap);  
 }  
}

package com.jxhs.asm.controller;  
  
import java.io.File;  
import java.io.FileInputStream;  
import java.io.InputStream;  
import java.text.DateFormat;  
import java.text.ParseException;  
import java.text.SimpleDateFormat;  
import java.util.\*;  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.asm.domain.\*;  
import com.jxhs.asm.service.IAsmAccessoryDetailService;  
import com.jxhs.asm.service.IAsmCompanyService;  
import com.jxhs.asm.service.IAsmModelDeviceDataService;  
import com.jxhs.asm.service.IAsmShowTableService;  
import com.jxhs.common.constant.UserConstants;  
import com.jxhs.common.core.domain.entity.SysDictData;  
import com.jxhs.common.core.domain.entity.SysUser;  
import com.jxhs.common.core.domain.model.LoginUser;  
import com.jxhs.common.exception.ServiceException;  
import com.jxhs.common.utils.SecurityUtils;  
import com.jxhs.common.utils.StringUtils;  
import com.jxhs.system.service.ISysDictTypeService;  
import com.jxhs.system.service.ISysParentService;  
import com.jxhs.system.service.ISysUserService;  
import lombok.extern.slf4j.Slf4j;  
import org.apache.poi.ss.usermodel.\*;  
import org.apache.poi.xssf.usermodel.XSSFWorkbook;  
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.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
import org.springframework.web.multipart.MultipartFile;  
  
*/\*\*  
 \* 【请填写功能名称】Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-27  
 \*/*@Slf4j  
@RestController  
@RequestMapping("/asm/devicedata")  
public class AsmModelDeviceDataController extends BaseController  
{  
 @Autowired  
 private IAsmModelDeviceDataService asmModelDeviceDataService;  
  
 @Autowired  
 private ISysUserService sysUserService;  
  
 @Autowired  
 private IAsmShowTableService asmShowTableService;  
  
 @Autowired  
 private IAsmCompanyService asmCompanyService;  
  
 @Autowired  
 private ISysParentService sysParentService;  
  
 @Autowired  
 private IAsmAccessoryDetailService asmAccessoryDetailService;  
  
 @Autowired  
 private ISysDictTypeService sysDictTypeService;  
  
 private final String elimination\_capacity="elimination\_capacity";  
 private final String limite\_capacity="limite\_capacity";  
 private final String approval\_compliance="approval\_compliance";  
  
  
*// @PreAuthorize("@ss.hasPermi('asm:devicedata:getuserid')")* @GetMapping("/getuserid")  
 public AjaxResult getuserid(String username)  
 {  
 return AjaxResult.*success*(sysUserService.selectUserByUserName(username));  
 }  
 @GetMapping("/applianceInformation")  
 public ApplianceInformation applianceInformation(){  
 ApplianceInformation applianceInformation = asmModelDeviceDataService.selectapplianceInformation();  
 return applianceInformation;  
 }  
  
  
 */\*\*  
 \* 查询【请填写功能名称】列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmModelDeviceData asmModelDeviceData)  
 {  
 startPage();  
 List<AsmModelDeviceData> asmModelDeviceDataList=asmModelDeviceDataService.selectAsmModelDeviceDataList(asmModelDeviceData);  
 List<SysDictData> eliminationCapacity=sysDictTypeService.selectDictDataByType("elimination\_capacity");  
 List<SysDictData> limiteCapacity=sysDictTypeService.selectDictDataByType("limite\_capacity");  
 List<SysDictData> approvalCompliance=sysDictTypeService.selectDictDataByType("approval\_compliance");  
 for (AsmModelDeviceData asm:asmModelDeviceDataList) {  
 if (asm.getIsExpand()!=null){  
 if (asm.getIsExpand().equals("1")){  
 asm.setIsExpand("是");  
 }else {  
 asm.setIsExpand("否");  
 }  
 }  
 if (asm.getIsClean()!=null){  
 if(asm.getIsClean().equals("1")){  
 asm.setIsClean("是");  
 }else {  
 asm.setIsClean("否");  
 }  
 }  
 if (asm.getIsProject()!=null){  
 if (asm.getIsProject().equals("1")){  
 asm.setIsProject("是");  
 }else {  
 asm.setIsProject("否");  
 }  
 }  
 if (asm.getStatus()!=null){  
 if(asm.getStatus().equals("1")){  
 asm.setStatus("正常运行");  
 }else if (asm.getStatus().equals("2")){  
 asm.setStatus("已改扩建");  
 }else if (asm.getStatus().equals("3")){  
 asm.setStatus("停工");  
 }else if (asm.getStatus().equals("4")){  
 asm.setStatus("已拆除");  
 }else if (asm.getStatus().equals("5")){  
 asm.setStatus("在建");  
 }else if (asm.getStatus().equals("6")){  
 asm.setStatus("未投产");  
 }else if (asm.getStatus().equals("7")){  
 asm.setStatus("拟建");  
 }  
  
 }  
 if (asm.getEliminationCapacity()!=null&&asm.getEliminationCapacity().equals("T")){  
 asm.setEliminationCapacity(eliminationCapacity.get(0).getDictValue());  
 }  
 if (asm.getLimiteCapacity()!=null&&asm.getLimiteCapacity().equals("X")){  
 asm.setLimiteCapacity(limiteCapacity.get(0).getDictValue());  
 }  
 if (asm.getApprovalCompliance()!=null&&asm.getApprovalCompliance().equals("W")){  
 asm.setApprovalCompliance(approvalCompliance.get(0).getDictValue());  
 }  
 }  
  
 return getDataTable(asmModelDeviceDataList);  
 }  
  
 */\*\*  
 \* 导出【请填写功能名称】列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:export')")  
 @Log(title = "数据数据导出", businessType = BusinessType.EXPORT)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmModelDeviceData asmModelDeviceData)  
 {  
 List<AsmModelDeviceData> list = asmModelDeviceDataService.selectAsmModelDeviceDataList(asmModelDeviceData);  
 ExcelUtil<AsmModelDeviceData> util = new ExcelUtil<AsmModelDeviceData>(AsmModelDeviceData.class);  
 util.exportExcel(response, list, "【请填写功能名称】数据");  
 }  
  
 */\*\*  
 \* 获取【请填写功能名称】详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:query')")  
 @GetMapping("/getInfo")  
 public AjaxResult getInfo(Long plantId)  
 {  
 return AjaxResult.success(asmModelDeviceDataService.selectAsmModelDeviceDataByPlantId(plantId));  
 }  
 */\*\*  
 \* 获取省份  
 \*/* @GetMapping("/getstate")  
 public AjaxResult getstate()  
 {  
 return AjaxResult.success(asmCompanyService.selectAllstate());  
 }  
  
 */\*\*  
 \* 获取区域  
 \*/* @GetMapping("/getzone")  
 public AjaxResult getzone()  
 {  
 return AjaxResult.success(asmCompanyService.selectAllzone());  
 }  
  
 */\*\*  
 \* 获取所属公司  
 \*/* @GetMapping("/getparent")  
 public AjaxResult getparent()  
 {  
 return AjaxResult.success(asmModelDeviceDataService.selectAsmModelDeviceDataparent());  
 }  
  
 */\*\*  
 \* 获取企业全称  
 \*/* @GetMapping("/getcompanyName")  
 public AjaxResult getcompanyName()  
 {  
 return AjaxResult.success(asmModelDeviceDataService.selectAsmModelDeviceDataCompanyName());  
 }  
  
 */\*\*  
 \* 获取装置类型  
 \*/* @GetMapping("/gettype")  
 public AjaxResult gettype()  
 {  
 return AjaxResult.success(asmModelDeviceDataService.selectAsmModelDevicetypeparent());  
 }  
 */\*\*  
 \* 新增【请填写功能名称】  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:add')")  
 @Log(title = "模型数据新增", businessType = BusinessType.INSERT)  
 @PostMapping("/add")  
 public AjaxResult add(@RequestBody AsmModelDeviceData asmModelDeviceData)  
 {  
 if (asmModelDeviceData.getApprovalProperty()!=null){  
 if(asmModelDeviceData.getApprovalProperty().equals("新建")){  
 if (asmModelDeviceDataService.checkasmModelDeviceDataUnique(asmModelDeviceData)!=null){  
 return AjaxResult.error("新增装置数据'" + asmModelDeviceData.getDeviceId() + "'失败，装置编号已存在");  
 }  
 }  
  
 }  
  
 return toAjax(asmModelDeviceDataService.insertAsmModelDeviceData(asmModelDeviceData));  
 }  
  
 */\*\*  
 \* 修改【请填写功能名称】  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:edit')")  
 @Log(title = "模型数据修改", businessType = BusinessType.UPDATE)  
 @PutMapping("/edit")  
 public AjaxResult edit(@RequestBody AsmModelDeviceData asmModelDeviceData)  
 {  
 System.out.println(asmModelDeviceData);  
 return toAjax(asmModelDeviceDataService.updateAsmModelDeviceData(asmModelDeviceData));  
 }  
  
 */\*\*  
 \* 删除【请填写功能名称】  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:remove')")  
 @Log(title = "模型数据删除", businessType = BusinessType.DELETE)  
 @DeleteMapping("/remove")  
 public AjaxResult remove(Long[] plantIds)  
 {  
 return toAjax(asmModelDeviceDataService.deleteAsmModelDeviceDataByPlantIds(plantIds));  
 }  
  
 */\*\*  
 \* 删除【请填写功能名称】  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:remove')")  
 @Log(title = "模型数据删除", businessType = BusinessType.DELETE)  
 @DeleteMapping("/removeaccessory")  
 public AjaxResult removeaccessory(Long[] accessoryId)  
 {  
 if (accessoryId!=null&&accessoryId.length>0){  
 return toAjax(asmAccessoryDetailService.deleteAsmAccessoryDetailByAccessoryIds(accessoryId));  
 }  
 return toAjax(false);  
 }  
 */\*\*  
 \* 导入企业字典列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:devicedata:IMPORT')")  
 @Log(title = "企业字典", businessType = BusinessType.IMPORT)  
 @PostMapping("/IMPORT")  
 public AjaxResult fileUpload(@RequestBody MultipartFile file, boolean isEmpty) {  
 if (!file.isEmpty()) {  
 if (isEmpty==true){  
 int rows=asmModelDeviceDataService.deleteAll();  
 }  
 try {  
  
 *//拿到文件 把它转成流的形式 因为读取文件是以流的形式读取的，只有把它转变成流的形式，程序才能辨别* String fileName = file.getOriginalFilename();  
 InputStream is = file.getInputStream();  
 Workbook workbook = new XSSFWorkbook(is);  
 */\*List<AsmModelDeviceData> listAsm = new ArrayList<AsmModelDeviceData>();  
 //调用Excel工具类里读取文件的方法，此方法是以流的形式读取的  
 List<Row> list = com.jxhs.common.utils.file.ExcelUtil.getExcelRead(fileName,is, true);  
 ExcelUtil<AsmModelDeviceData> util=new ExcelUtil<AsmModelDeviceData>(AsmModelDeviceData.class);  
 List<AsmModelDeviceData> asmModelDeviceDataList=util.importExcel(file.getInputStream());  
 return toAjax(asmModelDeviceDataService.insertlistAsm(sysParentList));\*/* List<AsmModelDeviceData> asmModelDeviceDataList=asmModelDeviceDataService.importAsmData(workbook);  
 } catch (Exception e) {  
 return AjaxResult.error("插入失败"+e.getMessage());  
 }  
 }else{  
 return AjaxResult.error("获取文件失败");  
 }  
 return AjaxResult.success("success");  
 }  
  
 private List<AsmModelDeviceData> importAsmData(Workbook workbook) {  
  
 List<AsmModelDeviceData> asmModelDeviceDataList = new ArrayList<>();  
 Sheet sheet = workbook.getSheetAt(0);  
 if (sheet == null) {  
 *// log.warn("处理全厂物料平衡表失败，不存在全厂物料平衡表");* return asmModelDeviceDataList;  
 }  
 AsmModelDeviceData asmModelDeviceData = new AsmModelDeviceData();  
 for (int i = 2; i <= sheet.getLastRowNum(); i++) {  
 Row row = sheet.getRow(i);  
 Cell propeterCell = row.getCell(0);  
 if (propeterCell == null) {  
 *// log.warn("处理得到最后一行： " + i);* break;  
 }  
 String deviceid1 = String.valueOf(propeterCell.getNumericCellValue());  
 String deviceid=deviceid1.substring(0,deviceid1.length()-2);  
 if (StringUtils.isEmpty(deviceid)) {  
 *//throw new ServiceException("处理全厂物料平衡表失败，无法读取装置编码。行号： " + i);* log.warn("处理得到最后一行： " + i);  
 break;  
 }  
 String companystate = row.getCell(1).getStringCellValue();  
 String companyzone = row.getCell(2).getStringCellValue();  
 String company\_parent = row.getCell(3).getStringCellValue();  
 String company\_name = row.getCell(4).getStringCellValue();  
 String format = "yyyy-MM-dd";  
 String company\_establish\_date = String.valueOf(row.getCell(5).getDateCellValue());  
 Date companyestablishdate = new Date();  
 if (company\_establish\_date.length() <= 5) {  
 try {  
 companyestablishdate = new SimpleDateFormat(format).parse(company\_establish\_date + "-12-31");  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 } else {  
*// companyestablishdate=new SimpleDateFormat(format)* }  
 String company\_address = row.getCell(6).getStringCellValue();  
 String ownership = row.getCell(7).getStringCellValue();  
 String device\_name = row.getCell(8).getStringCellValue();  
 String device\_type = row.getCell(9).getStringCellValue();  
 String approval\_date = String.valueOf(row.getCell(10).getDateCellValue());  
 Date approvaldate = new Date();  
 if (approval\_date.length() <= 5) {  
 try {  
 approvaldate = new SimpleDateFormat(format).parse(approval\_date + "-12-31");  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 } else {  
*// companyestablishdate=new SimpleDateFormat(format)* }  
 String approval\_department = row.getCell(11).getStringCellValue();  
 String approval\_details = row.getCell(12).getStringCellValue();  
 String approval\_property = row.getCell(13).getStringCellValue();  
 String start\_time = String.valueOf(row.getCell(16).getDateCellValue());  
 Date startime = new Date();  
 if (start\_time.length() <= 5) {  
 try {  
 startime = new SimpleDateFormat(format).parse(start\_time + "-12-31");  
 } catch (ParseException e) {  
 e.printStackTrace();  
 }  
 } else {  
*// companyestablishdate=new SimpleDateFormat(format)* }  
 String is\_expand = row.getCell(17).getStringCellValue();  
 String is\_project = row.getCell(18).getStringCellValue();  
 String is\_clean = row.getCell(19).getStringCellValue();  
 String status = row.getCell(20).getStringCellValue();  
 String remark = row.getCell(21).getStringCellValue();  
  
 Cell approval\_scale = row.getCell(14);  
 double approvalScale = 0.0;  
 if (approval\_scale.getCellType() == CellType.STRING)  
 approvalScale = Long.parseLong(approval\_scale.getStringCellValue());  
 else if (approval\_scale.getCellType() == CellType.NUMERIC)  
 approvalScale = approval\_scale.getNumericCellValue();  
 else {  
 approvalScale = 0.0;  
 }  
  
 Cell real\_scale = row.getCell(15);  
 double realScale = 0.0;  
 if (real\_scale.getCellType() == CellType.STRING)  
 realScale = Long.parseLong(real\_scale.getStringCellValue());  
 else if (real\_scale.getCellType() == CellType.NUMERIC)  
 realScale = real\_scale.getNumericCellValue();  
 else {  
 realScale = 0.0;  
 }  
  
 asmModelDeviceData.setDeviceId(deviceid);  
 asmModelDeviceData.setCompanyState(companystate);  
 asmModelDeviceData.setCompanyZone(companyzone);  
 asmModelDeviceData.setCompanyParent(company\_parent);  
 asmModelDeviceData.setCompanyName(company\_name);  
 asmModelDeviceData.setCompanyEstablishDate(companyestablishdate);  
 asmModelDeviceData.setCompanyAddress(company\_address);  
 asmModelDeviceData.setOwnership(ownership);  
 asmModelDeviceData.setDeviceName(device\_name);  
 asmModelDeviceData.setDeviceType(device\_type);  
 asmModelDeviceData.setApprovalDate(approvaldate);  
 asmModelDeviceData.setApprovalDepartment(approval\_department);  
 asmModelDeviceData.setApprovalDetails(approval\_details);  
 asmModelDeviceData.setApprovalProperty(approval\_property);  
 asmModelDeviceData.setApprovalScale(approvalScale);  
 asmModelDeviceData.setRealScale(realScale);  
 asmModelDeviceData.setStartTime(startime);  
 asmModelDeviceData.setIsExpand(is\_expand);  
 asmModelDeviceData.setIsProject(is\_project);  
 asmModelDeviceData.setIsClean(is\_clean);  
 asmModelDeviceData.setStatus(status);  
 asmModelDeviceData.setRemark(remark);  
 if (0 == asmModelDeviceDataService.insertAsmModelDeviceData(asmModelDeviceData)) {  
 throw new ServiceException("处理全场数据表失败，无法新增全场数据。行号： " + i);  
 }  
 asmModelDeviceDataList.add(asmModelDeviceData);  
 }  
 return asmModelDeviceDataList;  
 }  
*// @PreAuthorize("@ss.hasPermi('asm:devicedata:isshow')")* @Log(title = "用户自定义展示", businessType = BusinessType.EXPORT)  
 @PostMapping("/isshow")  
 public AjaxResult isshow(@RequestBody AsmShowTable asmShowTable)  
 {  
 if (asmShowTableService.selectAsmShowTableByUserId(Long.parseLong(asmShowTable.getUserId())).size()>0){  
 asmShowTableService.deleteAsmShowTableByUserId(Long.parseLong(asmShowTable.getUserId()));  
 };  
 asmShowTable.setField("detailId");  
 asmShowTable.setIsShow(asmShowTable.getDetailId());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("deviceId");  
 asmShowTable.setIsShow(asmShowTable.getDeviceId());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyState");  
 asmShowTable.setIsShow(asmShowTable.getCompanyState());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyZone");  
 asmShowTable.setIsShow(asmShowTable.getCompanyZone());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyParent");  
 asmShowTable.setIsShow(asmShowTable.getCompanyParent());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyName");  
 asmShowTable.setIsShow(asmShowTable.getCompanyName());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyEstablishDate");  
 asmShowTable.setIsShow(asmShowTable.getCompanyEstablishDate());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("companyAddress");  
 asmShowTable.setIsShow(asmShowTable.getCompanyAddress());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("ownership");  
 asmShowTable.setIsShow(asmShowTable.getOwnership());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("deviceName");  
 asmShowTable.setIsShow(asmShowTable.getDeviceName());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("deviceType");  
 asmShowTable.setIsShow(asmShowTable.getDeviceType());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalDate");  
 asmShowTable.setIsShow(asmShowTable.getApprovalDate());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalDepartment");  
 asmShowTable.setIsShow(asmShowTable.getApprovalDepartment());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalDepartmentLevel");  
 asmShowTable.setIsShow(asmShowTable.getApprovalDepartmentLevel());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalDetails");  
 asmShowTable.setIsShow(asmShowTable.getApprovalDetails());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalProperty");  
 asmShowTable.setIsShow(asmShowTable.getApprovalProperty());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalScale");  
 asmShowTable.setIsShow(asmShowTable.getApprovalScale());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("realScale");  
 asmShowTable.setIsShow(asmShowTable.getRealScale());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("startTime");  
 asmShowTable.setIsShow(asmShowTable.getStartTime());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("isExpand");  
 asmShowTable.setIsShow(asmShowTable.getIsExpand());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("expandDetail");  
 asmShowTable.setIsShow(asmShowTable.getExpandDetail());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("isProject");  
 asmShowTable.setIsShow(asmShowTable.getIsProject());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("isClean");  
 asmShowTable.setIsShow(asmShowTable.getIsClean());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("status");  
 asmShowTable.setIsShow(asmShowTable.getStatus());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("remark");  
 asmShowTable.setIsShow(asmShowTable.getRemark());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("eliminationCapacity");  
 asmShowTable.setIsShow(asmShowTable.getEliminationCapacity());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("limiteCapacity");  
 asmShowTable.setIsShow(asmShowTable.getLimiteCapacity());  
 asmShowTableService.insertAsmShowTable(asmShowTable);  
 asmShowTable.setField("approvalCompliance");  
 asmShowTable.setIsShow(asmShowTable.getApprovalCompliance());  
 if (asmShowTableService.insertAsmShowTable(asmShowTable)>0){  
 return AjaxResult.success("添加成功");  
 }  
 return AjaxResult.error("添加失败");  
 }  
  
  
 @GetMapping("/getshow")  
 public AjaxResult getshow (Long userId){  
 return AjaxResult.success(asmShowTableService.selectAsmShowTableByUserId(userId));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getareaeli")  
 public AjaxResult getareaeli()  
 {  
 String EliminationCapacity="T";  
 return AjaxResult.success(asmModelDeviceDataService.selectareaeli(EliminationCapacity));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getstateeli")  
 public AjaxResult getstateeli()  
 {  
 String EliminationCapacity="T";  
 return AjaxResult.success(asmModelDeviceDataService.getstateeli(EliminationCapacity));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getareaast")  
 public AjaxResult getareaast()  
 {  
 String limite="X";  
 return AjaxResult.success(asmModelDeviceDataService.selectareaast(limite));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getstateast")  
 public AjaxResult getstateast()  
 {  
 String limite="X";  
 return AjaxResult.success(asmModelDeviceDataService.getstateast(limite));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getareaout")  
 public AjaxResult getareaout()  
 {  
 String approval="W";  
 return AjaxResult.success(asmModelDeviceDataService.selectareaout(approval));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getstateout")  
 public AjaxResult getstateout()  
 {  
 String approval="W";  
 return AjaxResult.success(asmModelDeviceDataService.getstateout(approval));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getdiseli")  
 public AjaxResult getdiseli()  
 {  
 String EliminationCapacity="T";  
 return AjaxResult.success(asmModelDeviceDataService.selectdiseli(EliminationCapacity));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getdisast")  
 public AjaxResult getdisast()  
 {  
 String limite="X";  
 return AjaxResult.success(asmModelDeviceDataService.selectdisast(limite));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getdisout")  
 public AjaxResult getdisout()  
 {  
 String approval="W";  
 return AjaxResult.success(asmModelDeviceDataService.selectdisout(approval));  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getstateatmospheric")  
 public AjaxResult getstateatmospheric()  
 {  
 return AjaxResult.success(asmModelDeviceDataService.getstateatmospheric());  
 }  
  
 @PreAuthorize("@ss.hasPermi('asm:devicedata:list')")  
 @GetMapping("/getareaatmospheric")  
 public AjaxResult getareaatmospheric()  
 {  
 return AjaxResult.success(asmModelDeviceDataService.getareaatmospheric());  
 }  
}

package com.jxhs.asm.controller;  
  
import com.jxhs.common.annotation.Log;  
import com.jxhs.common.constant.UserConstants;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.core.domain.entity.SysParent;  
import com.jxhs.common.core.page.TableDataInfo;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.system.service.ISysParentService;  
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.\*;  
import org.springframework.web.multipart.MultipartFile;  
  
import javax.servlet.http.HttpServletResponse;  
import java.util.List;  
  
*/\*\*  
 \* 数据字典信息  
 \*  
 \* @author ruoyi  
 \*/*@RestController  
@RequestMapping("/asm/dict")  
public class AsmOsblDictController extends BaseController  
{  
 */\*@Autowired  
 private ISysDictTypeService dictTypeService;\*/* @Autowired  
 private ISysParentService sysParentService;  
  
 @PreAuthorize("@ss.hasPermi('asm:dict:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(SysParent sysParent)  
 {  
 startPage();  
 List<SysParent> list = sysParentService.selectParentList(sysParent);  
 return getDataTable(list);  
 }  
  
 @Log(title = "导出字典类型", businessType = BusinessType.*EXPORT*)  
 @PreAuthorize("@ss.hasPermi('asm:dict:export')")  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, SysParent sysParent)  
 {  
 List<SysParent> list = sysParentService.selectParentAll();  
 ExcelUtil<SysParent> util = new ExcelUtil<SysParent>(SysParent.class);  
 util.exportExcel(response, list, "字典类型");  
 }  
  
 */\*\*  
 \* 导入企业字典列表  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:IMPORT')")  
 @Log(title = "导入企业字典", businessType = BusinessType.*IMPORT*)  
 @PostMapping("/IMPORT")  
 public AjaxResult fileUpload(@RequestBody MultipartFile file, boolean isEmpty)  
 {  
 */\*List<AsmCompany> list = asmCompanyService.selectAsmCompanyList(asmCompany);  
 ExcelUtil<AsmCompany> util = new ExcelUtil<AsmCompany>(AsmCompany.class);\*/  
 //判断文件是否为空* if (!file.isEmpty()) {  
 if (isEmpty==true){  
 int rows=sysParentService.deleteAll();  
 }  
 return AjaxResult.*success*(sysParentService.fileUpload(file));  
 }else{  
 return AjaxResult.*error*("获取文件失败");  
 }  
 }  
 */\*\*  
 \* 查询字典类型详细  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:query')")  
 @GetMapping(value = "/getInfo")  
 public AjaxResult getInfo(@RequestParam("dictId") String dictId)  
 {  
 return AjaxResult.*success*(sysParentService.selectSysParentById(Integer.*valueOf*(dictId)));  
 }  
  
 */\*\*  
 \* 新增字典类型  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:add')")  
 @Log(title = "新增字典类型", businessType = BusinessType.*INSERT*)  
 @PostMapping("/add")  
 public AjaxResult add(@RequestBody SysParent sysParent)  
 {  
 if (UserConstants.*NOT\_UNIQUE*.equals(sysParentService.checkSysParentUnique(sysParent)))  
 {  
 return AjaxResult.*error*("新增字典'" + sysParent.getDictName() + "'失败，字典类型已存在");  
 }else if(sysParent.getCo2EmissionCoefficient()==null){  
 sysParent.setCo2EmissionCoefficient(0.0);  
 }else if (sysParent.getCalorificValue()==null){  
 sysParent.setCalorificValue(0.0);  
 }else if (sysParent.getFoldCoefficient()==null){  
 sysParent.setFoldCoefficient(0.0);  
 }else if (sysParent.getParentId()==null){  
 sysParent.setParentId(0);  
 }  
 return toAjax(sysParentService.insertSysParent(sysParent));  
 }  
  
 */\*\*  
 \* 修改字典类型  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:edit')")  
 @Log(title = "修改字典类型", businessType = BusinessType.*UPDATE*)  
 @PutMapping("/edit")  
 public AjaxResult edit(@Validated @RequestBody SysParent parent)  
 {  
 if (UserConstants.*NOT\_UNIQUE*.equals(sysParentService.checkSysParentUnique(parent)))  
 {  
 return AjaxResult.*error*("修改字典'" + parent.getDictName() + "'失败，字典类型已存在");  
 }else if(parent.getCo2EmissionCoefficient()==null){  
 parent.setCo2EmissionCoefficient(0.00);  
 }else if (parent.getCalorificValue()==null){  
 parent.setCalorificValue(0.0);  
 }else if (parent.getFoldCoefficient()==null){  
 parent.setFoldCoefficient(0.0);  
 }else if (parent.getParentId()==null){  
 parent.setParentId(0);  
 }  
 return toAjax(sysParentService.updateSysParent(parent));  
 }  
  
 */\*\*  
 \* 删除字典  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:remove')")  
 @Log(title = "删除字典类型", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/remove")  
 public AjaxResult remove(@RequestParam("dictIds") long[] dictIds){  
 for (long s:dictIds) {  
 int i = Integer.*parseInt*(String.*valueOf*(s));  
 sysParentService.deleteSysParentById(i);  
 }  
  
*// int[] array = Arrays.asList(dictIds).stream().mapToInt(Integer::parseInt).toArray();  
// int i = Integer.parseInt(String.valueOf(dictIds));  
// sysParentService.deleteSysParentByIds(dictIds);* return success();  
 }  
  
 */\*\*  
 \* 刷新字典缓存  
 \*/* @PreAuthorize("@ss.hasPermi('asm:dict:remove')")  
 @Log(title = "刷新字典类型缓存", businessType = BusinessType.*CLEAN*)  
 @DeleteMapping("/refreshCache")  
 public AjaxResult refreshCache()  
 {  
 sysParentService.resetDictCache();  
 return AjaxResult.*success*();  
 }  
  
 */\*\*  
 \* 获取父表选择框列表  
 \*/* @GetMapping("/optionselect")  
 public AjaxResult optionselect()  
 {  
 List<SysParent> sysParents = sysParentService.selectSysParentsAll();  
 return AjaxResult.*success*(sysParents);  
 }  
}

package com.jxhs.asm.controller;  
  
import java.util.List;  
import javax.servlet.http.HttpServletResponse;  
  
import com.jxhs.asm.domain.AsmModelMaterialBalance;  
import com.jxhs.asm.service.IAsmModelMaterialBalanceService;  
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.jxhs.common.annotation.Log;  
import com.jxhs.common.core.controller.BaseController;  
import com.jxhs.common.core.domain.AjaxResult;  
import com.jxhs.common.enums.BusinessType;  
import com.jxhs.common.utils.poi.ExcelUtil;  
import com.jxhs.common.core.page.TableDataInfo;  
  
*/\*\*  
 \* 物料平衡Controller  
 \*  
 \* @author ruoyi  
 \* @date 2022-04-21  
 \*/*@RestController  
@RequestMapping("/system/balance")  
public class AsmModelMaterialBalanceController extends BaseController  
{  
 @Autowired  
 private IAsmModelMaterialBalanceService asmModelMaterialBalanceService;  
  
 */\*\*  
 \* 查询物料平衡列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:list')")  
 @GetMapping("/list")  
 public TableDataInfo list(AsmModelMaterialBalance asmModelMaterialBalance)  
 {  
 startPage();  
 List<AsmModelMaterialBalance> list = asmModelMaterialBalanceService.selectAsmModelMaterialBalanceList(asmModelMaterialBalance);  
 return getDataTable(list);  
 }  
  
 */\*\*  
 \* 导出物料平衡列表  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:export')")  
 @Log(title = "物料平衡", businessType = BusinessType.*EXPORT*)  
 @PostMapping("/export")  
 public void export(HttpServletResponse response, AsmModelMaterialBalance asmModelMaterialBalance)  
 {  
 List<AsmModelMaterialBalance> list = asmModelMaterialBalanceService.selectAsmModelMaterialBalanceList(asmModelMaterialBalance);  
 ExcelUtil<AsmModelMaterialBalance> util = new ExcelUtil<AsmModelMaterialBalance>(AsmModelMaterialBalance.class);  
 util.exportExcel(response, list, "物料平衡数据");  
 }  
  
 */\*\*  
 \* 获取物料平衡详细信息  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:query')")  
 @GetMapping(value = "/{materielId}")  
 public AjaxResult getInfo(@PathVariable("materielId") Long materielId)  
 {  
 return AjaxResult.*success*(asmModelMaterialBalanceService.selectAsmModelMaterialBalanceByMaterielId(materielId));  
 }  
  
 */\*\*  
 \* 新增物料平衡  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:add')")  
 @Log(title = "物料平衡", businessType = BusinessType.*INSERT*)  
 @PostMapping  
 public AjaxResult add(@RequestBody AsmModelMaterialBalance asmModelMaterialBalance)  
 {  
 return toAjax(asmModelMaterialBalanceService.insertAsmModelMaterialBalance(asmModelMaterialBalance));  
 }  
  
 */\*\*  
 \* 修改物料平衡  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:edit')")  
 @Log(title = "物料平衡", businessType = BusinessType.*UPDATE*)  
 @PutMapping  
 public AjaxResult edit(@RequestBody AsmModelMaterialBalance asmModelMaterialBalance)  
 {  
 return toAjax(asmModelMaterialBalanceService.updateAsmModelMaterialBalance(asmModelMaterialBalance));  
 }  
  
 */\*\*  
 \* 删除物料平衡  
 \*/* @PreAuthorize("@ss.hasPermi('system:balance:remove')")  
 @Log(title = "物料平衡", businessType = BusinessType.*DELETE*)  
 @DeleteMapping("/{materielIds}")  
 public AjaxResult remove(@PathVariable Long[] materielIds)  
 {  
 return toAjax(asmModelMaterialBalanceService.deleteAsmModelMaterialBalanceByMaterielIds(materielIds));  
 }  
}

### Vue:

        describes:undefined,

        parentId:undefined

      };

      this.resetForm("form");

    },

    /\*\* 搜索按钮操作 \*/

    handleQuery() {

      this.pageNum = 1;

      this.getList();

    },

    /\*\* 重置按钮操作 \*/

    resetQuery() {

      this.dateRange = [];

      this.resetForm("queryForm");

      this.handleQuery();

    },

    /\*\* 新增按钮操作 \*/

    handleAdd() {

      this.reset();

      this.open = true;

      this.title = "添加字典类型";

    },

    // 多选框选中数据

    handleSelectionChange(selection) {

      this.ids = selection.map(item => item.dictId)

      this.names = selection.map(item => item.dictName)

      this.single = selection.length!=1

      this.multiple = !selection.length

    },

    /\*\* 修改按钮操作 \*/

    handleUpdate(row) {

      this.reset();

      const dictId = row.dictId || this.ids

      getType(dictId).then(response => {

        this.form = response.data;

        this.open = true;

        this.title = "修改能源类型";

      });

    },

    /\*\* 提交按钮 \*/

    submitForm: function () {

      let dictNames = []

      for (let item in this.dictNameList) {

        dictNames.push(this.dictNameList[item].dictName)

      }

      if (this.form.parentId == undefined) {

        this.form.parentId = 0

      }

      this.$refs["form"].validate(valid => {

        if (valid) {

          if (this.form.dictId != undefined) {

            updateType(this.form).then(response => {

              this.$modal.msgSuccess("修改成功");

              this.open = false;

              this.getList();

            });

          } else {

            if (dictNames.indexOf(this.form.dictName) === -1) {

              addType(this.form).then(response => {

                this.$modal.msgSuccess("新增成功");

                this.open = false;

                this.getList();

              });

            } else {

              this.$modal.msgError('能源名称"' + this.form.dictName + '"已存在！')

            }

          }

        }

      });

    },

    /\*\* 删除按钮操作 \*/

    handleDelete(row) {

      const dictIds = row.dictId || this.ids;

      const dictNames=row.dictName||this.names;

      this.$modal.confirm('是否确认删除能源名称为"' + dictNames + '"的数据项？').then(function() {

        return delType(dictIds);

      })

      .then(() => {

        this.getList();

        this.$modal.msgSuccess("删除成功");

      }).catch(() => {});

    },

    /\*\* 导入按钮操作 \*/

    handleUpload() {

      this.openUpload = true;

      this.uploadTitle = "导入excel文件";

    },

    // 确认导入文件

    submitFileForm() {

      this.$refs.upload.submit();

    },

    // 文件上传中处理

    handleFileUploadProgress(event, file, fileList) {

      this.upload.isUploading = true;

      this.fileList=file

    }, // 文件上传成功处理

    handleFileSuccess(res, file, fileList) {

      this.upload.open = false;

      this.upload.isUploading = false;

      this.$refs.upload.clearFiles();

      if (res.code == 200) {

        this.getList();

         this.$modal.msgSuccess("恭喜您，数据已全部导入成功！");

        //  setTimeout(function(){

           this.openUpload=false

        //  },1000)

      }

    },

    /\*\* 导出按钮操作 \*/

    handleExport() {

      this.download('asm/dict/export', {

        ...this.queryParams

      }, `type\_${new Date().getTime()}.xlsx`)

    },

    /\*\* 刷新缓存按钮操作 \*/

    handleRefreshCache() {

      refreshCache().then(() => {

        this.$modal.msgSuccess("刷新成功");

      });

    },

    UpNumber(e) {

      // 输入框中只允许输入小数点和数字，小数点后只能跟一位

      e.target.value = (e.target.value.match(/^\d\*(\.?\d{0,4})/g)[0]) || null

    }

  },

  watch: {

    total(){

      if(this.total===(this.currentPage-1)\*this.pageSize&&this.total!==0){

        this.currentPage=1;

        this.newtypeList.start=(this.currentPage-1)\*this.newtypeList.limit;

        this.getList()

      }

    }

  }

};

</script>

<template>

<div class="app-container">

<el-form size="small" :inline="true" :model="queryParams" ref="queryForm" label-width="68px">

<el-form-item label="模型名称" prop="modelName">

<el-input

v-model="queryParams.modelName"

placeholder="模型名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所属工厂" prop="modelFactory">

<el-select v-model="queryParams.modelFactory" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyName"

:key="item.id"

:label="item.name"

:value="item.id">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属区域" prop="modelRegion">

<el-select v-model="queryParams.modelRegion" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyZone"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属省份" prop="modelProvince">

<el-select v-model="queryParams.modelProvince" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyState"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属城市" prop="modelCity">

<el-select v-model="queryParams.modelCity" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyCity"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属区（县）" prop="modelArea" label-width="100px">

<el-select v-model="queryParams.modelArea" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyArea"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属公司" prop="modelCompany">

<el-select v-model="queryParams.modelCompany" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyParent"

:key="item"

:label="item"

:value="item">

</el-option>

</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="['asm:model:add']"

>新增</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="['asm:model: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="['asm:model:export']"

>导出</el-button

>

</el-col>

</el-row>

<el-table :data="modelList" @selection-change="handleSelectionChange" v-loading="loading">

<el-table-column type="selection" align="center" min-width="4%"></el-table-column>

<el-table-column label="模型名称" align="center" prop="modelName" min-width="6%"></el-table-column>

<el-table-column label="所属工厂" align="center" prop="modelFactory" min-width="12%"></el-table-column>

<el-table-column label="所属区域" align="center" prop="modelRegion" min-width="4%"></el-table-column>

<el-table-column label="所属省份" align="center" prop="modelProvince" min-width="4%"></el-table-column>

<el-table-column label="所属城市" align="center" prop="modelCity" min-width="4%"></el-table-column>

<el-table-column label="所属区/县" align="center" prop="modelArea" min-width="6%"></el-table-column>

<el-table-column label="所属公司" align="center" prop="modelCompany" min-width="6%"></el-table-column>

<el-table-column label="开始时间" align="center" prop="beginTime" min-width="8%">

<template slot-scope="scope">

<span>{{ parseTime(scope.row.beginTime, '{y}-{m}-{d}') }}</span>

</template>

</el-table-column>

<el-table-column label="结束时间" align="center" prop="endTime" min-width="8%">

<template slot-scope="scope">

<span>{{ parseTime(scope.row.endTime, '{y}-{m}-{d}') }}</span>

</template>

</el-table-column>

<el-table-column label="工厂碳排放" align="center" prop="totalFactoryCo2" min-width="7%">

<template slot-scope="scope">

{{scope.row.totalFactoryCo2 | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="产品碳排放" align="center" prop="totalProductCo2" min-width="7%">

<template slot-scope="scope">

{{scope.row.totalProductCo2 | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="总加工量" align="center" prop="totalInput" min-width="6%">

<template slot-scope="scope">

{{scope.row.totalInput | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="总产出量" align="center" prop="totalOutput" min-width="6%">

<template slot-scope="scope">

{{scope.row.totalOutput | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="价格体系" align="center" prop="priceSystem" min-width="6%">

<template slot-scope="scope">

<div class="flex\_center">

<el-button class="flex\_center stateBtn" type="text" v-if="scope.row.priceSystem" @click="openPrice(scope.row)">

{{scope.row.priceSystem}}

</el-button>

<el-button class="flex\_center font12 stateBtn" v-else @click="openPrice(scope.row)">添加+</el-button>

</div>

</template>

</el-table-column>

<el-table-column label="操作" align="center" min-width="10%">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="element-icons el-iconjisuanqi\_o"

@click="handleCalc(scope.row)"

v-hasPermi="['co2e:CModel:edit']"

>计算</el-button

>

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleState(scope.row)"

v-hasPermi="['asm:model:editgraph']"

>编辑</el-button

>

<el-button

size="mini"

type="text"

icon="el-icon-document"

@click="handleExport(scope.row)"

v-hasPermi="['asm:model:exportExcel']"

>导出</el-button

>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:model: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"

/>

<el-dialog title="导出Excel" :visible.sync="outerisShowExport" :close-on-click-modal=false>

<!-- <span>是否直接导出数据：</span>

<el-radio v-model="isPrint" :label="true">直接导出</el-radio>

<el-radio v-model="isPrint" :label="false">指定行数导出</el-radio> -->

<span slot="footer" class="dialog-footer">

<el-button @click="outerisShowExport=false">取消</el-button>

<el-button type="warning" @click="handleExportLineOut(0)">指定行数导出</el-button>

<el-button type="primary" @click="handleExportLineOut(1)">直接导出</el-button>

</span>

<el-dialog

title="选择框"

width="30%"

:append-to-body="true"

:close-on-click-modal=false

:visible.sync="isShowExport">

<!-- <span>是否导出打印格式数据：</span> -->

<!-- <el-radio v-model="isPrint" :label="true">是</el-radio>

<el-radio v-model="isPrint" :label="false">否</el-radio> -->

<el-form label-position="left" label-width="120px">

<el-form-item label="导出Excel行数" prop="numberRows">

<el-input v-model="numberRows" type="number" placeholder="请输入行数"/>

</el-form-item>

</el-form>

<span slot="footer" class="dialog-footer">

<el-button @click="isShowExport = false">取 消</el-button>

<el-button type="primary" @click="handleExportLine">确 定</el-button>

</span>

</el-dialog>

</el-dialog>

<!-- 修改价格体系 -->

<el-dialog title="价格体系" :visible.sync="visiblePrice" width="500px" append-to-body :close-on-click-modal=false>

<el-form :model="form">

<el-form-item label="价格体系">

<el-select v-model="priceSystem" style="width:380px" @change="priceChange">

<el-option

v-for="item in priceSystemArr"

:key="item.priceSystemId"

:label="item.priceSystemName"

:value="item">

</el-option>

</el-select>

</el-form-item>

</el-form>

<div slot="footer" class="dialog-footer">

<el-button @click="visiblePrice = false">取 消</el-button>

<el-button type="primary" @click="priceSubmit">确 定</el-button>

</div>

</el-dialog>

<!-- 添加或修改企业字典对话框 -->

<el-dialog :title="title" :visible.sync="addDialog" width="500px" append-to-body :close-on-click-modal=false>

<el-form label-position="left" ref="form" :model="form" :rules="rules" label-width="80px">

<el-form-item label="模型名称" prop="modelName">

<el-input v-model="form.modelName" placeholder="请输入模型名称" />

</el-form-item>

<el-form-item label="价格体系" prop="priceSystem">

<el-select v-model="priceSystemArr.priceSystemName" placeholder="请选择价格体系" @change="priceChoose" style="width:380px" value-key = 'priceSystemId'>

<el-option

v-for="item in priceSystemArr"

:key="item.priceSystemId"

:label="item.priceSystemName"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属工厂" prop="modelFactory">

<el-select v-model="companyList.companyName" placeholder="请选择工厂" @change="companyChoose" style="width:380px">

<el-option

v-for="item in companyName"

:key="item.id"

:label="item.name"

:value="item.id">

</el-option>

</el-select>

</el-form-item>

<el-row :gutter="20">

<el-col :span="12">

<el-form-item label="所属区域" prop="modelRegion">

<el-input v-model="companyList.companyZone" disabled placeholder="请输入所属公司"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="所属省份" prop="modelProvince">

<el-input v-model="companyList.companyState" disabled placeholder="请输入所属区域"/>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="20">

<el-col :span="12">

<el-form-item label="所属城市" prop="modelCity">

<el-input v-model="companyList.companyCity" disabled placeholder="请输入所属公司"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="所属区/县" prop="modelArea">

<el-input v-model="companyList.companyArea" disabled placeholder="请输入所属区域"/>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="20">

<el-col :span="12">

<el-form-item label="所属公司" prop="modelCompany">

<el-input v-model="companyList.companyParent" disabled placeholder="请输入所属区域"/>

</el-form-item>

</el-col>

</el-row>

<el-row :gutter="20">

<el-col :span="12">

<el-form-item label="开始时间" prop="beginTime">

<el-date-picker

style="width:140px"

v-model="form.beginTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="选择时间">

</el-date-picker>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="结束时间" prop="endTime">

<el-date-picker

style="width:140px"

v-model="form.endTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="选择时间">

</el-date-picker>

</el-form-item>

</el-col>

</el-row>

<el-form-item label="导入文件">

<el-upload

ref="upload"

:limit="1"

accept=".xlsx,.xls"

:headers="upload.headers"

:action="upload.url"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="true"

drag

>

<i class="el-icon-upload"></i>

<div class="el-upload\_\_text" v-if="!form.fileUrl">

将文件拖到此处，或

<em>点击上传</em>

</div>

<div class="el-upload\_\_text" v-else>

文件已上传，

<em>点击重新上传</em>

</div>

<div class="el-upload\_\_tip" style="color: red" slot="tip">

提示：仅允许导入“xls”或“xlsx”格式文件!

</div>

</el-upload>

</el-form-item>

</el-form>

<div slot="footer" class="dialog-footer">

<el-button type="primary" @click="submitFileForm" :loading="disabled">确 定</el-button>

<el-button @click="cancel">取 消</el-button>

</div>

</el-dialog>

<el-dialog :visible.sync="open" :fullscreen="true" :close-on-click-modal=false>

<div id="container" v-loading.fullscreen.lock="modelLoading"></div>

<el-dialog title="搜索装置" :visible.sync="showHighLightDevice" width="500px" append-to-body :close-on-click-modal=false>

<el-form :model="form">

<el-form-item label="装置名称">

<el-select v-model="deviceNameItem.name" clearable filterable allow-create style="width:380px" @change="deviceChange">

<el-option

v-for="item in deviceNameArr"

:key="item.id"

:label="item.name"

:value="item">

</el-option>

</el-select>

</el-form-item>

</el-form>

<div slot="footer" class="dialog-footer">

<el-button @click="showHighLightDevice = false">取 消</el-button>

<el-button type="primary" @click="deviceNameSubmit">确 定</el-button>

</div>

</el-dialog>

<div id="tooltipText" ref="tooltipText" class="tooltiptext"></div>

<div ref="alltipText" class="tooltiptext" v-for="item in nodes"></div>

<div ref="configure" class="configure"></div>

<div class="todoList">

<div @click="toolbarRedo()" class="icon">

<el-tooltip class="item" effect="dark" content="撤销" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="24" height="24">

<path d="M256 682.666667c0-102.741333 66.730667-213.333333 213.333333-213.333334 107.008 0 190.762667 56.576 230.570667 125.354667L611.968 682.666667H853.333333v-241.365334l-91.562666 91.562667C704.768 448.469333 601.130667 384 469.333333 384c-196.096 0-298.666667 150.229333-298.666666 298.666667h85.333333z" fill="" p-id="2041"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarUndo()" class="icon">

<el-tooltip class="item" effect="dark" content="回退" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="24" height="24">

<path d="M170.666667 682.666667h241.365333l-87.936-87.978667C363.904 525.909333 447.658667 469.333333 554.666667 469.333333c146.602667 0 213.333333 110.592 213.333333 213.333334h85.333333c0-148.437333-102.570667-298.666667-298.666666-298.666667-131.797333 0-235.392 64.469333-292.48 148.821333L170.666667 441.301333V682.666667z" fill="" p-id="2764"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarOut()" class="icon">

<el-tooltip class="item" effect="dark" content="放大" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="24" height="24">

<path d="M658.432 428.736a33.216 33.216 0 0 1-33.152 33.152H525.824v99.456a33.216 33.216 0 0 1-66.304 0V461.888H360.064a33.152 33.152 0 0 1 0-66.304H459.52V296.128a33.152 33.152 0 0 1 66.304 0V395.52H625.28c18.24 0 33.152 14.848 33.152 33.152z m299.776 521.792a43.328 43.328 0 0 1-60.864-6.912l-189.248-220.992a362.368 362.368 0 0 1-215.36 70.848 364.8 364.8 0 1 1 364.8-364.736 363.072 363.072 0 0 1-86.912 235.968l192.384 224.64a43.392 43.392 0 0 1-4.8 61.184z m-465.536-223.36a298.816 298.816 0 0 0 298.432-298.432 298.816 298.816 0 0 0-298.432-298.432A298.816 298.816 0 0 0 194.24 428.8a298.816 298.816 0 0 0 298.432 298.432z"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarIn()" class="icon">

<el-tooltip class="item" effect="dark" content="缩小" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="24" height="24">

<path d="M639.936 416a32 32 0 0 1-32 32h-256a32 32 0 0 1 0-64h256a32 32 0 0 1 32 32z m289.28 503.552a41.792 41.792 0 0 1-58.752-6.656l-182.656-213.248A349.76 349.76 0 0 1 480 768 352 352 0 1 1 832 416a350.4 350.4 0 0 1-83.84 227.712l185.664 216.768a41.856 41.856 0 0 1-4.608 59.072zM479.936 704c158.784 0 288-129.216 288-288S638.72 128 479.936 128a288.32 288.32 0 0 0-288 288c0 158.784 129.216 288 288 288z" p-id="3853"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarRealZoom()" class="icon">

<el-tooltip class="item" effect="dark" content="1：1比例" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="20" height="24">

<path d="M384 320v384H320V320h64z m256 0v384H576V320h64zM512 576v64H448V576h64z m0-192v64H448V384h64z m355.968 576H92.032A28.16 28.16 0 0 1 64 931.968V28.032C64 12.608 76.608 0 95.168 0h610.368L896 192v739.968a28.16 28.16 0 0 1-28.032 28.032zM704 64v128h128l-128-128z m128 192h-190.464V64H128v832h704V256z"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarautoZoom()" class="icon">

<el-tooltip class="item" effect="dark" content="总览" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" width="20" height="24">

<path d="M684.288 305.28l0.128-0.64-0.128-0.64V99.712c0-19.84 15.552-35.904 34.496-35.712a35.072 35.072 0 0 1 34.56 35.776v171.008h170.944c19.648 0 35.84 15.488 35.712 34.432a35.072 35.072 0 0 1-35.84 34.496h-204.16l-0.64-0.128a32.768 32.768 0 0 1-20.864-7.552c-1.344-1.024-2.816-1.664-3.968-2.816-0.384-0.32-0.512-0.768-0.832-1.088a33.472 33.472 0 0 1-9.408-22.848zM305.28 64a35.072 35.072 0 0 0-34.56 35.776v171.008H99.776A35.072 35.072 0 0 0 64 305.216c0 18.944 15.872 34.496 35.84 34.496h204.16l0.64-0.128a32.896 32.896 0 0 0 20.864-7.552c1.344-1.024 2.816-1.664 3.904-2.816 0.384-0.32 0.512-0.768 0.768-1.088a33.024 33.024 0 0 0 9.536-22.848l-0.128-0.64 0.128-0.704V99.712A35.008 35.008 0 0 0 305.216 64z m618.944 620.288h-204.16l-0.64 0.128-0.512-0.128c-7.808 0-14.72 3.2-20.48 7.68-1.28 1.024-2.752 1.664-3.84 2.752-0.384 0.32-0.512 0.768-0.832 1.088a33.664 33.664 0 0 0-9.408 22.912l0.128 0.64-0.128 0.704v204.288c0 19.712 15.552 35.904 34.496 35.712a35.072 35.072 0 0 0 34.56-35.776V753.28h170.944c19.648 0 35.84-15.488 35.712-34.432a35.072 35.072 0 0 0-35.84-34.496z m-593.92 11.52c-0.256-0.32-0.384-0.768-0.768-1.088-1.088-1.088-2.56-1.728-3.84-2.688a33.088 33.088 0 0 0-20.48-7.68l-0.512 0.064-0.64-0.128H99.84a35.072 35.072 0 0 0-35.84 34.496 35.072 35.072 0 0 0 35.712 34.432H270.72v171.008c0 19.84 15.552 35.84 34.56 35.776a35.008 35.008 0 0 0 34.432-35.712V720l-0.128-0.64 0.128-0.704a33.344 33.344 0 0 0-9.472-22.848zM512 374.144a137.92 137.92 0 1 0 0.128 275.84A137.92 137.92 0 0 0 512 374.08z"></path>

</svg>

</el-tooltip>

</div>

<div @click="toolbarCal()" class="icon">

<el-tooltip class="item" effect="dark" content="计算" placement="top-start">

<svg t="1660614215510" class="icon" viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" p-id="23348" width="24" height="24"><path d="M762 942H262c-71.68 0-130-58.32-130-130V212c0-71.68 58.32-130 130-130h500c71.68 0 130 58.32 130 130v600c0 71.68-58.32 130-130 130zM262 142c-38.59 0-70 31.41-70 70v600c0 38.59 31.41 70 70 70h500c38.59 0 70-31.41 70-70V212c0-38.59-31.41-70-70-70H262z" p-id="23349"></path><path d="M692 402H332c-38.59 0-70-31.41-70-70v-60c0-38.59 31.41-70 70-70h360c38.59 0 70 31.41 70 70v60c0 38.59-31.41 70-70 70zM332 262c-5.42 0-10 4.58-10 10v60c0 5.42 4.58 10 10 10h360c5.43 0 10-4.58 10-10v-60c0-5.42-4.57-10-10-10H332zM372 542h-60c-16.57 0-30-13.43-30-30s13.43-30 30-30h60c16.57 0 30 13.43 30 30s-13.43 30-30 30zM542 542h-60c-16.57 0-30-13.43-30-30s13.43-30 30-30h60c16.56 0 30 13.43 30 30s-13.44 30-30 30zM712 542h-60c-16.56 0-30-13.43-30-30s13.44-30 30-30h60c16.56 0 30 13.43 30 30s-13.44 30-30 30z" p-id="23350"></path><path d="M372 672h-60c-16.57 0-30-13.44-30-30s13.43-30 30-30h60c16.57 0 30 13.44 30 30s-13.43 30-30 30zM542 672h-60c-16.57 0-30-13.44-30-30s13.43-30 30-30h60c16.56 0 30 13.44 30 30s-13.44 30-30 30zM712 672h-60c-16.56 0-30-13.44-30-30s13.44-30 30-30h60c16.56 0 30 13.44 30 30s-13.44 30-30 30z" p-id="23351"></path><path d="M372 802h-60c-16.57 0-30-13.44-30-30s13.43-30 30-30h60c16.57 0 30 13.44 30 30s-13.43 30-30 30zM542 802h-60c-16.57 0-30-13.44-30-30s13.43-30 30-30h60c16.56 0 30 13.44 30 30s-13.44 30-30 30zM712 802h-60c-16.56 0-30-13.44-30-30s13.44-30 30-30h60c16.56 0 30 13.44 30 30s-13.44 30-30 30z" p-id="23352"></path></svg>

</el-tooltip>

</div>

<div @click="toolbarDialog()" class="icon">

<el-tooltip class="item" effect="dark" content="显示/隐藏浮窗信息" placement="top-start">

<svg t="1662455824725" class="icon" viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" p-id="3863" width="24" height="24"><path d="M737.3824 982.8352H81.8176V327.2704h159.9488v81.92h-78.0288v491.7248h491.7248V740.6592h81.92v242.176zM411.9552 162.0992h82.3296v81.92h-82.3296zM519.68 162.0992h82.3296v81.92h-82.3296z" fill="#000000" p-id="3864"></path><path d="M942.592 696.32H286.72V40.96h655.36zM368.64 614.4h491.52V122.88H368.64z" fill="#000000" p-id="3865"></path><path d="M627.5072 162.0992h82.3296v81.92h-82.3296z" fill="#000000" p-id="3866"></path></svg>

</el-tooltip>

</div>

<div @click="toolbarFindDevice()" class="icon">

<el-tooltip class="item" effect="dark" content="搜索装置名称" placement="top-start">

<svg t="1662455624116" class="icon" viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" p-id="2534" width="24" height="24"><path d="M934.242232 884.637975 738.287281 695.561689l1.836835-2.302439c52.767807-65.338101 81.829703-147.602709 81.829703-231.593636 0-203.716728-165.742858-369.473913-369.411491-369.473913-203.731054 0-369.505635 165.757185-369.505635 369.473913 0 203.715705 165.774581 369.45754 369.505635 369.45754 89.26607 0 175.511339-32.450063 242.841817-91.288123l2.209318-1.929955 195.485253 186.963159 1.959631 0 0.031722 1.61887c4.947685 3.483334 10.517541 5.319146 16.5837 5.319146 16.180518 0 29.308513-13.192464 29.308513-29.310559C940.96126 896.276037 938.72329 890.239554 934.242232 884.637975zM452.541305 772.413008c-171.358763 0-310.74637-139.388631-310.74637-310.74637 0-171.34239 139.387607-310.700321 310.74637-310.700321 171.309644 0 310.700321 139.357931 310.700321 310.700321C763.241626 633.024377 623.851972 772.413008 452.541305 772.413008z" p-id="2535"></path></svg>

</el-tooltip>

</div>

<div @click="toolbarExport()" class="icon">

<el-tooltip class="item" effect="dark" content="导出图片" placement="top-start">

<svg viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" p-id="3757" width="24" height="24">

<path d="M145.6 0c-44.8 0-80 36.8-80 81.6V944c0 44.8 35.2 80 80 80h732.8c44.8 0 81.6-35.2 81.6-80V326.4L657.6 0h-512z" fill="#49C9A7" p-id="3758"></path><path d="M960 326.4v16H755.2s-100.8-20.8-99.2-108.8c0 0 4.8 92.8 97.6 92.8H960z" fill="#37BB91" p-id="3759"></path><path d="M657.6 0v233.6c0 25.6 17.6 92.8 97.6 92.8H960L657.6 0z" fill="#FFFFFF" p-id="3760"></path><path d="M225.6 859.2V524.8H560v334.4H225.6z m300.8-300.8H259.2v201.6h267.2V558.4z m-153.6 134.4l62.4-84.8 20.8 33.6 20.8-6.4 16 89.6H283.2l56-52.8 33.6 20.8z m-60.8-59.2c-14.4 0-27.2-9.6-27.2-24 0-12.8 12.8-24 27.2-24 14.4 0 25.6 11.2 25.6 24 0 14.4-11.2 24-25.6 24z" fill="#FFFFFF" p-id="3761"></path>

</svg>

</el-tooltip>

</div>

</div>

<span slot="footer" class="dialog-footer">

<el-button @click="open = false">关 闭</el-button>

<el-button type="primary" @click="closeInit()">确 定</el-button>

</span>

<el-dialog

width="85%"

title="查看设备"

:visible.sync="innerOpen"

:close-on-click-modal=false

append-to-body>

<el-form label-width="150px" :model="viewList" style="width:70%;margin-left:15%">

<el-form-item label="装置编码：" prop="deviceCode">

<input type="text" v-model="viewList.deviceCode" disabled="disabled">

</el-form-item>

<el-form-item label="装置名称：" >

<input type="text" v-model="viewList.deviceName" disabled="disabled">

</el-form-item>

<el-form-item label="装置加工量：" >

<input type="text" v-model="viewList.deviceProcess" disabled="disabled">

</el-form-item>

<el-form-item label="装置碳足迹：" >

<input type="text" v-model="viewList.deviceCo2" disabled="disabled">

</el-form-item>

<el-form-item label="装置碳强度：">

<input type="text" v-model="viewList.deviceFactor" disabled="disabled">

</el-form-item>

<el-form-item label="虚拟装置：" >

<input type="text" v-model="viewList.isVirtual==0?'否':'是'" disabled="disabled">

</el-form-item>

</el-form>

<span style="color:#6879B6">装置消耗信息</span>

<el-divider></el-divider>

<el-table

v-loading="loading"

:height="280"

:data="viewList.osbls"

>

<el-table-column

type="selection"

width="55">

</el-table-column>

<el-table-column label="序号" align="center" type="index"></el-table-column>

<el-table-column label="公用工程编码" align="center" prop="osblCode"></el-table-column>

<el-table-column label="公用工程名称" align="center" prop="osblName"></el-table-column>

<el-table-column label="CO2排放系统" align="center" prop="co2EmissionCoefficient"></el-table-column>

<el-table-column label="消耗量（万吨）" align="center" prop="consumption" >

<template slot-scope="scope">

<el-input v-model="scope.row.consumption" placeholder="请输入" @blur="updateConsumption(scope.row)" @keyup.native="UpNumber" @keydown.native="UpNumber"></el-input>

</template>

</el-table-column>

<el-table-column label="碳排放量（吨）" align="center" prop="carbonEmission" >

<template slot-scope="scope">

{{scope.row.carbonEmission | numFilter}}

</template>

</el-table-column>

<el-table-column label="方案编码" align="center" prop="planCode"></el-table-column>

<el-table-column label="方案名称" align="center" prop="planName"></el-table-column>

</el-table>

<el-dialog title="" :visible.sync="isCalCo2" append-to-body :close-on-click-modal=false>

<span class="font16">是否重新计算碳：</span>

<div style="margin-left: 10px;display: inline-block;">

<el-radio v-model="isCal" size="medium" :label="1"><span class="font16">是</span></el-radio>

<el-radio v-model="isCal" size="medium" :label="0"><span class="font16">否</span></el-radio>

</div>

<span slot="footer" class="dialog-footer">

<el-button @click="isCalCo2=false">取 消</el-button>

<el-button type="primary" @click="handleSubmitCalCo2">确 定</el-button>

</span>

</el-dialog>

<span slot="footer" class="dialog-footer">

<el-button @click="innerOpen = false">关 闭</el-button>

<el-button type="primary" @click="submitOsbl">确 定</el-button>

</span>

</el-dialog>

<!-- 配置物料是否计算碳 -->

<el-dialog title="配置物料" :visible.sync="isCalcDialog" append-to-body :close-on-click-modal=false>

<span class="font16">物料是否计算碳：</span>

<div style="margin-left: 10px;display: inline-block;">

<el-radio v-model="isCalc" size="medium" :label="1"><span class="font16">是</span></el-radio>

<el-radio v-model="isCalc" size="medium" :label="0"><span class="font16">否</span></el-radio>

</div>

<span slot="footer" class="dialog-footer">

<el-button @click="isCalcDialog=false">取 消</el-button>

<el-button type="primary" @click="handleSubmitCal">确 定</el-button>

</span>

</el-dialog>

</el-dialog>

</div>

</template>

<style>

.tooltipTit{

width:90%;

margin-left: 5%;

height: 35px;

border-bottom: 1px solid #f3f1f1;

color:#000;

}

.toolLi{

height: 20px;

display: flex;

color:#787b8e;

align-items: center;

}

.configure\_item{height: 35px;padding-left: 10px}

.configure\_item:hover{background:#e3e3e3;font-weight: 550;cursor: pointer}

</style>

<script>

import {listModel,addModel,delModel,calcModel,editModel,viewModel,

exportModel,setdevicepos,getPriceSystem,updatePriceSystem,updateConfiguration,

updateOsbl} from "@/api/asm/model";

import {listCompanyDict,getCompanyDict} from "@/api/asm/companyDict";

import { getToken } from "@/utils/auth";

import { Graph,ObjectExt,DataUri } from '@antv/x6'

export default {

name: "Model",

data() {

return {

// 遮罩层

loading: true,

modelLoading:true,

// 选中数组

ids: [],

names:[],

// 非多个禁用

multiple: true,

// 非单个禁用

single: true,

// 模型名称表格数据

modelList:[],

//所属工厂集合数组

companyName:[],

companyZone:[],

companyState:[],

companyCity:[],

companyParent:[],

companyArea:[],

//当前选中所属工厂数据

companyList:[],

//价格体系列表

priceSystemArr:[],

visiblePrice:false,

priceSystem:'',

priceSystemId:'',

//提交修改价格体系

priceSystemData:[],

// 点击确认修改后回显页面上的价格体系

// priceSystem:'',

// 总条数

total: 0,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

modelName:null,

modelFactory:null,

modelRegion:null,

modelProvince:null,

modelCity:null,

modelArea:null,

modelCompany:null,

modelId:null

},

// 是否显示弹出层

addDialog: false,

// 弹出层标题

title:"",

// 表单参数

form: {},

rules: {

modelName: [

{ required: true, message: "模型名称不能为空", trigger: "blur" },

],

},

//配置物料是否计算碳

isCalc:1,

isCalcDialog:false,

inputIdList:[],

// 修改模型管理

open:false,

innerOpen:false,

upload: {

open: false, // 是否显示弹出层（用户导入）

title: "导入",

// 是否禁用上传

isUploading: false,

// 是否更新已经存在的用户数据

updateSupport: 0,

// 设置上传的请求头部

headers: { Authorization: getToken() },

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/common/upload",

},

//导出行数

numberRows:40,

//是否显示导出弹框

isShowExport:false,

outerisShowExport:false,

isPrint:true,

// 导出当前行

modelId:"",

nodeList:[],

//查看设备

viewList:{},

viewModelOsbls:[],

params:{

modelId:null

},

//点击右键配置物料列表

textList:{},

graph:"",

//X6序列化数组

cellsArr:[],

//是否显示全部浮窗

isShowClickAll:0,

offsetX:0,

offsetY:0,

disabled:false,

//模型装置点集

nodes:[],

//是否重新计算碳

isCalCo2:false,

isCal:1,

deviceId:'',

//模型管理搜索装置

showHighLightDevice:false,

deviceNameArr:[],

deviceNameItem:{}

};

},

created() {

this.getList();

this.getCompanyList();

},

methods: {

/\*\*查询列表 \*/

getList(){

listModel(this.queryParams).then(response=>{

this.modelList=response.rows;

this.total=response.total;

this.loading=false

})

getPriceSystem().then(response=>{

this.priceSystemArr=response.rows

})

},

getCompanyList(){

listCompanyDict().then(response=>{

let arr=response.rows;

let companyName=[];

let companyZone=[];

let companyState=[];

let companyCity=[];

let companyParent=[];

let companyArea=[];

for(let item in arr){

companyName.push({

"name":arr[item].companyName,

"id":arr[item].companyId,

});

companyZone.push(arr[item].companyZone);

companyState.push(arr[item].companyState);

companyCity.push(arr[item].companyCity)

companyParent.push(arr[item].companyParent);

companyArea.push(arr[item].companyArea);

}

this.companyName=new Set(companyName);

this.companyZone=new Set(companyZone);

this.companyState=new Set(companyState);

this.companyCity=new Set(companyCity);

this.companyParent=new Set(companyParent);

this.companyArea=new Set(companyArea);

})

},

// 取消按钮

cancel() {

this.addDialog = false;

this.reset();

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.addDialog = true;

this.title = "添加模型管理";

},

// 表单重置

reset() {

this.form = {

modelName: undefined,

beginTime:undefined,

endTime:undefined,

totalFactoryCo2:undefined,

totalProductCo2:undefined,

totalInput:undefined,

totalOutput:undefined,

fileUrl:undefined

};

this.companyList.companyName=undefined;

this.companyList.companyZone=undefined;

this.companyList.companyState=undefined;

this.companyList.companyCity=undefined;

this.companyList.companyArea=undefined;

this.companyList.companyParent=undefined;

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.modelId);

this.names=selection.map((item) => item.modelName);

this.single = selection.length !== 1;

this.multiple = !selection.length;

},

/\*\* 新增按钮操作 \*/

companyChoose(companyId){

getCompanyDict(companyId).then(response=>{

this.companyList=response.data;

this.form.modelId=companyId;

this.form.modelName=this.form.modelName;

this.form.modelFactory=this.companyList.companyName;

this.form.modelRegion=this.companyList.companyZone;

this.form.modelProvince=this.companyList.companyState;

this.form.modelCity=this.companyList.companyCity;

this.form.modelArea=this.companyList.companyArea;

this.form.modelCompany=this.companyList.companyParent;

this.form.beginTime=this.form.beginTime;

this.form.endTime=this.form.endTime;

})

},

priceChoose(val){

this.form.priceSystem=val.priceSystemName;

this.form.priceSystemId=val.priceSystemId;

},

//表格点击弹框修改价格体系

openPrice(val){

this.priceSystem=val.priceSystem;

this.priceSystemData=val

this.visiblePrice=true

},

priceChange(val){

this.priceSystem=val.priceSystemName;

this.priceSystemId=val.priceSystemId

// this.priceSystemData.priceSystem=val.priceSystemName;

// this.priceSystemData.priceSystemId=val.priceSystemId;

},

priceSubmit(){

let priceSystemData=this.priceSystemData;

priceSystemData.priceSystem=this.priceSystem;

priceSystemData.priceSystemId=this.priceSystemId;

this.visiblePrice=false;

updatePriceSystem(priceSystemData).then(response=>{

if(response.code==200){

this.getList();

}else{

this.$modal.msgError("修改失败");

}

})

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const modelIds = row.modelId || this.ids;

const modelNames=row.modelName||this.names

this.$modal

.confirm('是否确认删除"' + modelNames + '"的数据项？')

.then(function () {

return delModel(modelIds);

})

.then((res) => {

this.getList();

this.$modal.msgSuccess("删除成功");

})

.catch(() => {});

},

// 确认导入文件

submitFileForm() {

this.$refs["form"].validate((valid)=>{

if(valid){

this.disabled=true

this.loading=true;

addModel(this.form).then((response)=>{

this.$modal.msgSuccess("新增成功");

this.loading=false;

this.disabled=false

this.addDialog=false;

this.getList()

})

}

})

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

this.fileList=file

},

// 文件上传成功处理

handleFileSuccess(res, file, fileList) {

this.loading=true;

let url=res.url;

this.form.fileUrl=url.substring(url.indexOf('/profile'));

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

if (res.code == 200) {

this.$modal.msgSuccess("恭喜您，数据导入成功！");

}else{

this.$modal.msgError("数据上传失败！");

}

},

/\*\* 导出按钮操作 \*/

handleExport(row) {

if(row.modelId){

this.outerisShowExport=true;

this.modelId=row.modelId;

}else{

this.download('/asm/model/export', {

...this.queryParams

}, `type\_${new Date().getTime()}.xlsx`)

}

},

handleExportLine(){

const numberRows=Number(this.numberRows);

if(numberRows<40||numberRows>150){

this.$modal.msgWarning("行数请在40-150范围内，请重新输入！")

}else{

exportModel(this.modelId,numberRows).then(response => {

this.isShowExport=false;

this.outerisShowExport=false;

if(response.code==200){

this.$modal.msgSuccess("导出成功");

this.$download.name(response.msg);

}else{

this.$modal.msgError("导出错误");

}

})

}

},

handleExportLineOut(val){

if(val==0){

this.isShowExport=true;

}else if(val==1){

let numberRows=''

exportModel(this.modelId,numberRows).then(response => {

this.isShowExport=false;

this.outerisShowExport=false;

if(response.code==200){

this.$modal.msgSuccess("导出成功");

this.$download.name(response.msg);

}else{

this.$modal.msgError("导出错误");

}

})

}

},

//计算

handleCalc(row){

this.$modal.loading("计算中，请稍后...");

let ids=row.modelId

calcModel(ids).then((response)=>{

if(response.code==200){

this.getList();

this.loading=false;

this.$modal.closeLoading();

this.$modal.msgSuccess("操作成功");

}

})

},

//修改模型管理dialog

handleState(row){

this.queryParams.modelId=row.modelId;

this.open = true;

this.modelLoading=true;

editModel(row.modelId).then((response)=>{

this.nodeList=response.data;

this.initX6();

})

},

editDevice(id){

var deviceId = id.split("-")[1];

this.innerOpen=true;

viewModel(deviceId).then((response)=>{

this.viewModelOsbls=response.data.cModelDevice.osbls;

response.data.cModelDevice.osbls.map(item=>{

item.consumption=(parseInt(item.consumption)/10000).toFixed(2)

})

this.viewList=response.data.cModelDevice

})

},

// 修改消耗量

updateConsumption(val){

let row=JSON.parse(JSON.stringify(val));

let consumption=row.consumption\*10000;

row.consumption=consumption;

updateOsbl(row).then(response=>{

if(response.code==200){

this.editDevice(this.deviceId);

}else{

this.$modal.msgError("修改失败")

}

})

},

//是否重新计算碳

submitOsbl(){

// this.innerOpen=false;

this.isCalCo2=true;

},

handleSubmitCalCo2(){

this.isCalCo2=false;

this.innerOpen=false;

if(this.isCal==1){

this.$modal.loading("计算中，请稍后...");

let ids=this.queryParams.modelId;

calcModel(ids).then((response)=>{

if(response.code==200){

this.$modal.closeLoading();

this.modelLoading=true;

this.graph?this.graph.clearCells():'';

this.graph?this.graph.drawGrid():'';

editModel(ids).then((response)=>{

this.nodeList=response.data;

this.$modal.msgSuccess("计算成功！")

this.initX6();

})

}

})

}

},

initX6(){

this.textList=this.$refs.configure;

this.textList.innerHTML= '';

this.textList.style.display='none';

let nodeArr=this.nodeList.nodes;

let edgeArr=this.nodeList.edges;

let devicePosArr=this.nodeList.cModel.devices;

let nodes=[];

let edges=[];

// 点集

for(let item in nodeArr){

let str=''

for(let i in nodeArr[item].deviceName){

str+=(nodeArr[item].deviceName[i]+"\n")

}

nodes.push(

{

id:"device-"+nodeArr[item].deviceId.toString(),

label:str,

process:nodeArr[item].deviceProcess,

co2:nodeArr[item].deviceCo2,

factor:nodeArr[item].deviceFactor,

level:nodeArr[item].level,

input\_width:200,

output\_width: 100,

input\_height: 20,

output\_height: 20,

inputs:nodeArr[item].inputs.map(item => ({

id:"input-"+item.inputId,

name:item.inputName,

quantity:item.inputQuantity,

co2:item.inputCo2,

planname:item.planName,

fontSize:10,

superOutputId:item.superOutputId

})

),

node:nodeArr[item].inputs.map(item=>({

id:"node-"+item.inputId,

})),

outputs:nodeArr[item].outputs.map(item=>({

id:"outpput-"+item.outputId,

name:item.outputName,

quantity:item.outputQuantity,

co2:item.outputCo2,

factor:item.outputCo2Factor,

planname:item.planName,

isCalc:item.isCalc,

fontSize:10

})

)

},

);

}

this.nodes=nodes;

// 边集

for(let item in edgeArr){

let fromlist=edgeArr[item].from.fromlist;

let tolist=edgeArr[item].to.tolist;

edges.push({

source:"device-"+edgeArr[item].from.from.toString(),

sourceAnchor:edgeArr[item].fromIndex,

sourceAllPoint:fromlist[0].map((item)=>{

return "input-"+item.inputId.toString()

}),

sourceInputPoint:fromlist[0].map((item)=>{

return "input-"+item.inputId.toString()

}),

sourceOutputPoint:fromlist[1].map((item)=>{

return "outpput-"+item.outputId.toString()

}),

target:"device-"+edgeArr[item].to.to.toString(),

targetAnchor:edgeArr[item].toIndex,

targetAllPoint:tolist[0].map((item)=>{

return "input-"+item.inputId.toString()

}),

targetInputPoint:tolist[0].map((item)=>{

return "input-"+item.inputId.toString()

}),

targetOutputPoint:tolist[1].map((item)=>{

return "outpput-"+item.outputId.toString()

}),

sourcePoint:[],

targetPoint:[],

label:''

})

}

for(let i in edges){

for(let j in edges[i].sourceOutputPoint){

edges[i].sourceAllPoint.push(edges[i].sourceOutputPoint[j])

}

for(let j in edges[i].targetOutputPoint){

edges[i].targetAllPoint.push(edges[i].targetOutputPoint[j])

}

}

for(let i in edges){

edges[i].sourcePoint=edges[i].sourceAllPoint[edges[i].sourceAnchor];

edges[i].targetPoint=edges[i].targetAllPoint[edges[i].targetAnchor];

}

this.graph = new Graph({

container: document.getElementById('container'),

grid: true,

panning: {

enabled: true,

// modifiers: 'shift',

},

mousewheel: {

enabled: true,

zoomAtMousePosition: true,

modifiers: 'ctrl',

minScale: 0.5,

maxScale: 3,

},

connecting: {

router: {

name: 'manhattan',

args: {

padding: 1,

step:60,

startDirections: ['right'],

endDirections: ['left'],

},

},

connector: {

name: 'jumpover',

args: {

type:'gap',

radius: 8

},

},

anchor: 'center',

connectionPoint: 'anchor',

allowBlank: false,

allowLoop: false,

snap: {

radius: 20,

},

},

history: {

enabled: true

},

})

Graph.registerEdge(

'elk-edge',

{

attrs: {

line: {

stroke: '#aaafcd',

strokeWidth: 1.5,

targetMarker:false

},

},

zIndex: 0,

},

true,

)

Graph.registerNode(

'elk-node',

{

inherit: 'rect',

attrs: {

body: {

fill: '#d2edda',

stroke: '#3d714a',

strokeWidth: 2,

rx: 30,

ry: 30,

},

label: {

fill:'#3d714a',

fontWeight:550,

fontSize: 16,

},

},

ports: {

groups: {

left: {

position: {

name: 'left',

},

label: {

position: 'left', // 标签位置

},

attrs: {

circle: {

fill: '#ffffff',

stroke: '#02baff',

strokeWidth: 1,

r: 8,

magnet: false,

style: {

visibility: 'hidden',

},

},

text: {

fill: '#000',

fontSize: 12,

}

},

markup: [

{

tagName: 'circle',

selector: 'portBody',

},

],

},

right:{

position: {

name: 'right',

},

label: {

position: 'right', // 标签位置

},

attrs: {

circle: {

fill: '#ffffff',

stroke: '#02baff',

strokeWidth: 1,

r: 8,

magnet: false,

style: {

visibility: 'hidden',

},

},

text: {

fill: '#000',

fontSize: 12,

},

},

markup: [

{

tagName: 'circle',

selector: 'portBody',

},

],

}

},

},

},

true,

)

Graph.registerNode(

'tipNode',

{

inherit: 'rect',

markup: [

{

tagName: 'rect',

selector: 'body',

},

{

tagName: 'rect',

selector: 'name-rect',

},

{

tagName: 'rect',

selector: 'process-rect',

},

{

tagName: 'rect',

selector: 'co2-rect',

},

{

tagName: 'rect',

selector: 'factor-rect',

},

{

tagName: 'text',

selector: 'name-text',

},

{

tagName: 'text',

selector: 'process-text',

},

{

tagName: 'text',

selector: 'co2-text',

},

{

tagName: 'text',

selector: 'factor-text',

},

],

attrs: {

rect: {

width: 160,

},

body: {

stroke: '#fff',

},

'name-rect': {

fill: '#5f95ff',

stroke: '#fff',

strokeWidth: 0.5,

},

'process-rect': {

fill: '#eff4ff',

stroke: '#fff',

strokeWidth: 0.5,

},

'co2-rect': {

fill: '#eff4ff',

stroke: '#fff',

strokeWidth: 0.5,

},

'factor-rect': {

fill: '#eff4ff',

stroke: '#fff',

strokeWidth: 0.5,

},

'name-text': {

ref: 'name-rect',

refY: 0.5,

refX: 0.5,

textAnchor: 'middle',

fontWeight: 'bold',

fill: '#fff',

fontSize: 12,

},

'process-text': {

ref: 'process-rect',

refY: 0.5,

refX: 5,

textAnchor: 'left',

fill: 'black',

fontSize: 10,

},

'co2-text': {

ref: 'co2-rect',

refY: 0.5,

refX: 5,

textAnchor: 'left',

fill: 'black',

fontSize: 10,

},

'factor-text': {

ref: 'factor-rect',

refY: 0.5,

refX: 5,

textAnchor: 'left',

fill: 'black',

fontSize: 10,

},

},

// 通过钩子将自定义选项 label 应用到 'attrs/text/text' 属性上

propHooks(metadata) {

const { name,process,co2,factor, ...others } = metadata;

const rects = [

{ type: 'name', text: name },

{ type: 'process', text: process },

{ type: 'co2', text: co2 },

{ type: 'factor', text: factor }

]

let offsetY = 0

rects.forEach((rect) => {

if(rect.text){

const height = 30

ObjectExt.setByPath(

others,

`attrs/${rect.type}-text/text`,

rect.text,

)

ObjectExt.setByPath(others, `attrs/${rect.type}-rect/height`, height)

ObjectExt.setByPath(

others,

`attrs/${rect.type}-rect/transform`,

'translate(0,' + offsetY + ')',

)

offsetY += height

}

})

others.size = { width: 160, height: offsetY }

return others

},

},

true,

)

//节点自适应高度

let level=0

//第一个节点的坐标x,y

let y =200;

let x =200;

//节点的高度

let nodesHeight=0;

const portIdToNodeIdMap={};

const cells = [];

for(let i in nodes){

nodes[i].inputoutputs=[];

for(let j in nodes[i].outputs){

nodes[i].inputoutputs.push(nodes[i].outputs[j])

}

for(let j in nodes[i].inputs){

nodes[i].inputoutputs.push(nodes[i].inputs[j])

}

}

// //循环节点

// for(let i in devicePosArr){

// devicePosArr[i].deviceId="device-"+devicePosArr[i].deviceId.toString();

// if(devicePosArr[i].positionX!=0&&devicePosArr[i].positionY!=0){

// devicePosArr[i].position={

// x:devicePosArr[i].positionX,

// y:devicePosArr[i].positionY,

// }

// }

// }

nodes.sort(this.compare('level'))

nodes.forEach((child) => {

if(level!=child.level){

y=200;

x=level\*400;

level=child.level

}else{

y=y+200+nodesHeight

}

//判断inputs和outputs长度取最大值定义节点的高度

if(child.inputs.length>child.outputs.length){

nodesHeight=(child.inputs.length+1)\*30

}else{

nodesHeight=(child.outputs.length+1)\*30

}

// for(let i in devicePosArr){

// if(devicePosArr[i].deviceId==child.id){

// child.position=devicePosArr[i].position

// }

// }

const position={

x:x,

y:y,

};

child.position=position;

const tipPos={

x:child.position?child.position.x+60:x+60,

y:child.position?child.position.y+nodesHeight:y+nodesHeight,

};

cells.push(

this.graph.createNode({

shape: 'elk-node',

id: child.id,

position,

label: child.label,

size: {

width: 60,

height: nodesHeight >= 150 ? nodesHeight : 150,

},

ports: {

items: (child.inputoutputs || []).map((item) => {

portIdToNodeIdMap[item.id] = child.id

if (item.id.search('input') != -1) {

return {

id: item.id,

group: 'left',

attrs: {

text: { text: item.name },

},

}

} else {

return {

id: item.id,

group: 'right',

attrs: {

text: { text: item.name },

},

}

}

}),

},

}),

)

if(this.isShowClickAll==1){

cells.push(

this.graph.createNode({

shape: 'tipNode',

id: 'tip-'+child.id,

name:"装置名称："+child.label.replace(/\n/g,''),

process:"处理量："+child.process.toFixed(0)+'吨',

co2:"碳足迹："+child.co2.toFixed(0)+'吨',

factor:"碳强度："+child.factor.toFixed(2)+'吨/吨',

position:tipPos,

}),

)

}

})

this.graph.resetCells(cells);

//循环边

edges.forEach((child) => {

const sourcePortId = child.sourcePoint;

const targetPortId = child.targetPoint;

const sourceNodeId = portIdToNodeIdMap[sourcePortId];

const targetNodeId = portIdToNodeIdMap[targetPortId];

cells.push(

this.graph.createEdge({

shape: 'elk-edge',

source: {

cell: sourceNodeId,

port: sourcePortId,

},

target: {

cell: targetNodeId,

port: targetPortId,

},

// vertices:pos,

}),

)

})

this.graph.resetCells(cells);

this.modelLoading=false;

this.graph.centerContent();

this.graph.zoom(-0.6)

//点击边

this.graph.on('edge:click', ({ e, x, y, edge, view }) => {

this.resetInit()

edge.attr('line/stroke', '#779ac0');

edge.attr('line/strokeWidth', '4');

edge.addTools([

{

name: 'segments',

args: {

snapRadius: 10,

threshold: 10,

attrs: {

fill: '#666',

},

},

},{

name: 'vertices',

args: {

attrs: { fill: '#666' },

},

},

])

})

//双击节点

this.graph.on('node:dblclick', (e) => {

if(e.cell.id.includes('tip')==false){

this.deviceId=e.node.id

this.editDevice(e.node.id);

}

});

//单击节点

this.graph.on('node:click', ({e,node,view}) => {

const nodesArr=this.nodeList.nodes;

let clickNodeName='';

let clickNodeNameDevice='';

let nodeId=node.id.split('-')[1];

nodesArr.map(item=>{

if(nodeId==item.deviceId){

let ports=item.outputs;

for(let i in ports){

if(e.target.innerHTML==ports[i].outputName){

clickNodeName=ports[i].outputId;

clickNodeNameDevice='outpput-'+(ports[i].outputId).toString();

}

}

}

})

//获取所有的装置

const nodesAll=this.graph.getNodes(node, { deep: true });

let outputPortsDevice=e.target.outerHTML.split('"')[1];

let outputPorts=(e.target.outerHTML.split('"')[1]).split('-')[1];

if(clickNodeName){

outputPorts=clickNodeName;

outputPortsDevice=clickNodeNameDevice

}

if(outputPorts&&outputPortsDevice.includes('out')==true){

this.resetInit();

//点击装置高亮

node.attr('body/fill', '#fff3cd');

node.attr('body/stroke', '#a48b55');

node.attr('label/fill','#a48b55');

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId=nodes[j].inputs[h].superOutputId;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

// for(let a in nodesAll[z].port.ports){

const edges = this.graph.getConnectedEdges(node, { deep: true });

for(let b in edges){

if(outputPortsDevice==edges[b].store.data.source.port){

edges[b].attr('line/stroke', '#a48b55');

edges[b].attr('line/strokeWidth', '3');

for(let d in nodes){

if(edges[b].store.data.target.cell==nodes[d].id){

var secondDevice=nodes[d]

}

}

}

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

}

// }

}

}

}

}

}

let thirdDevice=[];

if(secondDevice){

for(let a in secondDevice.outputs){

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId='outpput-'+nodes[j].inputs[h].superOutputId;

let outputPorts=secondDevice.outputs[a].id;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

const secondEdges = this.graph.getConnectedEdges(nodesAll[z], { deep: true });

for(let b in secondEdges){

if(outputPorts==secondEdges[b].store.data.source.port){

secondEdges[b].attr('line/stroke', '#a48b55');

secondEdges[b].attr('line/strokeWidth', '3');

for(let d in nodes){

if(secondEdges[b].store.data.target.cell==nodes[d].id){

thirdDevice.push(nodes[d])

}

}

}

}

}

}

}

}

}

}

}

let fourthDevice=[];

for(let a in thirdDevice){

for(let b in thirdDevice[a].outputs){

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId='outpput-'+nodes[j].inputs[h].superOutputId;

let outputPorts=thirdDevice[a].outputs[b].id;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

const thirdEdges = this.graph.getConnectedEdges(nodesAll[z], { deep: true });

for(let c in thirdEdges){

if(outputPorts==thirdEdges[c].store.data.source.port){

thirdEdges[c].attr('line/stroke', '#a48b55');

thirdEdges[c].attr('line/strokeWidth', '3');

for(let d in nodes){

if(thirdEdges[c].store.data.target.cell==nodes[d].id){

fourthDevice.push(nodes[d])

}

}

}

}

}

}

}

}

}

}

}

let fifthDevice=[];

for(let a in fourthDevice){

for(let b in fourthDevice[a].outputs){

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId='outpput-'+nodes[j].inputs[h].superOutputId;

let outputPorts=fourthDevice[a].outputs[b].id;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

const fourthEdges = this.graph.getConnectedEdges(nodesAll[z], { deep: true });

for(let c in fourthEdges){

if(outputPorts==fourthEdges[c].store.data.source.port){

fourthEdges[c].attr('line/stroke', '#a48b55');

fourthEdges[c].attr('line/strokeWidth', '3');

for(let d in nodes){

if(fourthEdges[c].store.data.target.cell==nodes[d].id){

fifthDevice.push(nodes[d])

}

}

}

}

}

}

}

}

}

}

}

let sixthDevice=[]

for(let a in fifthDevice){

for(let b in fifthDevice[a].outputs){

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId='outpput-'+nodes[j].inputs[h].superOutputId;

let outputPorts=fifthDevice[a].outputs[b].id;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

const fifthEdges = this.graph.getConnectedEdges(nodesAll[z], { deep: true });

for(let c in fifthEdges){

if(outputPorts==fifthEdges[c].store.data.source.port){

fifthEdges[c].attr('line/stroke', '#a48b55');

fifthEdges[c].attr('line/strokeWidth', '3');

for(let d in nodes){

if(fifthEdges[c].store.data.target.cell==nodes[d].id){

sixthDevice.push(nodes[d])

}

}

}

}

}

}

}

}

}

}

}

for(let a in sixthDevice){

for(let b in sixthDevice[a].outputs){

for(let j in nodes){

for(let h in nodes[j].inputs){

let inputsSuperId='outpput-'+nodes[j].inputs[h].superOutputId;

let outputPorts=sixthDevice[a].outputs[b].id;

if(outputPorts==inputsSuperId){

for(let z in nodesAll){

if(nodes[j].id==nodesAll[z].id){

nodesAll[z].attr('body/fill', '#fff3cd');

nodesAll[z].attr('body/stroke', '#a48b55');

nodesAll[z].attr('label/fill','#a48b55');

const sixthEdges = this.graph.getConnectedEdges(nodesAll[z], { deep: true });

for(let c in sixthEdges){

if(outputPorts==sixthEdges[c].store.data.source.port){

sixthEdges[c].attr('line/stroke', '#a48b55');

sixthEdges[c].attr('line/strokeWidth', '3');

}

}

}

}

}

}

}

}

}

}else{

//获取关联的装置和线

const edges = this.graph.getConnectedEdges(node, { deep: true });

const nodesc=this.graph.getNeighbors(node, { deep: true });

if(node.id.includes('tip')==false){

this.resetInit();

//除关联线以外的所有线条隐藏

const allEdges=this.graph.getEdges();

for(let i in allEdges){

allEdges[i].attr('line/strokeWidth', '0');

}

//点击装置高亮

node.attr('body/fill', '#fff3cd');

node.attr('body/stroke', '#a48b55');

node.attr('label/fill','#a48b55');

//关联线高亮

for(let i in edges){

edges[i].attr('line/stroke', '#a48b55');

edges[i].attr('line/strokeWidth', '3');

}

//关联装置高亮

for(let i in nodesc){

nodesc[i].attr('body/fill', '#fff3cd');

nodesc[i].attr('body/stroke', '#a48b55');

nodesc[i].attr('label/fill','#a48b55')

}

}

}

});

//单击空白处画布

this.graph.on('blank:click',()=>{

this.resetInit();

})

this.graph.on('node:mouseover', ({e,node,view}) => {

if(node.id.includes('tip')==false){

this.cellsArr=this.graph.toJSON().cells;

const text=this.$refs.tooltipText;

text.style.fontSize="12px";

var str1

for (var i=0;i<nodes.length;i++){

if(nodes[i].id==node.id){

if(e.target.parentElement.parentNode.firstChild.outerHTML.split('"')[1].split('-')[0]=='input'||e.target.parentElement.parentNode.firstChild.outerHTML.split('"')[1].split('-')[0]=='outpput'){

for (var a=0;a<nodes[i].inputoutputs.length;a++){

if(nodes[i].inputoutputs[a].id==e.target.parentElement.parentNode.firstChild.outerHTML.split('"')[1]){

str1=nodes[i].inputoutputs[a];

if(e.target.parentElement.parentNode.firstChild.outerHTML.split('"')[1].split('-')[0]=='input'){

text.innerHTML= '<div class="flex\_center fontWeight550 tooltipTit"><span>投入碳排放数据</span></div><ul style="padding: 0 30px"><li class="toolLi font10"">产出物名称: '+str1.name+'</li><li class="toolLi font10"">投入量: '+str1.quantity.toFixed(0)+' 吨 </li><li class="toolLi font10"">碳足迹：'+str1.co2.toFixed(0)+'吨 </li></ul>'

}else{

if(str1.isCalc==null){

this.isCalc=1

}else{

this.isCalc=Number(str1.isCalc)

}

text.innerHTML= '<div class="flex\_center fontWeight550 tooltipTit"><span>产出碳排放数据</span></div><ul style="padding: 0 30px"><li class="toolLi font10"">产出物名称: '+str1.name+'</li><li class="toolLi font10"">产出量: '+str1.quantity.toFixed(0)+' 吨 </li><li class="toolLi font10"">碳足迹：'+str1.co2.toFixed(0)+'吨 </li><li class="toolLi font10"">碳强度：'+str1.factor.toFixed(2)+' 吨/吨 </li></ul>'

}

}

}

}else if(e.target.outerHTML.split('"')[1].split('-')[0]=='input'||e.target.outerHTML.split('"')[1].split('-')[0]=='outpput'){

for (var a=0;a<nodes[i].inputoutputs.length;a++){

if(nodes[i].inputoutputs[a].id==e.target.outerHTML.split('"')[1]){

str1=nodes[i].inputoutputs[a]

if(e.target.outerHTML.split('"')[1].split('-')[0]=='input'){

text.innerHTML= '<div class="flex\_center fontWeight550 tooltipTit"><span>投入碳排放数据</span></div><ul style="padding: 0 30px"><li class="toolLi font10"">产出物名称: '+str1.name+'</li><li class="toolLi font10"">投入量: '+str1.quantity.toFixed(0)+' 吨 </li><li class="toolLi font10"">碳足迹：'+str1.co2.toFixed(0)+'吨 </li></ul>'

}else{

if(str1.isCalc==null){

this.isCalc=1

}else{

this.isCalc=Number(str1.isCalc)

}

text.innerHTML= '<div class="flex\_center fontWeight550 tooltipTit"><span>产出碳排放数据</span></div><ul style="padding: 0 30px"><li class="toolLi font10"">产出物名称: '+str1.name+'</li><li class="toolLi font10"">产出量: '+str1.quantity.toFixed(0)+' 吨 </li><li class="toolLi font10"">碳足迹：'+str1.co2.toFixed(0)+'吨 </li><li class="toolLi font10"">碳强度：'+str1.factor.toFixed(2)+' 吨/吨 </li></ul>'

}

}

}

}else {

str1=nodes[i]

text.innerHTML= '<div class="flex\_center fontWeight550 tooltipTit"><span>产出碳排放数据</span></div><ul style="padding: 0 30px"><li class="toolLi font10">装置名称: '+str1.label+'</li><li class="toolLi font10">处理量: '+str1.process.toFixed(0)+' 吨 </li><li class="toolLi font10">碳足迹：'+str1.co2.toFixed(0)+'吨 </li><li class="toolLi font10">碳强度：'+str1.factor.toFixed(2)+' 吨/吨 </li></ul>'

}

}

}

this.touchmove(e)

text.style.display='block'

text.style.left= (this.offsetX+25)+'px'

text.style.top= (this.offsetY+25)+'px'

//显示/隐藏连接桩

let portItem=node.port.ports;

let portArr=[]

for(let i in portItem){

portArr.push(portItem[i].id)

}

let ports = container.querySelectorAll('.x6-port-body')?container.querySelectorAll('.x6-port-body'):'';

let portName=''

let portsArr=[]

for(let i in ports){

if(ports[i].outerHTML){

portName=ports[i].outerHTML.split('"')[1]

}

for(let j in portArr){

if(portName==portArr[j]){

portsArr.push(ports[i])

}

}

}

this.showPorts(portsArr, true)

}

}),

this.graph.on('node:mouseleave', ({e,node,view}) => {

if(node.id.includes('tip')==false){

const text=this.$refs.tooltipText

text.innerHTML= ''

text.style.display='none'

//显示/隐藏连接桩

let ports = container.querySelectorAll('.x6-port-body')?container.querySelectorAll('.x6-port-body'):'';

this.showPorts(ports, false)

}

}),

//鼠标右击事件

this.graph.on('node:contextmenu',({e,node,view})=>{

this.inputIdList=[];

this.textList=this.$refs.configure;

const nodeName=e.target.outerHTML.split('"')[1];

const outputNode=nodeName.substr(8);

const nodesArr=this.nodeList.nodes;

const \_this=this;

if(!e.target.innerHTML){

nodesArr.map(item=>{

for(let i in item.inputs){

if(nodeName.includes('outpput')===true){

if(outputNode==item.inputs[i].superOutputId){

this.textList.innerHTML=`<p class="flex\_align\_center configure\_item" onclick="handleChooseList()">配置物料</p>`;

window.handleChooseList =function(){

\_this.isCalcDialog=true;

}

this.inputIdList.push(item.inputs[i].inputId)

}

}else{

//隐藏右键生成的配置列表

this.textList.innerHTML= ''

this.textList.style.display='none'

}

}

})

this.touchmove(e)

this.textList.style.display='block'

this.textList.style.left= (this.offsetX+25)+'px'

this.textList.style.top= (this.offsetY+60)+'px'

}else{

//点击物料名称

let clickNodeName='';

let nodeId=node.id.split('-')[1];

nodesArr.map(item=>{

if(nodeId==item.deviceId){

let ports=item.outputs;

for(let i in ports){

if(e.target.innerHTML==ports[i].outputName){

clickNodeName=ports[i].outputId;

}

}

}

})

nodesArr.map(item=>{

for(let i in item.inputs){

if(clickNodeName==item.inputs[i].superOutputId){

this.textList.style.display='none'

this.textList.innerHTML=`<p class="flex\_align\_center configure\_item" onclick="handleChooseList()">配置物料</p>`;

window.handleChooseList =function(){

\_this.isCalcDialog=true;

}

this.inputIdList.push(item.inputs[i].inputId)

}

}

})

this.touchmove(e)

this.textList.style.display='block'

this.textList.style.left= (this.offsetX+25)+'px'

this.textList.style.top= (this.offsetY+60)+'px'

}

})

},

showPorts(ports, show){

if(ports[0]){

for (let i = 0, len = ports.length; i < len; i = i + 1) {

ports[i].style.visibility = show ? 'visible' : 'hidden'

}

}

},

compare(property) {

return function(a, b) {

let value1 = a[property];

let value2 = b[property];

return value1 - value2;

};

},

touchmove(e) {

this.offsetX=e.offsetX

this.offsetY=e.offsetY

},

resetInit() {

const edges = this.graph.getEdges();

const nodes = this.graph.getNodes();

edges.forEach((edge) => {

edge.attr('line/stroke', '#aaafcd');

edge.attr('line/strokeWidth', '1.5');

})

nodes.forEach((node) => {

if(node.id.includes('tip')==false){

node.attr('body/fill', '#d2edda');

node.attr('body/stroke', '#3d714a');

node.attr('label/fill','#3d714a');

}

})

//点击空白处隐藏右键生成的配置列表

this.textList.innerHTML= ''

this.textList.style.display='none'

},

toolbarRedo(){

this.graph.redo();

},

toolbarUndo(){

this.graph.undo();

},

toolbarOut(){

this.graph.zoom(0.2)

},

toolbarIn(){

this.graph.zoom(-0.2)

},

toolbarRealZoom(){

this.graph.zoomTo(1)

},

toolbarautoZoom(){

this.graph.zoomToFit({ padding: 10, maxScale: 1 })

},

toolbarCal(){

this.$modal.loading("计算中，请稍后...");

let ids=this.queryParams.modelId;

calcModel(ids).then((response)=>{

if(response.code==200){

this.$modal.closeLoading();

this.modelLoading=true;

this.graph?this.graph.clearCells():'';

this.graph?this.graph.drawGrid():'';

editModel(ids).then((response)=>{

this.nodeList=response.data;

this.$modal.msgSuccess("计算成功！")

this.initX6();

})

}

})

},

toolbarDialog(){

this.graph?this.graph.clearCells():'';

this.graph?this.graph.drawGrid():'';

this.isShowClickAll==0?this.isShowClickAll=1:this.isShowClickAll=0;

this.initX6();

},

toolbarFindDevice(){

this.deviceNameArr=[];

this.showHighLightDevice=true;

let nodeArr=this.nodeList.nodes;

for(let i in nodeArr){

this.deviceNameArr.push({

name:nodeArr[i].deviceName,

id:nodeArr[i].deviceId

})

}

},

toolbarExport(){

this.$modal.loading("生成图片中，请稍后...");

this.graph.zoomTo(1);

this.graph.toPNG((dataUri) => {

// 下载

DataUri.downloadDataUri(dataUri, 'model.png');

}, {

padding: {

top: 20,

right: 30,

bottom: 40,

left: 50,

},

copyStyles:false

})

this.graph.zoom(-0.6)

let \_this = this

setTimeout(function() {

\_this.$modal.closeLoading()

},1000);

},

closeInit(){

this.open=false;

this.graph.clearCells()

let cellsArr=this.cellsArr;

let nodesArr=[]

for(let i in cellsArr){

if(cellsArr[i].id.search('device')!=-1){

cellsArr[i].deviceId=Number(cellsArr[i].id.replace('device-','')) ;

nodesArr.push(cellsArr[i])

}

}

this.cellsArr=nodesArr;

setdevicepos(this.cellsArr);

},

//通过装置名称搜索对应装置

deviceChange(val){

this.deviceNameItem.name=val.name;

this.deviceNameItem.id=val.id

},

deviceNameSubmit(){

let id=this.deviceNameItem.id;

let nodes=this.nodes;

let position;

for(let i in nodes){

if(nodes[i].id.indexOf(id)!=-1){

position=nodes[i].position;

this.graph.centerPoint(position.x, position.y);

}

}

const nodesAll=this.graph.getNodes({ deep: true });

for(let i in nodesAll){

nodesAll[i].attr('body/fill', '#d2edda');

nodesAll[i].attr('body/stroke', '#3d714a');

nodesAll[i].attr('label/fill','#3d714a');

if(nodesAll[i].id.indexOf(id)!=-1){

nodesAll[i].attr('body/fill', '#fff3cd');

nodesAll[i].attr('body/stroke', '#a48b55');

nodesAll[i].attr('label/fill','#a48b55');

}

}

this.showHighLightDevice=false;

this.$set(this.deviceNameItem,'name', '')

this.$set(this.deviceNameItem,'id', '')

},

//配置物料是否计算碳

handleSubmitCal(){

this.isCalcDialog=false;

const isCalc=String(this.isCalc);

if(this.inputIdList[0]){

updateConfiguration(this.inputIdList,isCalc).then(response=>{

this.textList=this.$refs.configure;

this.textList.innerHTML= '';

this.textList.style.display='none';

if(response.code==200){

this.$modal.msgSuccess("操作成功");

}else{

this.$modal.msgError("操作失败");

}

})

}

},

UpNumber(e) {

// 输入框中只允许输入小数点和数字，小数点后只能跟一位

e.target.value = (e.target.value.match(/^\d\*(\.?\d{0,4})/g)[0]) || null

}

},

watch:{

open(newVal, oldVal){

if(newVal==false&&oldVal==true){

//关闭画布清空数据

this.nodeList=[];

this.graph?this.graph.clearCells():'';

this.graph?this.graph.drawGrid():'';

this.textList=this.$refs.configure;

this.textList.innerHTML= '';

this.textList.style.display='none';

}

},

isCalcDialog(newVal, oldVal){

if(newVal==false&&oldVal==true){

this.textList=this.$refs.configure;

this.textList.innerHTML= '';

this.textList.style.display='none';

}

},

},

filters: {

numFilter (value) {

// 截取当前数据到小数点后两位

let realVal = parseFloat(value).toFixed(2)

return realVal

},

ChangeDecimalToPercentage(value) {

let data = (value/10000).toFixed(2)

return data

}

}

};

</script>

<style scoped>

#mountNode{

height: 82vh;

}

#container{

margin: 0;

padding: 0;

height: 81vh;

width: 100%;

border-top:2px solid #f5f5f5;

border-bottom:2px solid #f5f5f5;

}

input{

width:480px;

height: 35px;

background: #EEEEEE;

border: 1px solid #529BD1;

padding-left: 5px;

border-radius: 5px;

}

input:active{

box-shadow:1px 1px 10px 1px #529BD1;

}

.todoList{

position: absolute;

left: 0;

top: 0;

width: 28%;

height: 7%;

background: #f5f5f5;

box-shadow: 0 0 5px #e6e6e6;

display:flex;

justify-content: space-around;

align-items: center;

}

.todoList>div{

width:14%;

height: 80%;

display:flex;

justify-content:center;

align-items: center;

}

.icon:hover{

background: #d0d0d0;

border-radius: 10px;

display:flex;

justify-content: center;

align-items: center;

cursor: pointer;

}

.tooltiptext {

width: 200px;

background-color: rgba(255,255,255,0.9);

color: #545454;

border-radius: 6px;

padding: 5px 0;

position: absolute;

z-index: 1000;

display: none;

box-shadow:rgb(217, 217, 217) 0px 0px 10px;

}

.configure{

width: 150px;

background-color: rgba(255,255,255,0.9);

color: #545454;

border-radius:2px;

position: absolute;

z-index: 1001;

display: none;

box-shadow:rgb(217, 217, 217) 0px 0px 10px;

}

.stateBtn{

width: 45px;

height: 25px;

background: #fcf6ed;

border: 1px solid #f1dbb6;

border-radius: 4px;

color: #dca550;

}

.stateBtn:hover{

border: none;

background: #dca550;

color: #fff;

}

.stateBtn:active{

background: #c78928;

}

/deep/.el-dialog{

position:absolute;

left:50%;

top:50%;

transform: translate(-50%,-50%);

margin-top:0!important;

}

</style>

<template>

<div class="app-container">

<!-- <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="120px">

<el-form-item label="标准物料名称" prop="materialName">

<el-input

v-model="queryParams.materialName"

placeholder="请输入标准物料名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="标准编码" prop="materialCode">

<el-input

v-model="queryParams.materialCode"

placeholder="请输入标准编码"

clearable

@keyup.enter.native="handleQuery"

/>

</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"

>新增</el-button

>

</el-col>

<right-toolbar :showSearch.sync="showSearch" @queryTable="getList"></right-toolbar>

</el-row>

<el-table

row-key="materialId"

v-loading="loading"

:data="materialList"

:height="tableHight">

<el-table-column label="工厂物料名称" align="center" prop="oilCodeReference.factoryName" min-width="20%">

<template slot-scope="scope">

<el-select v-model="scope.row.oilCodeReference.factoryName" clearable filterable allow-create @change="materieChoose(scope.row.oilCodeReference.factoryName,scope.row,scope.$index)" >

<el-option

v-for="item in materialSelectList"

:key="item.materielId"

:label="item.materielName"

:value="item.materielName">

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column label="工厂物料编码" align="center" prop="oilCodeReference.factoryCode" min-width="20%">

<template slot-scope="scope">

<el-select v-model="scope.row.oilCodeReference.factoryCode" clearable filterable allow-create @change="materieCodeChoose(scope.row.oilCodeReference.factoryCode,scope.row,scope.$index)">

<el-option

v-for="item in materialSelectList"

:key="item.materielId"

:label="item.materielCode"

:value="item.materielCode">

</el-option>

</el-select>

</template>

</el-table-column>

<el-table-column label="标准物料名称" align="center" prop="materialName" min-width="10%"/>

<el-table-column label="标准编码" align="center" prop="materialCode" min-width="10%"/>

<el-table-column label="物料类型" align="center" prop="materialType" min-width="10%">

<template slot-scope="scope">

<dict-tag :options="dict.type.material\_type" :value="scope.row.materialType"/>

</template>

</el-table-column>

<el-table-column label="物料属性1" align="center" prop="materialProp1" min-width="10%">

<template slot-scope="scope">

<dict-tag :options="dict.type.material\_prop1" :value="scope.row.materialProp1"/>

</template>

</el-table-column>

<el-table-column label="物料属性2" align="center" prop="materialProp2" min-width="10%">

<template slot-scope="scope">

<dict-tag :options="dict.type.material\_prop2" :value="scope.row.materialProp2"/>

</template>

</el-table-column>

<el-table-column label="物料属性3" align="center" prop="materialProp3" min-width="10%">

<template slot-scope="scope">

<dict-tag :options="dict.type.material\_prop3" :value="scope.row.materialProp3"/>

</template>

</el-table-column>

</el-table>

<!-- <pagination

v-show="materialList.length>0"

:total="materialList.length"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/> -->

<el-dialog title="新增物料" :visible.sync="addDialog" append-to-body :close-on-click-modal=false>

<el-form label-position="left" ref="form" :model="form" :rules="rules" label-width="120px">

<el-form-item label="标准物料名称" prop="materialName">

<el-input v-model="form.materialName" placeholder="请输入标准物料名称" />

</el-form-item>

<el-form-item label="标准物料编码" prop="materialCode">

<el-input v-model="form.materialCode" placeholder="请输入标准物料编码" />

</el-form-item>

<el-form-item label="物料类型" prop="materialType">

<el-select v-model="form.materialType" placeholder="请选择物料类型" clearable style="width:100%">

<el-option

v-for="dict in dict.type.material\_type"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="物料属性1" prop="materialProp1">

<el-select v-model="form.materialProp1" placeholder="请选择物料属性1" clearable style="width:100%">

<el-option

v-for="dict in dict.type.material\_prop1"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="物料属性2" prop="materialProp2">

<el-select v-model="form.materialProp2" placeholder="请选择物料属性2" clearable style="width:100%">

<el-option

v-for="dict in dict.type.material\_prop2"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="物料属性3" prop="materialProp3">

<el-select v-model="form.materialProp3" placeholder="请选择物料属性3" clearable style="width:100%">

<el-option

v-for="dict in dict.type.material\_prop3"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="工厂物料名称" prop="factoryName">

<el-select v-model="form.factoryName" placeholder="请选择工厂物料名称" clearable filterable allow-create style="width:100%">

<el-option

v-for="item in materialSelectList"

:key="item.materielId"

:label="item.materielName"

:value="item.materielName">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="工厂物料编码" prop="factoryCode">

<el-select v-model="form.factoryCode" placeholder="请选择工厂物料编码" clearable filterable allow-create style="width:100%">

<el-option

v-for="item in materialSelectList"

:key="item.materielId"

:label="item.materielCode"

:value="item.materielCode">

</el-option>

</el-select>

</el-form-item>

</el-form>

<span slot="footer" class="dialog-footer">

<el-button @click="addDialog = false">取 消</el-button>

<el-button type="primary" @click="addSubmit">确 定</el-button>

</span>

</el-dialog>

</div>

</template>

<script>

import {getMaterial,getSelectMaterial,addfactorylist,addMaterialList} from "@/api/asm/model";

export default {

name: "Configure",

dicts: ['material\_type','material\_prop1', 'material\_prop2', 'material\_prop3'],

data() {

return {

// 遮罩层

loading: true,

//表格的高度

tableHight:document.documentElement.clientHeight-200,

// 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,

// 查询参数

queryParams: {

materialName:null,

materialCode:null,

},

// 石化产品管理表格数据

materialList: [],

modelObj:{},

visibleConfig:false,

//新增物料弹框

addDialog:false,

//新增物料

form:{},

rules:{

materialName: [

{ required: true, message: "标准物料名称不能为空", trigger: "blur" },

],

materialCode: [

{ required: true, message: "标准物料编码不能为空", trigger: "blur" },

],

materialType: [

{ required: true, message: "物料类型不能为空", trigger: "change" },

],

factoryName: [

{ required: true, message: "工厂物料名称不能为空", trigger: 'change' },

],

factoryCode: [

{ required: true, message: "工厂物料编码不能为空", trigger: 'change' },

],

},

materialSelectList:[],

materielName:''

};

},

created() {

this.modelObj=JSON.parse(localStorage.getItem('modelObj'))

this.getList();

window.onresize = function () {

this.tableHight=document.documentElement.clientHeight-200;

};

},

methods: {

/\*\* 查询石化产品管理列表 \*/

getList() {

this.loading = true;

getMaterial(this.queryParams,this.modelObj.name,this.modelObj.id).then(response => {

// console.log(response);

this.materialList=response.rows

this.total = response.total;

this.loading = false;

});

getSelectMaterial(this.modelObj.id).then(response=>{

// console.log(response);

this.materialSelectList=response.rows

})

},

// 取消按钮

cancel() {

this.reset();

},

// 表单重置

reset() {

this.form = {

materialName: null,

materialCode:null,

materialType:null,

materialProp1:null,

materialProp2:null,

materialProp3:null,

factoryName:null,

factoryCode:null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

//新增按钮操作

handleAdd(){

this.reset();

this.addDialog = true;

this.form.modelFactory=this.modelObj.name;

this.title = "添加模型管理";

},

addSubmit(){

const list=[];

list.push(this.form);

this.$refs["form"].validate((valid)=>{

if(valid){

addMaterialList(list).then(response=>{

this.addDialog=false;

if(response.code==200){

this.$modal.msgSuccess("操作成功");

this.getList()

}else{

this.$modal.msgError("添加失败")

}

})

}

})

},

//配置物料名称下拉框选择

materieChoose(name,val){

let index=this.materialList.indexOf(val);

const materialSelectList=this.materialSelectList;

for(let i in materialSelectList){

if(name==materialSelectList[i].materielName){

this.materialList[index].oilCodeReference.factoryCode=materialSelectList[i].materielCode

}

}

const obj={

modelFactory:this.modelObj.name,

materialId:val.materialId,

materialCode:val.materialCode,

refernceId:val.oilCodeReference.refernceId?val.oilCodeReference.refernceId:'',

factoryName:val.oilCodeReference.factoryName?val.oilCodeReference.factoryName:name,

factoryCode:val.oilCodeReference.factoryCode

}

const list=[];

list.push(obj)

addfactorylist(list).then(response=>{

if(response.code!=200){

this.$modal.msgError("操作失败")

}

})

},

//配置物料编码下拉框选择

materieCodeChoose(name,val){

let index=this.materialList.indexOf(val);

const materialSelectList=this.materialSelectList;

for(let i in materialSelectList){

if(name==materialSelectList[i].materielCode){

this.materialList[index].oilCodeReference.factoryName=materialSelectList[i].materielName

}

}

const obj={

modelFactory:this.modelObj.name,

materialId:val.materialId,

materialCode:val.materialCode,

refernceId:val.oilCodeReference.refernceId?val.oilCodeReference.refernceId:'',

factoryName:val.oilCodeReference.factoryName?val.oilCodeReference.factoryName:name,

factoryCode:val.oilCodeReference.factoryCode

}

const list=[];

list.push(obj)

addfactorylist(list).then(response=>{

if(response.code!=200){

this.$modal.msgError("操作失败")

}

})

},

},

};

</script>

<style scoped>

/deep/.el-scrollbar {

overflow:auto;

}

</style>

<template>

<div class="app-container">

<el-form size="small" :inline="true" :model="queryParams" ref="queryForm" label-width="68px">

<el-form-item label="模型名称" prop="modelName">

<el-input

v-model="queryParams.modelName"

placeholder="模型名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所属工厂" prop="modelFactory">

<el-select v-model="queryParams.modelFactory" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyName"

:key="item.id"

:label="item.name"

:value="item.id">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属区域" prop="modelRegion">

<el-select v-model="queryParams.modelRegion" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyZone"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属省份" prop="modelProvince">

<el-select v-model="queryParams.modelProvince" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyState"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属城市" prop="modelCity">

<el-select v-model="queryParams.modelCity" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyCity"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属区（县）" prop="modelArea" label-width="100px">

<el-select v-model="queryParams.modelArea" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyArea"

:key="item"

:label="item"

:value="item">

</el-option>

</el-select>

</el-form-item>

<el-form-item label="所属公司" prop="modelCompany">

<el-select v-model="queryParams.modelCompany" placeholder="请选择" clearable @keyup.enter.native="handleQuery">

<el-option

v-for="item in companyParent"

:key="item"

:label="item"

:value="item">

</el-option>

</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-table :data="modelList" v-loading="loading">

<el-table-column label="模型名称" align="center" prop="modelName" min-width="6%"></el-table-column>

<el-table-column label="所属工厂" align="center" prop="modelFactory" min-width="12%"></el-table-column>

<el-table-column label="所属区域" align="center" prop="modelRegion" min-width="5%"></el-table-column>

<el-table-column label="所属省份" align="center" prop="modelProvince" min-width="5%"></el-table-column>

<el-table-column label="所属城市" align="center" prop="modelCity" min-width="5%"></el-table-column>

<el-table-column label="所属区/县" align="center" prop="modelArea" min-width="6%"></el-table-column>

<el-table-column label="所属公司" align="center" prop="modelCompany" min-width="6%"></el-table-column>

<el-table-column label="开始时间" align="center" prop="beginTime" min-width="8%">

<template slot-scope="scope">

<span>{{ parseTime(scope.row.beginTime, '{y}-{m}-{d}') }}</span>

</template>

</el-table-column>

<el-table-column label="结束时间" align="center" prop="endTime" min-width="8%">

<template slot-scope="scope">

<span>{{ parseTime(scope.row.endTime, '{y}-{m}-{d}') }}</span>

</template>

</el-table-column>

<el-table-column label="工厂碳排放" align="center" prop="totalFactoryCo2" min-width="7%">

<template slot-scope="scope">

{{scope.row.totalFactoryCo2 | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="产品碳排放" align="center" prop="totalProductCo2" min-width="7%">

<template slot-scope="scope">

{{scope.row.totalProductCo2 | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="总加工量" align="center" prop="totalInput" min-width="6%">

<template slot-scope="scope">

{{scope.row.totalInput | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="总产出量" align="center" prop="totalOutput" min-width="6%">

<template slot-scope="scope">

{{scope.row.totalOutput | ChangeDecimalToPercentage}}

</template>

</el-table-column>

<el-table-column label="价格体系" align="center" prop="priceSystem" min-width="6%">

<template slot-scope="scope">

<div class="flex\_center">

<el-button class="flex\_center stateBtn" type="text" v-if="scope.row.priceSystem" @click="openPrice(scope.row)">

{{scope.row.priceSystem}}

</el-button>

<el-button class="flex\_center font12 stateBtn" v-else @click="openPrice(scope.row)">添加+</el-button>

</div>

</template>

</el-table-column>

<el-table-column label="操作" align="center" min-width="10%">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-link"

@click="toconfigure(scope.row)"

>配置物料编码</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-edit-outline"

@click="statePrice(scope.row)"

>配置物料价格</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"

/>

<!-- 修改价格体系 -->

<el-dialog title="价格体系" :visible.sync="visiblePrice" width="500px" append-to-body :close-on-click-modal=false>

<el-form :model="form">

<el-form-item label="价格体系">

<el-select v-model="priceSystem" style="width:380px" @change="priceChange">

<el-option

v-for="item in priceSystemArr"

:key="item.priceSystemId"

:label="item.priceSystemName"

:value="item">

</el-option>

</el-select>

</el-form-item>

</el-form>

<div slot="footer" class="dialog-footer">

<el-button @click="visiblePrice = false">取 消</el-button>

<el-button type="primary" @click="priceSubmit">确 定</el-button>

</div>

</el-dialog>

<!-- 配置物料价格 -->

<el-dialog title="配置物料价格" :visible.sync="dialogPrice" :close-on-click-modal=false>

<el-table :data="priceList" v-loading="priceLoading" height="600px">

<el-table-column prop="materialName" align="center" label="标准物料名称"></el-table-column>

<el-table-column prop="materialCode" align="center" label="标准物料编码"></el-table-column>

<el-table-column prop="factoryName" align="center" label="工厂物料名称"></el-table-column>

<el-table-column prop="factoryCode" align="center" label="工厂编码"></el-table-column>

<el-table-column prop="oilSystemPrice.priceValue" align="center" label="物料价格">

<template slot-scope="scope">

<el-input v-model.number="scope.row.oilSystemPrice.priceValue" @blur="priceValBlur(scope.row)" placeholder="请输入物料价格"></el-input>

</template>

</el-table-column>

</el-table>

<span slot="footer" class="dialog-footer">

<el-button @click="dialogPrice = false">取 消</el-button>

<el-button type="primary" @click="submitPriceForm">确 定</el-button>

</span>

</el-dialog>

<!-- 配置物料是否计算碳 -->

<el-dialog title="配置物料" :visible.sync="isCalcDialog" append-to-body :close-on-click-modal=false>

<span class="font16">物料是否计算碳：</span>

<div style="margin-left: 10px;display: inline-block;">

<el-radio v-model="isCalc" size="medium" :label="0"><span class="font16">是</span></el-radio>

<el-radio v-model="isCalc" size="medium" :label="1"><span class="font16">否</span></el-radio>

</div>

<span slot="footer" class="dialog-footer">

<el-button @click="isCalcDialog = false">取 消</el-button>

<el-button type="primary" @click="handleSubmitCal">确 定</el-button>

</span>

</el-dialog>

</div>

</template>

<script>

import {listModel,getPriceSystem,updatePriceSystem,getfactorypricelist,

updateFactoryPricelist,updateConfiguration} from "@/api/asm/model";

import {listCompanyDict} from "@/api/asm/companyDict";

export default {

name: "ModelConfig",

data() {

return {

// 遮罩层

loading: true,

// 模型名称表格数据

modelList:[],

//所属工厂集合数组

companyName:[],

companyZone:[],

companyState:[],

companyCity:[],

companyParent:[],

companyArea:[],

//当前选中所属工厂数据

companyList:[],

//价格体系列表

priceSystemArr:[],

visiblePrice:false,

priceSystem:'',

priceSystemId:'',

//提交修改价格体系

priceSystemData:[],

// 点击确认修改后回显页面上的价格体系

// priceSystem:'',

// 总条数

total: 0,

form:{},

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

modelName:null,

modelFactory:null,

modelRegion:null,

modelProvince:null,

modelCity:null,

modelArea:null,

modelCompany:null,

modelId:null

},

// 弹出层标题

title:"",

//配置物料价格dialog

dialogPrice:false,

priceList:[],

priceLoading:true,

//提交修改物料价格后的数组

updatePriceList:[],

//配置物料是否计算碳

isCalc:0,

isCalcDialog:false,

inputIdList:[],

};

},

created() {

this.getList();

this.getCompanyList();

},

methods: {

/\*\*查询列表 \*/

getList(){

listModel(this.queryParams).then(response=>{

this.modelList=response.rows;

this.total=response.total;

this.loading=false

})

getPriceSystem().then(response=>{

this.priceSystemArr=response.rows

})

},

getCompanyList(){

listCompanyDict().then(response=>{

let arr=response.rows;

let companyName=[];

let companyZone=[];

let companyState=[];

let companyCity=[];

let companyParent=[];

let companyArea=[];

for(let item in arr){

companyName.push({

"name":arr[item].companyName,

"id":arr[item].companyId,

});

companyZone.push(arr[item].companyZone);

companyState.push(arr[item].companyState);

companyCity.push(arr[item].companyCity)

companyParent.push(arr[item].companyParent);

companyArea.push(arr[item].companyArea);

}

this.companyName=new Set(companyName);

this.companyZone=new Set(companyZone);

this.companyState=new Set(companyState);

this.companyCity=new Set(companyCity);

this.companyParent=new Set(companyParent);

this.companyArea=new Set(companyArea);

})

},

// 取消按钮

cancel() {

this.reset();

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.title = "添加模型管理";

},

// 表单重置

reset() {

this.form = {

modelName: undefined,

beginTime:undefined,

endTime:undefined,

totalFactoryCo2:undefined,

totalProductCo2:undefined,

totalInput:undefined,

totalOutput:undefined,

fileUrl:undefined

};

this.companyList.companyName=undefined;

this.companyList.companyZone=undefined;

this.companyList.companyState=undefined;

this.companyList.companyCity=undefined;

this.companyList.companyArea=undefined;

this.companyList.companyParent=undefined;

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();

},

priceChoose(val){

this.form.priceSystem=val.priceSystemName;

this.form.priceSystemId=val.priceSystemId;

},

//表格点击弹框修改价格体系

openPrice(val){

this.priceSystem=val.priceSystem;

this.priceSystemData=val

this.visiblePrice=true

},

priceChange(val){

this.priceSystem=val.priceSystemName;

this.priceSystemId=val.priceSystemId

// this.priceSystemData.priceSystem=val.priceSystemName;

// this.priceSystemData.priceSystemId=val.priceSystemId;

},

priceSubmit(){

let priceSystemData=this.priceSystemData;

priceSystemData.priceSystem=this.priceSystem;

priceSystemData.priceSystemId=this.priceSystemId;

this.visiblePrice=false;

updatePriceSystem(priceSystemData).then(response=>{

if(response.code==200){

this.getList();

}else{

this.$modal.msgError("修改失败");

}

})

},

toconfigure(val){

const modelObj={

name:val.modelName,

id:val.modelId

};

this.$router.push({path: '/modeling/model-configure/index'})

localStorage.setItem('modelObj',JSON.stringify(modelObj))

},

//查询配置价格列表

statePrice(val){

this.dialogPrice=true;

getfactorypricelist(val.modelName,val.priceSystemId).then(response=>{

// console.log(response);

this.priceList=response.data;

this.priceLoading=false

})

},

//修改价格框失去焦点

priceValBlur(row){

this.updatePriceList.push({

materialId:row.oilSystemPrice.materialId,

priceId:row.oilSystemPrice.priceId,

refernceId:row.oilSystemPrice.refernceId,

systemId:row.oilSystemPrice.systemId,

priceValue:row.oilSystemPrice.priceValue

})

},

//提交配置好的价格

submitPriceForm(){

if(this.updatePriceList[0]){

updateFactoryPricelist(this.updatePriceList).then(response=>{

this.dialogPrice=false;

if(response.code==200){

this.$modal.msgSuccess("操作成功");

}else{

this.$modal.msgError("修改失败")

}

})

}else{

this.dialogPrice=false;

}

},

//配置物料是否计算碳

handleSubmitCal(){

const isCalc=String(this.isCalc);

if(this.inputIdList[0]){

updateConfiguration(this.inputIdList,isCalc).then(response=>{

this.isCalcDialog=false;

if(response.code==200){

this.$modal.msgSuccess("操作成功");

}else{

this.$modal.msgError("操作失败")

}

})

}

}

},

filters: {

ChangeDecimalToPercentage(value) {

let data = (value/10000).toFixed(2)

return data

}

}

};

</script>

<style scoped>

.stateBtn{

width: 45px;

height: 25px;

background: #fcf6ed;

border: 1px solid #f1dbb6;

border-radius: 4px;

color: #dca550;

}

.stateBtn:hover{

border: none;

background: #dca550;

color: #fff;

}

.stateBtn:active{

background: #c78928;

}

/deep/.el-dialog{

position:absolute;

left:50%;

top:50%;

transform: translate(-50%,-50%);

margin-top:0!important;

}

</style>