石化行业原材料及产品全生命周期碳数据库管理系统源代码

北京杰迅鸿翔信息技术有限公司

### **JavaScript**:

<template>

<div class="app-container">

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

<el-form-item label="企业全称" prop="companyName">

<el-input

v-model="queryParams.companyName"

placeholder="请输入企业全称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="企业简称" prop="companyAbb">

<el-input

v-model="queryParams.companyAbb"

placeholder="请输入企业简称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所处区域" prop="companyZone">

<el-input

v-model="queryParams.companyZone"

placeholder="请输入所处区域"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所处省份" prop="companyState">

<el-input

v-model="queryParams.companyState"

placeholder="请输入所处省份"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所处城市" prop="companyCity">

<el-input

v-model="queryParams.companyCity"

placeholder="请输入所处城市"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所在区县" prop="companyArea">

<el-input

v-model="queryParams.companyArea"

placeholder="请输入所在区县"

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"

v-hasPermi="['asm:companyDict:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:companyDict:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:companyDict: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:companyDict:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="企业ID" align="center" prop="companyId" />

<el-table-column label="企业全称" align="center" prop="companyName" />

<el-table-column label="企业简称" align="center" prop="companyAbb" />

<el-table-column label="显示顺序" align="center" prop="companyOrder" />

<el-table-column label="所属公司" align="center" prop="companyParent" />

<el-table-column label="所处区域" align="center" prop="companyZone" />

<el-table-column label="所处省份" align="center" prop="companyState" />

<el-table-column label="所处城市" align="center" prop="companyCity" />

<el-table-column label="所在区县" align="center" prop="companyArea" />

<el-table-column label="炼厂规模" align="center" prop="companyScale" />

<el-table-column label="乙烯规模" align="center" prop="comanyEthyleneScale" />

<el-table-column label="PX" align="center" prop="comanyPxScale" />

<el-table-column label="碳排放标准量" align="center" prop="companyCo2Plan" />

<el-table-column label="碳排放强度标准量" align="center" prop="companyFactorPlan" />

<el-table-column label="万元产值碳排放标准量" align="center" prop="companyValueCo2Plan" />

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:companyDict:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:companyDict:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改企业字典对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="企业全称" prop="companyName">

<el-input v-model="form.companyName" placeholder="请输入企业全称" />

</el-form-item>

<el-form-item label="企业简称" prop="companyAbb">

<el-input v-model="form.companyAbb" placeholder="请输入企业简称" />

</el-form-item>

<el-form-item label="显示顺序" prop="companyOrder">

<el-input v-model="form.companyOrder" placeholder="请输入显示顺序" />

</el-form-item>

<el-form-item label="所处区域" prop="companyZone">

<el-input v-model="form.companyZone" placeholder="请输入所处区域" />

</el-form-item>

<el-form-item label="所处省份" prop="companyState">

<el-input v-model="form.companyState" placeholder="请输入所处省份" />

</el-form-item>

<el-form-item label="所处城市" prop="companyCity">

<el-input v-model="form.companyCity" placeholder="请输入所处城市" />

</el-form-item>

<el-form-item label="所在区县" prop="companyArea">

<el-input v-model="form.companyArea" placeholder="请输入所在区县" />

</el-form-item>

<el-form-item label="炼厂规模" prop="companyScale">

<el-input v-model="form.companyScale" placeholder="请输入炼厂规模" />

</el-form-item>

<el-form-item label="乙烯规模" prop="comanyEthyleneScale">

<el-input v-model="form.comanyEthyleneScale" placeholder="请输入乙烯规模" />

</el-form-item>

<el-form-item label="PX" prop="comanyPxScale">

<el-input v-model="form.comanyPxScale" placeholder="请输入PX" />

</el-form-item>

<el-form-item label="碳排放标准量" prop="companyCo2Plan">

<el-input v-model="form.companyCo2Plan" placeholder="请输入碳排放标准量" />

</el-form-item>

<el-form-item label="碳排放强度标准量" prop="companyFactorPlan">

<el-input v-model="form.companyFactorPlan" placeholder="请输入碳排放强度标准量" />

</el-form-item>

<el-form-item label="万元产值碳排放标准量" prop="companyValueCo2Plan">

<el-input v-model="form.companyValueCo2Plan" placeholder="请输入万元产值碳排放标准量" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

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

export default {

name: "CompanyDict",

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 企业字典表格数据

companyDictList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

companyName: null,

companyAbb: null,

companyParent: null,

companyZone: null,

companyState: null,

companyCity: null,

companyArea: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

companyName: [

{ required: true, message: "企业全称不能为空", trigger: "blur" }

],

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询企业字典列表 \*/

getList() {

this.loading = true;

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

this.companyDictList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

companyId: null,

companyName: null,

companyAbb: null,

companyOrder: null,

companyParent: null,

companyZone: null,

companyState: null,

companyCity: null,

companyArea: null,

companyScale: null,

comanyEthyleneScale: null,

comanyPxScale: null,

companyCo2Plan: null,

companyFactorPlan: null,

companyValueCo2Plan: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加企业字典";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const companyId = row.companyId || this.ids

getCompanyDict(companyId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改企业字典";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.companyId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const companyIds = row.companyId || this.ids;

this.$modal.confirm('是否确认删除企业字典编号为"' + companyIds + '"的数据项？').then(function() {

return delCompanyDict(companyIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="装置ID" prop="deviceId">

<el-input

v-model="queryParams.deviceId"

placeholder="请输入装置ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.deviceCode"

placeholder="请输入装置编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="上一级产出ID" prop="superOutputId">

<el-input

v-model="queryParams.superOutputId"

placeholder="请输入上一级产出ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input

v-model="queryParams.modelId"

placeholder="请输入所属模型id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input

v-model="queryParams.modelCode"

placeholder="请输入模型编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelName"

placeholder="请输入所属模型名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input

v-model="queryParams.planCode"

placeholder="请输入方案编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input

v-model="queryParams.planName"

placeholder="请输入方案名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料编码" prop="inputCode">

<el-input

v-model="queryParams.inputCode"

placeholder="请输入物料编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料名称" prop="inputName">

<el-input

v-model="queryParams.inputName"

placeholder="请输入物料名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.materialProperty"

placeholder="请输入物料属性"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="是否计算碳" prop="isCalc">

<el-select v-model="queryParams.isCalc" placeholder="请选择是否计算碳" clearable>

<el-option

v-for="dict in dict.type.cms\_content\_charge"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input

v-model="queryParams.createId"

placeholder="请输入创建者id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者" prop="createBy">

<el-input

v-model="queryParams.createBy"

placeholder="请输入创建者"

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"

v-hasPermi="['asm:deviceInput:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:deviceInput:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:deviceInput: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:deviceInput:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="装置投入ID" align="center" prop="inputId" />

<el-table-column label="装置ID" align="center" prop="deviceId" />

<el-table-column label="装置编号" align="center" prop="deviceCode" />

<el-table-column label="上一级产出ID" align="center" prop="superOutputId" />

<el-table-column label="投入量" align="center" prop="inputQuantity" />

<el-table-column label="原料碳足迹" align="center" prop="inputCo2" />

<el-table-column label="所属模型id" align="center" prop="modelId" />

<el-table-column label="模型编号" align="center" prop="modelCode" />

<el-table-column label="所属模型名称" align="center" prop="modelName" />

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

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

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

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

<el-table-column label="物料属性" align="center" prop="materialProperty" />

<el-table-column label="是否计算碳" align="center" prop="isCalc">

<template slot-scope="scope">

<dict-tag :options="dict.type.cms\_content\_charge" :value="scope.row.isCalc"/>

</template>

</el-table-column>

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:deviceInput:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:deviceInput:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改装置投入对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="装置ID" prop="deviceId">

<el-input v-model="form.deviceId" placeholder="请输入装置ID" />

</el-form-item>

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

<el-input v-model="form.deviceCode" placeholder="请输入装置编号" />

</el-form-item>

<el-form-item label="上一级产出ID" prop="superOutputId">

<el-input v-model="form.superOutputId" placeholder="请输入上一级产出ID" />

</el-form-item>

<el-form-item label="投入量" prop="inputQuantity">

<el-input v-model="form.inputQuantity" placeholder="请输入投入量" />

</el-form-item>

<el-form-item label="原料碳足迹" prop="inputCo2">

<el-input v-model="form.inputCo2" placeholder="请输入原料碳足迹" />

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input v-model="form.modelId" placeholder="请输入所属模型id" />

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入模型编号" />

</el-form-item>

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

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

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input v-model="form.planCode" placeholder="请输入方案编码" />

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input v-model="form.planName" placeholder="请输入方案名称" />

</el-form-item>

<el-form-item label="物料编码" prop="inputCode">

<el-input v-model="form.inputCode" placeholder="请输入物料编码" />

</el-form-item>

<el-form-item label="物料名称" prop="inputName">

<el-input v-model="form.inputName" placeholder="请输入物料名称" />

</el-form-item>

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

<el-input v-model="form.materialProperty" placeholder="请输入物料属性" />

</el-form-item>

<el-form-item label="是否计算碳" prop="isCalc">

<el-select v-model="form.isCalc" placeholder="请选择是否计算碳">

<el-option

v-for="dict in dict.type.cms\_content\_charge"

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import { listDeviceInput, getDeviceInput, delDeviceInput, addDeviceInput, updateDeviceInput } from "@/api/asm/deviceInput";

export default {

name: "DeviceInput",

dicts: ['cms\_content\_charge'],

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 装置投入表格数据

deviceInputList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

deviceId: null,

deviceCode: null,

superOutputId: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

inputCode: null,

inputName: null,

materialProperty: null,

isCalc: null,

createId: null,

createBy: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询装置投入列表 \*/

getList() {

this.loading = true;

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

this.deviceInputList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

inputId: null,

deviceId: null,

deviceCode: null,

superOutputId: null,

inputQuantity: null,

inputCo2: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

inputCode: null,

inputName: null,

materialProperty: null,

isCalc: null,

createId: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加装置投入";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const inputId = row.inputId || this.ids

getDeviceInput(inputId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改装置投入";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.inputId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const inputIds = row.inputId || this.ids;

this.$modal.confirm('是否确认删除装置投入编号为"' + inputIds + '"的数据项？').then(function() {

return delDeviceInput(inputIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="设备ID" prop="deviceId">

<el-input

v-model="queryParams.deviceId"

placeholder="请输入设备ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.deviceCode"

placeholder="请输入装置编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="装置名称" prop="deviceName">

<el-input

v-model="queryParams.deviceName"

placeholder="请输入装置名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="公用工程ID" prop="osblId">

<el-input

v-model="queryParams.osblId"

placeholder="请输入公用工程ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="公用工程名称" prop="osblName">

<el-input

v-model="queryParams.osblName"

placeholder="请输入公用工程名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="公用工程编码" prop="osblCode">

<el-input

v-model="queryParams.osblCode"

placeholder="请输入公用工程编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input

v-model="queryParams.modelId"

placeholder="请输入所属模型id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input

v-model="queryParams.modelCode"

placeholder="请输入模型编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelName"

placeholder="请输入所属模型名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input

v-model="queryParams.planCode"

placeholder="请输入方案编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input

v-model="queryParams.planName"

placeholder="请输入方案名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input

v-model="queryParams.createId"

placeholder="请输入创建者id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者" prop="createBy">

<el-input

v-model="queryParams.createBy"

placeholder="请输入创建者"

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"

v-hasPermi="['asm:deviceOsbl:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:deviceOsbl:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:deviceOsbl: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:deviceOsbl:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="消耗ID" align="center" prop="consumeId" />

<el-table-column label="设备ID" align="center" prop="deviceId" />

<el-table-column label="装置编号" align="center" prop="deviceCode" />

<el-table-column label="装置名称" align="center" prop="deviceName" />

<el-table-column label="公用工程ID" align="center" prop="osblId" />

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

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

<el-table-column label="消耗量" align="center" prop="consumption" />

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

<el-table-column label="所属模型id" align="center" prop="modelId" />

<el-table-column label="模型编号" align="center" prop="modelCode" />

<el-table-column label="所属模型名称" align="center" prop="modelName" />

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

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

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:deviceOsbl:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:deviceOsbl:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改装置消耗对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="设备ID" prop="deviceId">

<el-input v-model="form.deviceId" placeholder="请输入设备ID" />

</el-form-item>

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

<el-input v-model="form.deviceCode" placeholder="请输入装置编号" />

</el-form-item>

<el-form-item label="装置名称" prop="deviceName">

<el-input v-model="form.deviceName" placeholder="请输入装置名称" />

</el-form-item>

<el-form-item label="公用工程ID" prop="osblId">

<el-input v-model="form.osblId" placeholder="请输入公用工程ID" />

</el-form-item>

<el-form-item label="公用工程名称" prop="osblName">

<el-input v-model="form.osblName" placeholder="请输入公用工程名称" />

</el-form-item>

<el-form-item label="公用工程编码" prop="osblCode">

<el-input v-model="form.osblCode" placeholder="请输入公用工程编码" />

</el-form-item>

<el-form-item label="消耗量" prop="consumption">

<el-input v-model="form.consumption" placeholder="请输入消耗量" />

</el-form-item>

<el-form-item label="碳排放量" prop="carbonEmission">

<el-input v-model="form.carbonEmission" placeholder="请输入碳排放量" />

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input v-model="form.modelId" placeholder="请输入所属模型id" />

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入模型编号" />

</el-form-item>

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

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

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input v-model="form.planCode" placeholder="请输入方案编码" />

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input v-model="form.planName" placeholder="请输入方案名称" />

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import { listDeviceOsbl, getDeviceOsbl, delDeviceOsbl, addDeviceOsbl, updateDeviceOsbl } from "@/api/asm/deviceOsbl";

export default {

name: "DeviceOsbl",

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 装置消耗表格数据

deviceOsblList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

deviceId: null,

deviceCode: null,

deviceName: null,

osblId: null,

osblName: null,

osblCode: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

createId: null,

createBy: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询装置消耗列表 \*/

getList() {

this.loading = true;

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

this.deviceOsblList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

consumeId: null,

deviceId: null,

deviceCode: null,

deviceName: null,

osblId: null,

osblName: null,

osblCode: null,

consumption: null,

carbonEmission: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

createId: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加装置消耗";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const consumeId = row.consumeId || this.ids

getDeviceOsbl(consumeId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改装置消耗";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.consumeId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const consumeIds = row.consumeId || this.ids;

this.$modal.confirm('是否确认删除装置消耗编号为"' + consumeIds + '"的数据项？').then(function() {

return delDeviceOsbl(consumeIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="设备ID" prop="deviceId">

<el-input

v-model="queryParams.deviceId"

placeholder="请输入设备ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.deviceCode"

placeholder="请输入装置编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料名称" prop="outputName">

<el-input

v-model="queryParams.outputName"

placeholder="请输入物料名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料编码" prop="outputCode">

<el-input

v-model="queryParams.outputCode"

placeholder="请输入物料编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input

v-model="queryParams.modelId"

placeholder="请输入所属模型id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input

v-model="queryParams.modelCode"

placeholder="请输入模型编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelName"

placeholder="请输入所属模型名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input

v-model="queryParams.planCode"

placeholder="请输入方案编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input

v-model="queryParams.planName"

placeholder="请输入方案名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.materialProperty"

placeholder="请输入物料属性"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="重点产品分类" prop="keyProudctClass">

<el-input

v-model="queryParams.keyProudctClass"

placeholder="请输入重点产品分类"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="是否计算碳" prop="isCalc">

<el-select v-model="queryParams.isCalc" placeholder="请选择是否计算碳" clearable>

<el-option

v-for="dict in dict.type.cms\_content\_charge"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input

v-model="queryParams.createId"

placeholder="请输入创建者id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者" prop="createBy">

<el-input

v-model="queryParams.createBy"

placeholder="请输入创建者"

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"

v-hasPermi="['asm:deviceOutput:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:deviceOutput:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:deviceOutput: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:deviceOutput:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="产出ID" align="center" prop="outputId" />

<el-table-column label="设备ID" align="center" prop="deviceId" />

<el-table-column label="装置编号" align="center" prop="deviceCode" />

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

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

<el-table-column label="产出量" align="center" prop="outputQuantity" />

<el-table-column label="物料碳足迹" align="center" prop="outputCo2" />

<el-table-column label="物料碳强度" align="center" prop="outputCo2Factor" />

<el-table-column label="所属模型id" align="center" prop="modelId" />

<el-table-column label="模型编号" align="center" prop="modelCode" />

<el-table-column label="所属模型名称" align="center" prop="modelName" />

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

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

<el-table-column label="物料属性" align="center" prop="materialProperty" />

<el-table-column label="产品价格" align="center" prop="productPrice" />

<el-table-column label="重点产品分类" align="center" prop="keyProudctClass" />

<el-table-column label="是否计算碳" align="center" prop="isCalc">

<template slot-scope="scope">

<dict-tag :options="dict.type.cms\_content\_charge" :value="scope.row.isCalc"/>

</template>

</el-table-column>

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:deviceOutput:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:deviceOutput:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改装置产出对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="设备ID" prop="deviceId">

<el-input v-model="form.deviceId" placeholder="请输入设备ID" />

</el-form-item>

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

<el-input v-model="form.deviceCode" placeholder="请输入装置编号" />

</el-form-item>

<el-form-item label="物料名称" prop="outputName">

<el-input v-model="form.outputName" placeholder="请输入物料名称" />

</el-form-item>

<el-form-item label="物料编码" prop="outputCode">

<el-input v-model="form.outputCode" placeholder="请输入物料编码" />

</el-form-item>

<el-form-item label="产出量" prop="outputQuantity">

<el-input v-model="form.outputQuantity" placeholder="请输入产出量" />

</el-form-item>

<el-form-item label="物料碳足迹" prop="outputCo2">

<el-input v-model="form.outputCo2" placeholder="请输入物料碳足迹" />

</el-form-item>

<el-form-item label="物料碳强度" prop="outputCo2Factor">

<el-input v-model="form.outputCo2Factor" placeholder="请输入物料碳强度" />

</el-form-item>

<el-form-item label="所属模型id" prop="modelId">

<el-input v-model="form.modelId" placeholder="请输入所属模型id" />

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入模型编号" />

</el-form-item>

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

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

</el-form-item>

<el-form-item label="方案编码" prop="planCode">

<el-input v-model="form.planCode" placeholder="请输入方案编码" />

</el-form-item>

<el-form-item label="方案名称" prop="planName">

<el-input v-model="form.planName" placeholder="请输入方案名称" />

</el-form-item>

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

<el-input v-model="form.materialProperty" placeholder="请输入物料属性" />

</el-form-item>

<el-form-item label="产品价格" prop="productPrice">

<el-input v-model="form.productPrice" placeholder="请输入产品价格" />

</el-form-item>

<el-form-item label="重点产品分类" prop="keyProudctClass">

<el-input v-model="form.keyProudctClass" placeholder="请输入重点产品分类" />

</el-form-item>

<el-form-item label="是否计算碳" prop="isCalc">

<el-select v-model="form.isCalc" placeholder="请选择是否计算碳">

<el-option

v-for="dict in dict.type.cms\_content\_charge"

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import { listDeviceOutput, getDeviceOutput, delDeviceOutput, addDeviceOutput, updateDeviceOutput } from "@/api/asm/deviceOutput";

export default {

name: "DeviceOutput",

dicts: ['cms\_content\_charge'],

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 装置产出表格数据

deviceOutputList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

deviceId: null,

deviceCode: null,

outputName: null,

outputCode: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

materialProperty: null,

keyProudctClass: null,

isCalc: null,

createId: null,

createBy: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询装置产出列表 \*/

getList() {

this.loading = true;

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

this.deviceOutputList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

outputId: null,

deviceId: null,

deviceCode: null,

outputName: null,

outputCode: null,

outputQuantity: null,

outputCo2: null,

outputCo2Factor: null,

modelId: null,

modelCode: null,

modelName: null,

planCode: null,

planName: null,

materialProperty: null,

productPrice: null,

keyProudctClass: null,

isCalc: null,

createId: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加装置产出";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const outputId = row.outputId || this.ids

getDeviceOutput(outputId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改装置产出";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.outputId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const outputIds = row.outputId || this.ids;

this.$modal.confirm('是否确认删除装置产出编号为"' + outputIds + '"的数据项？').then(function() {

return delDeviceOutput(outputIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="模型ID" prop="modelId">

<el-input

v-model="queryParams.modelId"

placeholder="请输入模型ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input

v-model="queryParams.modelCode"

placeholder="请输入模型编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.materielProperty"

placeholder="请输入物料属性"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料编码" prop="materielCode">

<el-input

v-model="queryParams.materielCode"

placeholder="请输入物料编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料名称" prop="materielName">

<el-input

v-model="queryParams.materielName"

placeholder="请输入物料名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input

v-model="queryParams.createId"

placeholder="请输入创建者id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者" prop="createBy">

<el-input

v-model="queryParams.createBy"

placeholder="请输入创建者"

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"

v-hasPermi="['asm:materialBalance:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:materialBalance:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:materialBalance: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:materialBalance:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="物料ID" align="center" prop="materielId" />

<el-table-column label="模型ID" align="center" prop="modelId" />

<el-table-column label="模型编号" align="center" prop="modelCode" />

<el-table-column label="物料属性" align="center" prop="materielProperty" />

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

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

<el-table-column label="量/万吨" align="center" prop="materielQuan" />

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:materialBalance:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:materialBalance:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改物料平衡对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="模型ID" prop="modelId">

<el-input v-model="form.modelId" placeholder="请输入模型ID" />

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入模型编号" />

</el-form-item>

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

<el-input v-model="form.materielProperty" placeholder="请输入物料属性" />

</el-form-item>

<el-form-item label="物料编码" prop="materielCode">

<el-input v-model="form.materielCode" placeholder="请输入物料编码" />

</el-form-item>

<el-form-item label="物料名称" prop="materielName">

<el-input v-model="form.materielName" placeholder="请输入物料名称" />

</el-form-item>

<el-form-item label="量/万吨" prop="materielQuan">

<el-input v-model="form.materielQuan" placeholder="请输入量/万吨" />

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import { listMaterialBalance, getMaterialBalance, delMaterialBalance, addMaterialBalance, updateMaterialBalance } from "@/api/asm/materialBalance";

export default {

name: "MaterialBalance",

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 物料平衡表格数据

materialBalanceList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

modelId: null,

modelCode: null,

materielProperty: null,

materielCode: null,

materielName: null,

createId: null,

createBy: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询物料平衡列表 \*/

getList() {

this.loading = true;

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

this.materialBalanceList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

materielId: null,

modelId: null,

modelCode: null,

materielProperty: null,

materielCode: null,

materielName: null,

materielQuan: null,

createId: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加物料平衡";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const materielId = row.materielId || this.ids

getMaterialBalance(materielId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改物料平衡";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.materielId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const materielIds = row.materielId || this.ids;

this.$modal.confirm('是否确认删除物料平衡编号为"' + materielIds + '"的数据项？').then(function() {

return delMaterialBalance(materielIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

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

<el-input

v-model="queryParams.modelName"

placeholder="请输入名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<!-- <el-form-item label="编号" prop="modelCode">-->

<!-- <el-input-->

<!-- v-model="queryParams.modelCode"-->

<!-- placeholder="请输入编号"-->

<!-- clearable-->

<!-- @keyup.enter.native="handleQuery"-->

<!-- />-->

<!-- </el-form-item>-->

<!-- <el-form-item label="所属企业ID" prop="modelCompanyId">-->

<!-- <el-input-->

<!-- v-model="queryParams.modelCompanyId"-->

<!-- placeholder="请输入所属企业ID"-->

<!-- clearable-->

<!-- @keyup.enter.native="handleQuery"-->

<!-- />-->

<!-- </el-form-item>-->

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

<el-input

v-model="queryParams.modelFactory"

placeholder="请输入所属工厂"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelRegion"

placeholder="请输入所属区域"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelProvince"

placeholder="请输入所属省份"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelCity"

placeholder="请输入所属城市"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelArea"

placeholder="请输入所属区县"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-input

v-model="queryParams.modelCompany"

placeholder="请输入所属公司"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

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

<el-date-picker clearable

v-model="queryParams.beginTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择开始时间">

</el-date-picker>

</el-form-item>

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

<el-date-picker clearable

v-model="queryParams.endTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择结束时间">

</el-date-picker>

</el-form-item>

<!-- <el-form-item label="创建者id" prop="createId">-->

<!-- <el-input-->

<!-- v-model="queryParams.createId"-->

<!-- placeholder="请输入创建者id"-->

<!-- clearable-->

<!-- @keyup.enter.native="handleQuery"-->

<!-- />-->

<!-- </el-form-item>-->

<!-- <el-form-item label="创建者" prop="createBy">-->

<!-- <el-input-->

<!-- v-model="queryParams.createBy"-->

<!-- placeholder="请输入创建者"-->

<!-- clearable-->

<!-- @keyup.enter.native="handleQuery"-->

<!-- />-->

<!-- </el-form-item>-->

<!-- <el-form-item label="更新者id" prop="updateId">-->

<!-- <el-input-->

<!-- v-model="queryParams.updateId"-->

<!-- placeholder="请输入更新者id"-->

<!-- clearable-->

<!-- @keyup.enter.native="handleQuery"-->

<!-- />-->

<!-- </el-form-item>-->

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

<el-input

v-model="queryParams.priceSystemId"

placeholder="请输入价格体系"

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"

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

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

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

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

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

>删除</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-upload2"

size="mini"

@click="handleImport"

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

>导入

</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>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<!-- <el-table-column label="ID" align="center" prop="modelId" />-->

<el-table-column label="名称" align="center" prop="modelName" />

<!-- <el-table-column label="编号" align="center" prop="modelCode" />-->

<!-- <el-table-column label="所属企业ID" align="center" prop="modelCompanyId" />-->

<el-table-column label="所属工厂" align="center" prop="modelCompany" width="300"/>

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

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

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

<el-table-column label="所属区县" align="center" prop="modelArea" width="120"/>

<el-table-column label="所属公司" align="center" prop="modelFactory"/>

<el-table-column label="开始时间" align="center" prop="beginTime" width="100">

<template v-slot="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" width="100">

<template v-slot="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" width="120" :formatter="dataFilter"/>

<el-table-column label="产品碳排放" align="center" prop="totalProductCo2" width="120" :formatter="dataFilter"/>

<el-table-column label="总加工量" align="center" prop="totalInput" width="120" :formatter="dataFilter"/>

<el-table-column label="总产出量" align="center" prop="totalOutput" width="120" :formatter="dataFilter"/>

<el-table-column label="价格体系" align="center" prop="priceSystemId" width="120" :formatter="dataFilter"/>

<!-- <el-table-column label="操作" align="center" class-name="small-padding fixed-width">-->

<!-- <template v-slot="scope">-->

<!-- <el-button-->

<!-- size="mini"-->

<!-- type="text"-->

<!-- icon="el-icon-edit"-->

<!-- @click="handleUpdate(scope.row)"-->

<!-- v-hasPermi="['asm:model:edit']"-->

<!-- >修改</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"

/>

<!-- 添加或修改模型对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

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

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

</el-form-item>

<el-form-item label="编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入编号" />

</el-form-item>

<el-form-item label="所属企业ID" prop="modelCompanyId">

<el-input v-model="form.modelCompanyId" placeholder="请输入所属企业ID" />

</el-form-item>

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

<el-input v-model="form.modelFactory" placeholder="请输入所属工厂" />

</el-form-item>

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

<el-input v-model="form.modelRegion" placeholder="请输入所属区域" />

</el-form-item>

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

<el-input v-model="form.modelProvince" placeholder="请输入所属省份" />

</el-form-item>

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

<el-input v-model="form.modelCity" placeholder="请输入所属城市" />

</el-form-item>

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

<el-input v-model="form.modelCompany" placeholder="请输入所属公司" />

</el-form-item>

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

<el-input v-model="form.modelArea" placeholder="请输入所属区县" />

</el-form-item>

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

<el-date-picker clearable

v-model="form.beginTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择开始时间">

</el-date-picker>

</el-form-item>

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

<el-date-picker clearable

v-model="form.endTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择结束时间">

</el-date-picker>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

<el-form-item label="更新者id" prop="updateId">

<el-input v-model="form.updateId" placeholder="请输入更新者id" />

</el-form-item>

<el-form-item label="工厂碳排放" prop="totalFactoryCo2">

<el-input v-model="form.totalFactoryCo2" placeholder="请输入工厂碳排放" />

</el-form-item>

<el-form-item label="产品碳排放" prop="totalProductCo2">

<el-input v-model="form.totalProductCo2" placeholder="请输入产品碳排放" />

</el-form-item>

<el-form-item label="总加工量" prop="totalInput">

<el-input v-model="form.totalInput" placeholder="请输入总加工量" />

</el-form-item>

<el-form-item label="总产出量" prop="totalOutput">

<el-input v-model="form.totalOutput" placeholder="请输入总产出量" />

</el-form-item>

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

<el-input v-model="form.priceSystemId" placeholder="请输入价格体系" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

<!-- 导入对话框 -->

<el-dialog :title="upload.title" :close-on-click-modal="false" :visible.sync="upload.open" width="400px" append-to-body>

<el-upload

ref="upload"

:limit="1"

accept=".xlsx, .xls"

:headers="upload.headers"

:action="upload.url + '?updateSupport=' + upload.updateSupport"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="false"

drag

>

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

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

<div class="el-upload\_\_tip text-center" slot="tip">

<!-- <div class="el-upload\_\_tip" slot="tip">-->

<!-- <el-checkbox v-model="upload.updateSupport"/>-->

<!-- 是否更新已经存在的数据-->

<!-- </div>-->

<span>仅允许导入xls、xlsx格式文件。</span>

<!-- <el-link type="primary" :underline="false" style="font-size:12px;vertical-align: baseline;" @click="importTemplate">下载模板</el-link>-->

</div>

</el-upload>

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

<el-button type="primary" @click="submitFileForm" v-loading.fullscreen.lock="fullscreenLoading">确 定</el-button>

<el-button @click="upload.open = false">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import { listModel, getModel, delModel, addModel, updateModel } from "@/api/asm/model";

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

export default {

name: "Model",

data() {

return {

fullscreenLoading:false,

loadProgress: 0, // 动态显示进度条

progressFlag: false, // 关闭进度条

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 模型表格数据

modelList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

modelName: null,

modelCode: null,

modelCompanyId: null,

modelFactory: null,

modelRegion: null,

modelProvince: null,

modelCity: null,

modelCompany: null,

modelArea: null,

beginTime: null,

endTime: null,

createId: null,

createBy: null,

updateId: null,

priceSystemId: null

},

// 表单参数

form: {},

// 表单校验

rules: {

modelName: [

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

],

modelCode: [

{ required: true, message: "编号不能为空", trigger: "blur" }

],

},

// 导入参数

upload: {

// 是否显示弹出层

open: false,

// 弹出层标题

title: "",

// 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: "Bearer " + getToken()},

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/asm/model/importData"

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询模型列表 \*/

getList() {

this.loading = true;

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

this.modelList = response.rows;

this.total = response.total;

this.loading = false;

});

},

dataFilter(row, column, cellValue, index){

return (cellValue/10000).toFixed(2)

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

modelId: null,

modelName: null,

modelCode: null,

modelCompanyId: null,

modelFactory: null,

modelRegion: null,

modelProvince: null,

modelCity: null,

modelCompany: null,

modelArea: null,

beginTime: null,

endTime: null,

createId: null,

createBy: null,

createTime: null,

updateId: null,

updateBy: null,

updateTime: null,

totalFactoryCo2: null,

totalProductCo2: null,

totalInput: null,

totalOutput: null,

priceSystem: null,

priceSystemId: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加模型";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const modelId = row.modelId || this.ids

getModel(modelId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改模型";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.modelId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

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

this.$modal.confirm('是否确认删除模型编号为"' + modelIds + '"的数据项？').then(function() {

return delModel(modelIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导入按钮操作 \*/

handleImport() {

this.upload.title = "导入";

this.upload.open = true;

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

},

// 文件上传成功处理

handleFileSuccess(response, file, fileList) {

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

this.$alert("<div style='overflow: auto;overflow-x: hidden;max-height: 70vh;padding: 10px 20px 0;'>" + response.msg + "</div>", "导入结果", {dangerouslyUseHTMLString: true});

this.getList();

this.fullscreenLoading = false;

},

// 提交上传文件

submitFileForm() {

this.fullscreenLoading=true

this.$refs.upload.submit();

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="模型ID" prop="modelId">

<el-input

v-model="queryParams.modelId"

placeholder="请输入模型ID"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input

v-model="queryParams.modelCode"

placeholder="请输入模型编号"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料编码" prop="materielCode">

<el-input

v-model="queryParams.materielCode"

placeholder="请输入物料编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="物料名称" prop="materielName">

<el-input

v-model="queryParams.materielName"

placeholder="请输入物料名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="性质编码" prop="propertyCode">

<el-input

v-model="queryParams.propertyCode"

placeholder="请输入性质编码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="性质名称" prop="propertyName">

<el-input

v-model="queryParams.propertyName"

placeholder="请输入性质名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input

v-model="queryParams.createId"

placeholder="请输入创建者id"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="创建者" prop="createBy">

<el-input

v-model="queryParams.createBy"

placeholder="请输入创建者"

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"

v-hasPermi="['asm:modelBlend:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:modelBlend:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:modelBlend: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:modelBlend:export']"

>导出</el-button>

</el-col>

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

</el-row>

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

<el-table-column type="selection" width="55" align="center" />

<el-table-column label="调和ID" align="center" prop="blendId" />

<el-table-column label="模型ID" align="center" prop="modelId" />

<el-table-column label="模型编号" align="center" prop="modelCode" />

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

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

<el-table-column label="性质编码" align="center" prop="propertyCode" />

<el-table-column label="性质名称" align="center" prop="propertyName" />

<el-table-column label="物性值" align="center" prop="propertyValue" />

<el-table-column label="物性单位" align="center" prop="propertyUnit" />

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:modelBlend:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:modelBlend:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改油品调和对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="模型ID" prop="modelId">

<el-input v-model="form.modelId" placeholder="请输入模型ID" />

</el-form-item>

<el-form-item label="模型编号" prop="modelCode">

<el-input v-model="form.modelCode" placeholder="请输入模型编号" />

</el-form-item>

<el-form-item label="物料编码" prop="materielCode">

<el-input v-model="form.materielCode" placeholder="请输入物料编码" />

</el-form-item>

<el-form-item label="物料名称" prop="materielName">

<el-input v-model="form.materielName" placeholder="请输入物料名称" />

</el-form-item>

<el-form-item label="性质编码" prop="propertyCode">

<el-input v-model="form.propertyCode" placeholder="请输入性质编码" />

</el-form-item>

<el-form-item label="性质名称" prop="propertyName">

<el-input v-model="form.propertyName" placeholder="请输入性质名称" />

</el-form-item>

<el-form-item label="物性值" prop="propertyValue">

<el-input v-model="form.propertyValue" placeholder="请输入物性值" />

</el-form-item>

<el-form-item label="物性单位" prop="propertyUnit">

<el-input v-model="form.propertyUnit" placeholder="请输入物性单位" />

</el-form-item>

<el-form-item label="创建者id" prop="createId">

<el-input v-model="form.createId" placeholder="请输入创建者id" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import { listModelBlend, getModelBlend, delModelBlend, addModelBlend, updateModelBlend } from "@/api/asm/modelBlend";

export default {

name: "ModelBlend",

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 油品调和表格数据

modelBlendList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

modelId: null,

modelCode: null,

materielCode: null,

materielName: null,

propertyCode: null,

propertyName: null,

createId: null,

createBy: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询油品调和列表 \*/

getList() {

this.loading = true;

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

this.modelBlendList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

blendId: null,

modelId: null,

modelCode: null,

materielCode: null,

materielName: null,

propertyCode: null,

propertyName: null,

propertyValue: null,

propertyUnit: null,

createId: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加油品调和";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const blendId = row.blendId || this.ids

getModelBlend(blendId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改油品调和";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.blendId != null) {

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const blendIds = row.blendId || this.ids;

this.$modal.confirm('是否确认删除油品调和编号为"' + blendIds + '"的数据项？').then(function() {

return delModelBlend(blendIds);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

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

...this.queryParams

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

}

}

};

</script>

<template>  
 <div class="app-container">  
 <el-form :model="queryParams" ref="queryForm" size="small" :inline="true" v-show="showSearch" label-width="68px">  
 <el-form-item label="模型ID" prop="modelId">  
 <el-input  
 v-model="queryParams.modelId"  
 placeholder="请输入模型ID"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="模型编号" prop="modelCode">  
 <el-input  
 v-model="queryParams.modelCode"  
 placeholder="请输入模型编号"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="模型名称" prop="modelName">  
 <el-input  
 v-model="queryParams.modelName"  
 placeholder="请输入模型名称"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="装置编码" prop="deviceCode">  
 <el-input  
 v-model="queryParams.deviceCode"  
 placeholder="请输入装置编码"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="装置名称" prop="deviceName">  
 <el-input  
 v-model="queryParams.deviceName"  
 placeholder="请输入装置名称"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="虚拟装置" prop="isVirtual">  
 <el-select v-model="queryParams.isVirtual" placeholder="请选择虚拟装置" clearable>  
 <el-option  
 v-for="dict in dict.type.cms\_del\_state"  
 :key="dict.value"  
 :label="dict.label"  
 :value="dict.value"  
 />  
 </el-select>  
 </el-form-item>  
 <el-form-item label="创建者id" prop="createId">  
 <el-input  
 v-model="queryParams.createId"  
 placeholder="请输入创建者id"  
 clearable  
 @keyup.enter.native="handleQuery"  
 />  
 </el-form-item>  
 <el-form-item label="创建者" prop="createBy">  
 <el-input  
 v-model="queryParams.createBy"  
 placeholder="请输入创建者"  
 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"  
 v-hasPermi="['asm:modelDevice:add']"  
 >新增</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="success"  
 plain  
 icon="el-icon-edit"  
 size="mini"  
 :disabled="single"  
 @click="handleUpdate"  
 v-hasPermi="['asm:modelDevice:edit']"  
 >修改</el-button>  
 </el-col>  
 <el-col :span="1.5">  
 <el-button  
 type="danger"  
 plain  
 icon="el-icon-delete"  
 size="mini"  
 :disabled="multiple"  
 @click="handleDelete"  
 v-hasPermi="['asm:modelDevice: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:modelDevice:export']"  
 >导出</el-button>  
 </el-col>  
 <right-toolbar :showSearch.sync="showSearch" @queryTable="getList"></right-toolbar>  
 </el-row>  
  
 <el-table v-loading="loading" :data="modelDeviceList" @selection-change="handleSelectionChange">  
 <el-table-column type="selection" width="55" align="center" />  
 <el-table-column label="id" align="center" prop="deviceId" />  
 <el-table-column label="模型ID" align="center" prop="modelId" />  
 <el-table-column label="模型编号" align="center" prop="modelCode" />  
 <el-table-column label="模型名称" align="center" prop="modelName" />  
 <el-table-column label="装置编码" align="center" prop="deviceCode" />  
 <el-table-column label="装置名称" align="center" prop="deviceName" />  
 <el-table-column label="装置加工量" align="center" prop="deviceProcess" />  
 <el-table-column label="装置碳足迹" align="center" prop="deviceCo2" />  
 <el-table-column label="吨原料碳排放因子" align="center" prop="deviceFactor" />  
 <el-table-column label="虚拟装置" align="center" prop="isVirtual">  
 <template slot-scope="scope">  
 <dict-tag :options="dict.type.cms\_del\_state" :value="scope.row.isVirtual"/>  
 </template>  
 </el-table-column>  
 <el-table-column label="装置x坐标" align="center" prop="positionX" />  
 <el-table-column label="装置y坐标" align="center" prop="positionY" />  
 <el-table-column label="操作" align="center" class-name="small-padding fixed-width">  
 <template slot-scope="scope">  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-edit"  
 @click="handleUpdate(scope.row)"  
 v-hasPermi="['asm:modelDevice:edit']"  
 >修改</el-button>  
 <el-button  
 size="mini"  
 type="text"  
 icon="el-icon-delete"  
 @click="handleDelete(scope.row)"  
 v-hasPermi="['asm:modelDevice:remove']"  
 >删除</el-button>  
 </template>  
 </el-table-column>  
 </el-table>  
   
 <pagination  
 v-show="total>0"  
 :total="total"  
 :page.sync="queryParams.pageNum"  
 :limit.sync="queryParams.pageSize"  
 @pagination="getList"  
 />  
  
 <!-- 添加或修改装置对话框 -->  
 <el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>  
 <el-form ref="form" :model="form" :rules="rules" label-width="80px">  
 <el-form-item label="模型ID" prop="modelId">  
 <el-input v-model="form.modelId" placeholder="请输入模型ID" />  
 </el-form-item>  
 <el-form-item label="模型编号" prop="modelCode">  
 <el-input v-model="form.modelCode" placeholder="请输入模型编号" />  
 </el-form-item>  
 <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="装置编码" prop="deviceCode">  
 <el-input v-model="form.deviceCode" placeholder="请输入装置编码" />  
 </el-form-item>  
 <el-form-item label="装置名称" prop="deviceName">  
 <el-input v-model="form.deviceName" placeholder="请输入装置名称" />  
 </el-form-item>  
 <el-form-item label="装置加工量" prop="deviceProcess">  
 <el-input v-model="form.deviceProcess" placeholder="请输入装置加工量" />  
 </el-form-item>  
 <el-form-item label="装置碳足迹" prop="deviceCo2">  
 <el-input v-model="form.deviceCo2" placeholder="请输入装置碳足迹" />  
 </el-form-item>  
 <el-form-item label="吨原料碳排放因子" prop="deviceFactor">  
 <el-input v-model="form.deviceFactor" placeholder="请输入吨原料碳排放因子" />  
 </el-form-item>  
 <el-form-item label="虚拟装置">  
 <el-radio-group v-model="form.isVirtual">  
 <el-radio  
 v-for="dict in dict.type.cms\_del\_state"  
 :key="dict.value"  
:label="parseInt(dict.value)"  
 >{{dict.label}}</el-radio>  
 </el-radio-group>  
 </el-form-item>  
 <el-form-item label="装置x坐标" prop="positionX">  
 <el-input v-model="form.positionX" placeholder="请输入装置x坐标" />  
 </el-form-item>  
 <el-form-item label="装置y坐标" prop="positionY">  
 <el-input v-model="form.positionY" placeholder="请输入装置y坐标" />  
 </el-form-item>  
 <el-form-item label="创建者id" prop="createId">  
 <el-input v-model="form.createId" placeholder="请输入创建者id" />  
 </el-form-item>  
 </el-form>  
 <div slot="footer" class="dialog-footer">  
 <el-button type="primary" @click="submitForm">确 定</el-button>  
 <el-button @click="cancel">取 消</el-button>  
 </div>  
 </el-dialog>  
 </div>  
</template>  
  
<script>  
import { listModelDevice, getModelDevice, delModelDevice, addModelDevice, updateModelDevice } from "@/api/asm/modelDevice";  
  
export default {  
 name: "ModelDevice",  
 dicts: ['cms\_del\_state'],  
 data() {  
 return {  
 // 遮罩层  
 loading: true,  
 // 选中数组  
 ids: [],  
 // 非单个禁用  
 single: true,  
 // 非多个禁用  
 multiple: true,  
 // 显示搜索条件  
 showSearch: true,  
 // 总条数  
 total: 0,  
 // 装置表格数据  
 modelDeviceList: [],  
 // 弹出层标题  
 title: "",  
 // 是否显示弹出层  
 open: false,  
 // 查询参数  
 queryParams: {  
 pageNum: 1,  
 pageSize: 10,  
 modelId: null,  
 modelCode: null,  
 modelName: null,  
 deviceCode: null,  
 deviceName: null,  
 isVirtual: null,  
 createId: null,  
 createBy: null,  
 },  
 // 表单参数  
 form: {},  
 // 表单校验  
 rules: {  
 }  
 };  
 },  
 created() {  
 this.getList();  
 },  
 methods: {  
 /\*\* 查询装置列表 \*/  
 getList() {  
 this.loading = true;  
 listModelDevice(this.queryParams).then(response => {  
 this.modelDeviceList = response.rows;  
 this.total = response.total;  
 this.loading = false;  
 });  
 },  
 // 取消按钮  
 cancel() {  
 this.open = false;  
 this.reset();  
 },  
 // 表单重置  
 reset() {  
 this.form = {  
 deviceId: null,  
 modelId: null,  
 modelCode: null,  
 modelName: null,  
 deviceCode: null,  
 deviceName: null,  
 deviceProcess: null,  
 deviceCo2: null,  
 deviceFactor: null,  
 isVirtual: 0,  
 positionX: null,  
 positionY: null,  
 createId: null,  
 createBy: null,  
 createTime: null  
 };  
 this.resetForm("form");  
 },  
 /\*\* 搜索按钮操作 \*/  
 handleQuery() {  
 this.queryParams.pageNum = 1;  
 this.getList();  
 },  
 /\*\* 重置按钮操作 \*/  
 resetQuery() {  
 this.resetForm("queryForm");  
 this.handleQuery();  
 },  
 // 多选框选中数据  
 handleSelectionChange(selection) {  
 this.ids = selection.map(item => item.deviceId)  
 this.single = selection.length!==1  
 this.multiple = !selection.length  
 },  
 /\*\* 新增按钮操作 \*/  
 handleAdd() {  
 this.reset();  
 this.open = true;  
 this.title = "添加装置";  
 },  
 /\*\* 修改按钮操作 \*/  
 handleUpdate(row) {  
 this.reset();  
 const deviceId = row.deviceId || this.ids  
 getModelDevice(deviceId).then(response => {  
 this.form = response.data;  
 this.open = true;  
 this.title = "修改装置";  
 });  
 },  
 /\*\* 提交按钮 \*/  
 submitForm() {  
 this.$refs["form"].validate(valid => {  
 if (valid) {  
 if (this.form.deviceId != null) {  
 updateModelDevice(this.form).then(response => {  
 this.$modal.msgSuccess("修改成功");  
 this.open = false;  
 this.getList();  
 });  
 } else {  
 addModelDevice(this.form).then(response => {  
 this.$modal.msgSuccess("新增成功");  
 this.open = false;  
 this.getList();  
 });  
 }  
 }  
 });  
 },  
 /\*\* 删除按钮操作 \*/  
 handleDelete(row) {  
 const deviceIds = row.deviceId || this.ids;  
 this.$modal.confirm('是否确认删除装置编号为"' + deviceIds + '"的数据项？').then(function() {  
 return delModelDevice(deviceIds);  
 }).then(() => {  
 this.getList();  
 this.$modal.msgSuccess("删除成功");  
 }).catch(() => {});  
 },  
 /\*\* 导出按钮操作 \*/  
 handleExport() {  
 this.download('asm/modelDevice/export', {  
 ...this.queryParams  
 }, `modelDevice\_${new Date().getTime()}.xlsx`)  
 }  
 }  
};  
</script>

<template>

<div class="app-container">

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

<el-form-item label="中文名称" prop="cnName">

<el-input

v-model="queryParams.cnName"

placeholder="请输入中文名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="英文名称" prop="enName">

<el-input

v-model="queryParams.enName"

placeholder="请输入英文名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="评价日期">

<el-date-picker

v-model="daterangeEvaluateTime"

style="width: 240px"

value-format="yyyy-MM-dd"

type="daterange"

range-separator="-"

start-placeholder="开始日期"

end-placeholder="结束日期"

></el-date-picker>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input

v-model="queryParams.statisticalYear"

placeholder="请输入统计年份"

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"

v-hasPermi="['asm:oil\_exploitation:add']"

>新增

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:oil\_exploitation:edit']"

>修改

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:oil\_exploitation:remove']"

>删除

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-upload2"

size="mini"

@click="handleImport"

v-hasPermi="['asm:oil\_exploitation:import']"

>导入

</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:oil\_exploitation:export']"

>导出

</el-button>

</el-col>

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

</el-row>

<el-table v-loading="loading" :data="oil\_exploitationList" @selection-change="handleSelectionChange">

<el-table-column type="selection" width="55" align="center"/>

<!--<el-table-column label="ID" align="center" prop="uuid" />-->

<el-table-column label="序号" align="center" prop="orderNum" width="60"/>

<el-table-column label="中文名称" align="center" prop="cnName"/>

<el-table-column label="英文名称" align="center" prop="enName"/>

<el-table-column label="出产国家" align="center" prop="productCountry"/>

<el-table-column label="地区" align="center" prop="area"/>

<el-table-column label="评价日期" align="center" prop="evaluateTime" width="160">

<template slot-scope="scope">

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

</template>

</el-table-column>

<el-table-column label="评价API" align="center" prop="evaluateApi"/>

<el-table-column label="评价硫含量" align="center" prop="evaluateSulfur"/>

<el-table-column label="评价酸值" align="center" prop="evaluateAcid"/>

<el-table-column label="统计年份" align="center" prop="statisticalYear"/>

<el-table-column label="碳排放强度" align="center" prop="carbonEmissionIntensity"/>

<el-table-column label="备注" align="center" prop="remark"/>

<el-table-column label="创建者" align="center" prop="createBy"/>

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:oil\_exploitation:edit']"

>修改

</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:oil\_exploitation:remove']"

>删除

</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改原油开采端对话框 -->

<el-dialog :title="title" :visible.sync="open" width="70%" append-to-body>

<el-form ref="form" :model="form" :rules="rules" label-width="110px">

<el-row>

<el-col :span="11">

<el-form-item label="中文名称" prop="cnName">

<el-input v-model="form.cnName" placeholder="请输入中文名称"/>

</el-form-item>

</el-col>

<el-col :span="11">

<el-form-item label="英文名称" prop="enName">

<el-input v-model="form.enName" placeholder="请输入英文名称"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="11">

<el-form-item label="出产国家" prop="productCountry">

<el-input v-model="form.productCountry" placeholder="请输入出产国家"/>

</el-form-item>

</el-col>

<el-col :span="11">

<el-form-item label="地区" prop="area">

<el-input v-model="form.area" placeholder="请输入地区"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="11">

<el-form-item label="评价日期" prop="evaluateTime">

<el-date-picker clearable

v-model="form.evaluateTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择评价日期">

</el-date-picker>

</el-form-item>

</el-col>

<el-col :span="11">

<el-form-item label="评价API" prop="evaluateApi">

<el-input v-model="form.evaluateApi" placeholder="请输入评价API"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="11">

<el-form-item label="评价硫含量" prop="evaluateSulfur">

<el-input v-model="form.evaluateSulfur" placeholder="请输入评价硫含量"/>

</el-form-item>

</el-col>

<el-col :span="11">

<el-form-item label="评价酸值" prop="evaluateAcid">

<el-input v-model="form.evaluateAcid" placeholder="请输入评价酸值"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="11">

<el-form-item label="统计年份" prop="statisticalYear">

<el-input v-model="form.statisticalYear" placeholder="请输入统计年份"/>

</el-form-item>

</el-col>

<el-col :span="11">

<el-form-item label="碳排放强度" prop="carbonEmissionIntensity">

<el-input v-model="form.carbonEmissionIntensity" placeholder="请输入碳排放强度"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="11">

<el-form-item label="序号" prop="orderNum">

<el-input-number v-model="form.orderNum" placeholder="请输入排序" controls-position="right" :min="0"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-form-item label="备注" prop="remark">

<el-input v-model="form.remark" type="textarea" style="width: 90%" placeholder="请输入内容"/>

</el-form-item>

</el-row>

</el-form>

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

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

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

</div>

</el-dialog>

<!-- 导入对话框 -->

<el-dialog :title="upload.title" :visible.sync="upload.open" width="400px" append-to-body>

<el-upload

ref="upload"

:limit="1"

accept=".xlsx, .xls"

:headers="upload.headers"

:action="upload.url + '?updateSupport=' + upload.updateSupport"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="false"

drag

>

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

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

<div class="el-upload\_\_tip text-center" slot="tip">

<div class="el-upload\_\_tip" slot="tip">

<el-checkbox v-model="upload.updateSupport"/>

是否更新已经存在的数据

</div>

<span>仅允许导入xls、xlsx格式文件。</span>

<el-link type="primary" :underline="false" style="font-size:12px;vertical-align: baseline;" @click="importTemplate">下载模板</el-link>

</div>

</el-upload>

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

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

<el-button @click="upload.open = false">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import {

listOil\_exploitation,

getOil\_exploitation,

delOil\_exploitation,

addOil\_exploitation,

updateOil\_exploitation

} from "@/api/asm/oil\_exploitation";

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

export default {

name: "Oil\_exploitation",

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 原油开采端表格数据

oil\_exploitationList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 创建人id时间范围

daterangeEvaluateTime: [],

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

cnName: null,

enName: null,

productCountryId: null,

areaId: null,

evaluateTime: null,

evaluateApi: null,

statisticalYear: null

},

// 表单参数

form: {},

// 表单校验

rules: {},

// 导入参数

upload: {

// 是否显示弹出层

open: false,

// 弹出层标题

title: "",

// 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: "Bearer " + getToken()},

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/asm/oil\_exploitation/importData"

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询原油开采端列表 \*/

getList() {

this.loading = true;

this.queryParams.params = {};

if (null != this.daterangeEvaluateTime && '' != this.daterangeEvaluateTime) {

this.queryParams.params["beginEvaluateTime"] = this.daterangeEvaluateTime[0];

this.queryParams.params["endEvaluateTime"] = this.daterangeEvaluateTime[1];

}

listOil\_exploitation(this.queryParams).then(response => {

this.oil\_exploitationList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

uuid: null,

orderNum: null,

cnName: null,

enName: null,

productCountryId: null,

areaId: null,

evaluateTime: null,

evaluateApi: null,

evaluateSulfur: null,

evaluateAcid: null,

statisticalYear: null,

carbonEmissionIntensity: null,

remark: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.daterangeEvaluateTime = [];

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length !== 1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加原油开采端";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const uuid = row.uuid || this.ids

getOil\_exploitation(uuid).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改原油开采端";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.uuid != null) {

updateOil\_exploitation(this.form).then(response => {

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

addOil\_exploitation(this.form).then(response => {

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const uuids = row.uuid || this.ids;

this.$modal.confirm('是否确认删除原油开采端编号为"' + uuids + '"的数据项？').then(function () {

return delOil\_exploitation(uuids);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {

});

},

/\*\* 导出按钮操作 \*/

handleExport() {

this.download('asm/oil\_exploitation/export', {

...this.queryParams

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

},

/\*\* 导入按钮操作 \*/

handleImport() {

this.upload.title = "导入";

this.upload.open = true;

},

/\*\* 下载模板操作 \*/

importTemplate() {

this.download('asm/oil\_exploitation/importTemplate', {}, `oil\_exploitation\_template\_${new Date().getTime()}.xlsx`)

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

},

// 文件上传成功处理

handleFileSuccess(response, file, fileList) {

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

this.$alert("<div style='overflow: auto;overflow-x: hidden;max-height: 70vh;padding: 10px 20px 0;'>" + response.msg + "</div>", "导入结果", {dangerouslyUseHTMLString: true});

this.getList();

},

// 提交上传文件

submitFileForm() {

this.$refs.upload.submit();

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="中文名称" prop="cnName">

<el-input

v-model="queryParams.cnName"

placeholder="请输入原油中文名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="英文名称" prop="enName">

<el-input

v-model="queryParams.enName"

placeholder="请输入原油英文名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="评价日期" prop="evaluateTime">

<el-date-picker clearable

v-model="queryParams.evaluateTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择评价数据日期">

</el-date-picker>

</el-form-item>

<el-form-item label="评价API" prop="evaluateApi">

<el-input

v-model="queryParams.evaluateApi"

placeholder="请输入评价API"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="运输方式" prop="transportTypeKey">

<el-select v-model="queryParams.transportTypeKey" placeholder="请选择运输方式" clearable>

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input

v-model="queryParams.statisticalYear"

placeholder="请输入运输端碳排放统计年份"

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"

v-hasPermi="['asm:oil\_transport:add']"

>新增

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:oil\_transport:edit']"

>修改

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:oil\_transport:remove']"

>删除

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-upload2"

size="mini"

@click="handleImport"

v-hasPermi="['asm:oil\_transport:import']"

>导入

</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:oil\_transport:export']"

>导出

</el-button>

</el-col>

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

</el-row>

<el-table v-loading="loading" :data="oil\_transportList" @selection-change="handleSelectionChange">

<el-table-column type="selection" width="55" align="center"/>

<el-table-column label="序号" align="center" prop="orderNum" width="60"/>

<el-table-column label="原油中文名称" align="center" prop="cnName"/>

<el-table-column label="原油英文名称" align="center" prop="enName"/>

<el-table-column label="出产国家" align="center" prop="productCountry"/>

<el-table-column label="地区" align="center" prop="area"/>

<el-table-column label="评价数据日期" align="center" prop="evaluateTime" width="160">

<template slot-scope="scope">

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

</template>

</el-table-column>

<el-table-column label="评价API" align="center" prop="evaluateApi"/>

<el-table-column label="评价硫含量" align="center" prop="evaluateSulfur"/>

<el-table-column label="评价酸值" align="center" prop="evaluateAcid"/>

<el-table-column label="运输方式" align="center" prop="transportTypeKey">

<template slot-scope="scope">

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

</template>

</el-table-column>

<el-table-column label="统计年份" align="center" prop="statisticalYear"/>

<el-table-column label="碳排放强度" align="center" prop="carbonEmissionIntensity"/>

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:oil\_transport:edit']"

>修改

</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:oil\_transport:remove']"

>删除

</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改原油运输端对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="序号" prop="orderNum">

<el-input-number v-model="form.orderNum" placeholder="请输入排序" controls-position="right" :min="0"/>

</el-form-item>

<el-form-item label="中文名称" prop="cnName">

<el-input v-model="form.cnName" placeholder="请输入原油中文名称"/>

</el-form-item>

<el-form-item label="英文名称" prop="enName">

<el-input v-model="form.enName" placeholder="请输入原油英文名称"/>

</el-form-item>

<el-form-item label="出产国家" prop="productCountryId">

<el-input v-model="form.productCountry" placeholder="请输入出产国家"/>

</el-form-item>

<el-form-item label="地区" prop="areaId">

<el-input v-model="form.area" placeholder="请输入地区"/>

</el-form-item>

<el-form-item label="评价日期" prop="evaluateTime">

<el-date-picker clearable

v-model="form.evaluateTime"

type="date"

value-format="yyyy-MM-dd"

placeholder="请选择评价数据日期">

</el-date-picker>

</el-form-item>

<el-form-item label="评价API" prop="evaluateApi">

<el-input v-model="form.evaluateApi" placeholder="请输入评价API"/>

</el-form-item>

<el-form-item label="评价硫含量" prop="evaluateSulfur">

<el-input v-model="form.evaluateSulfur" placeholder="请输入评价硫含量"/>

</el-form-item>

<el-form-item label="评价酸值" prop="evaluateAcid">

<el-input v-model="form.evaluateAcid" placeholder="请输入评价酸值"/>

</el-form-item>

<el-form-item label="运输方式" prop="transportTypeKey">

<el-select v-model="form.transportTypeKey" placeholder="请选择运输方式">

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="碳排放统计年份" prop="statisticalYear">

<el-input v-model="form.statisticalYear" placeholder="请输入运输端碳排放统计年份"/>

</el-form-item>

<el-form-item label="碳排放强度" prop="carbonEmissionIntensity">

<el-input v-model="form.carbonEmissionIntensity" placeholder="请输入运输端碳排放强度"/>

</el-form-item>

<el-form-item label="备注" prop="remark">

<el-input v-model="form.remark" type="textarea" placeholder="请输入内容"/>

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

<!-- 导入对话框 -->

<el-dialog :title="upload.title" :visible.sync="upload.open" width="400px" append-to-body>

<el-upload

ref="upload"

:limit="1"

accept=".xlsx, .xls"

:headers="upload.headers"

:action="upload.url + '?updateSupport=' + upload.updateSupport"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="false"

drag

>

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

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

<div class="el-upload\_\_tip text-center" slot="tip">

<!--<div class="el-upload\_\_tip" slot="tip">

<el-checkbox v-model="upload.updateSupport"/>

是否更新已经存在的数据

</div>-->

<span>仅允许导入xls、xlsx格式文件。</span>

<el-link type="primary" :underline="false" style="font-size:12px;vertical-align: baseline;" @click="importTemplate">下载模板</el-link>

</div>

</el-upload>

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

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

<el-button @click="upload.open = false">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import {listOil\_transport, getOil\_transport, delOil\_transport, addOil\_transport, updateOil\_transport} from "@/api/asm/oil\_transport";

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

export default {

name: "Oil\_transport",

dicts: ['transport\_type'],

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 原油运输端表格数据

oil\_transportList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

cnName: null,

enName: null,

productCountryId: null,

areaId: null,

evaluateTime: null,

evaluateApi: null,

transportTypeKey: null,

statisticalYear: null

},

// 表单参数

form: {},

// 表单校验

rules: {},

// 导入参数

upload: {

// 是否显示弹出层

open: false,

// 弹出层标题

title: "",

// 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: "Bearer " + getToken()},

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/asm/oil\_transport/importData"

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询原油运输端列表 \*/

getList() {

this.loading = true;

listOil\_transport(this.queryParams).then(response => {

this.oil\_transportList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

uuid: null,

orderNum: null,

cnName: null,

enName: null,

productCountry: null,

area: null,

productCountryId: null,

areaId: null,

evaluateTime: null,

evaluateApi: null,

evaluateSulfur: null,

evaluateAcid: null,

transportTypeKey: null,

transportTypeLable: null,

statisticalYear: null,

carbonEmissionIntensity: null,

remark: null,

createUser: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length !== 1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加原油运输端";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const uuid = row.uuid || this.ids

getOil\_transport(uuid).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改原油运输端";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.uuid != null) {

updateOil\_transport(this.form).then(response => {

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

addOil\_transport(this.form).then(response => {

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const uuids = row.uuid || this.ids;

this.$modal.confirm('是否确认删除原油运输端编号为"' + uuids + '"的数据项？').then(function () {

return delOil\_transport(uuids);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {

});

},

/\*\* 导出按钮操作 \*/

handleExport() {

this.download('asm/oil\_transport/export', {

...this.queryParams

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

},

/\*\* 导入按钮操作 \*/

handleImport() {

this.upload.title = "导入";

this.upload.open = true;

},

/\*\* 下载模板操作 \*/

importTemplate() {

this.download('asm/oil\_transport/importTemplate', {}, `oil\_transport\_template\_${new Date().getTime()}.xlsx`)

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

},

// 文件上传成功处理

handleFileSuccess(response, file, fileList) {

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

this.$alert("<div style='overflow: auto;overflow-x: hidden;max-height: 70vh;padding: 10px 20px 0;'>" + response.msg + "</div>", "导入结果", {dangerouslyUseHTMLString: true});

this.getList();

},

// 提交上传文件

submitFileForm() {

this.$refs.upload.submit();

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="字典代码" prop="dictCode">

<el-input

v-model="queryParams.dictCode"

placeholder="请输入字典代码"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="字典名称" prop="dictName">

<el-input

v-model="queryParams.dictName"

placeholder="请输入字典名称"

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"

v-hasPermi="['asm:osblDict:add']"

>新增

</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-sort"

size="mini"

@click="toggleExpandAll"

>展开/折叠

</el-button>

</el-col>

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

</el-row>

<el-table

v-if="refreshTable"

v-loading="loading"

:data="osblDictList"

row-key="dictId"

:default-expand-all="isExpandAll"

:tree-props="{children: 'children', hasChildren: 'hasChildren'}"

>

<el-table-column label="字典代码" prop="dictCode"/>

<el-table-column label="字典名称" align="center" prop="dictName"/>

<el-table-column label="co2排放系数" align="center" prop="co2EmissionCoefficient"/>

<el-table-column label="co2排放系数单位" align="center" prop="co2EmissionCoefficientUnit"/>

<el-table-column label="折标系数" align="center" prop="foldCoefficient"/>

<el-table-column label="折标系数单位" align="center" prop="foldCoefficientUnit"/>

<el-table-column label="热值" align="center" prop="calorificValue"/>

<el-table-column label="热值单位" align="center" prop="calorificValueUnit"/>

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:osblDict:edit']"

>修改

</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-plus"

@click="handleAdd(scope.row)"

v-hasPermi="['asm:osblDict:add']"

>新增

</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:osblDict:remove']"

>删除

</el-button>

</template>

</el-table-column>

</el-table>

<!-- 添加或修改公用工程字典对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="字典代码" prop="dictCode">

<el-input v-model="form.dictCode" placeholder="请输入字典代码"/>

</el-form-item>

<el-form-item label="字典名称" prop="dictName">

<el-input v-model="form.dictName" placeholder="请输入字典名称"/>

</el-form-item>

<el-form-item label="co2排放系数" prop="co2EmissionCoefficient">

<el-input v-model="form.co2EmissionCoefficient" placeholder="请输入co2排放系数"/>

</el-form-item>

<el-form-item label="co2排放系数单位" prop="co2EmissionCoefficientUnit">

<el-input v-model="form.co2EmissionCoefficientUnit" placeholder="请输入co2排放系数单位"/>

</el-form-item>

<el-form-item label="折标系数" prop="foldCoefficient">

<el-input v-model="form.foldCoefficient" placeholder="请输入折标系数"/>

</el-form-item>

<el-form-item label="折标系数单位" prop="foldCoefficientUnit">

<el-input v-model="form.foldCoefficientUnit" placeholder="请输入折标系数单位"/>

</el-form-item>

<el-form-item label="热值" prop="calorificValue">

<el-input v-model="form.calorificValue" placeholder="请输入热值"/>

</el-form-item>

<el-form-item label="热值单位" prop="calorificValueUnit">

<el-input v-model="form.calorificValueUnit" placeholder="请输入热值单位"/>

</el-form-item>

<el-form-item label="备注" prop="describes">

<el-input v-model="form.describes" placeholder="请输入备注"/>

</el-form-item>

<el-form-item label="父id" prop="parentId">

<treeselect v-model="form.parentId" :options="osblDictOptions" :normalizer="normalizer" placeholder="请选择父id"/>

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

</div>

</template>

<script>

import {listOsblDict, getOsblDict, delOsblDict, addOsblDict, updateOsblDict} from "@/api/asm/osblDict";

import Treeselect from "@riophae/vue-treeselect";

import "@riophae/vue-treeselect/dist/vue-treeselect.css";

export default {

name: "OsblDict",

components: {

Treeselect

},

data() {

return {

// 遮罩层

loading: true,

// 显示搜索条件

showSearch: true,

// 公用工程字典表格数据

osblDictList: [],

// 公用工程字典树选项

osblDictOptions: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 是否展开，默认全部展开

isExpandAll: true,

// 重新渲染表格状态

refreshTable: true,

// 查询参数

queryParams: {

dictCode: null,

dictName: null

},

// 表单参数

form: {},

// 表单校验

rules: {

dictName: [

{required: true, message: "字典名称不能为空", trigger: "blur"}

],

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询公用工程字典列表 \*/

getList() {

this.loading = true;

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

this.osblDictList = this.handleTree(response.data, "dictId", "parentId");

this.loading = false;

});

},

/\*\* 转换公用工程字典数据结构 \*/

normalizer(node) {

if (node.children && !node.children.length) {

delete node.children;

}

return {

id: node.dictId,

label: node.dictName,

children: node.children

};

},

/\*\* 查询公用工程字典下拉树结构 \*/

getTreeselect() {

listOsblDict().then(response => {

this.osblDictOptions = [];

const data = {dictId: 0, dictName: '顶级节点', children: []};

data.children = this.handleTree(response.data, "dictId", "parentId");

this.osblDictOptions.push(data);

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

dictId: null,

dictCode: null,

dictName: null,

co2EmissionCoefficient: null,

co2EmissionCoefficientUnit: null,

foldCoefficient: null,

foldCoefficientUnit: null,

calorificValue: null,

calorificValueUnit: null,

describes: null,

parentId: 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(row) {

this.reset();

this.getTreeselect();

if (row != null && row.dictId) {

this.form.parentId = row.dictId;

} else {

this.form.parentId = 0;

}

this.open = true;

this.title = "添加公用工程字典";

},

/\*\* 展开/折叠操作 \*/

toggleExpandAll() {

this.refreshTable = false;

this.isExpandAll = !this.isExpandAll;

this.$nextTick(() => {

this.refreshTable = true;

});

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

this.getTreeselect();

if (row != null) {

this.form.parentId = row.dictId;

}

getOsblDict(row.dictId).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改公用工程字典";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

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

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

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

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

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

this.$modal.confirm('是否确认删除公用工程字典编号为"' + row.dictId + '"的数据项？').then(function () {

return delOsblDict(row.dictId);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {

});

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="产品名称" prop="cnName">

<el-input

v-model="queryParams.cnName"

placeholder="请输入产品名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="运输方式" prop="transportTypeKey">

<el-select v-model="queryParams.transportTypeKey" placeholder="请选择运输方式" clearable>

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input

v-model="queryParams.statisticalYear"

placeholder="请输入运输端碳排放统计年份"

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"

v-hasPermi="['asm:product\_transport:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:product\_transport:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:product\_transport:remove']"

>删除</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-upload2"

size="mini"

@click="handleImport"

v-hasPermi="['asm:product\_transport:import']"

>导入

</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:product\_transport:export']"

>导出</el-button>

</el-col>

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

</el-row>

<el-table v-loading="loading" :data="product\_transportList" @selection-change="handleSelectionChange">

<el-table-column type="selection" width="55" align="center" />

<!--<el-table-column label="ID" align="center" prop="uuid" />-->

<el-table-column label="序号" align="center" prop="orderNum" width="60"/>

<el-table-column label="产品名称" align="center" prop="cnName" />

<el-table-column label="运输方式" align="center" prop="transportTypeKey">

<template slot-scope="scope">

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

</template>

</el-table-column>

<el-table-column label="统计年份" align="center" prop="statisticalYear" />

<el-table-column label="碳排放强度" align="center" prop="carbonEmissionIntensity" />

<el-table-column label="备注" align="center" prop="remark" />

<!--<el-table-column label="创建者" align="center" prop="createBy" />-->

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:product\_transport:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:product\_transport:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改产品运输端对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="序号" prop="orderNum">

<el-input-number v-model="form.orderNum" placeholder="请输入排序" controls-position="right" :min="0"/>

</el-form-item>

<el-form-item label="产品名称" prop="cnName">

<el-input v-model="form.cnName" placeholder="请输入产品名称" />

</el-form-item>

<el-form-item label="运输方式" prop="transportTypeKey">

<el-select v-model="form.transportTypeKey" placeholder="请选择运输方式">

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input v-model="form.statisticalYear" placeholder="请输入运输端碳排放统计年份" />

</el-form-item>

<el-form-item label="碳排放强度" prop="carbonEmissionIntensity">

<el-input v-model="form.carbonEmissionIntensity" placeholder="请输入运输端碳排放强度" />

</el-form-item>

<el-form-item label="备注" prop="remark">

<el-input v-model="form.remark" type="textarea" placeholder="请输入内容" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

<!-- 导入对话框 -->

<el-dialog :title="upload.title" :visible.sync="upload.open" width="400px" append-to-body>

<el-upload

ref="upload"

:limit="1"

accept=".xlsx, .xls"

:headers="upload.headers"

:action="upload.url + '?updateSupport=' + upload.updateSupport"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="false"

drag

>

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

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

<div class="el-upload\_\_tip text-center" slot="tip">

<!--<div class="el-upload\_\_tip" slot="tip">

<el-checkbox v-model="upload.updateSupport"/>

是否更新已经存在的数据

</div>-->

<span>仅允许导入xls、xlsx格式文件。</span>

<el-link type="primary" :underline="false" style="font-size:12px;vertical-align: baseline;" @click="importTemplate">下载模板</el-link>

</div>

</el-upload>

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

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

<el-button @click="upload.open = false">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import { listProduct\_transport, getProduct\_transport, delProduct\_transport, addProduct\_transport, updateProduct\_transport } from "@/api/asm/product\_transport";

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

export default {

name: "Product\_transport",

dicts: ['transport\_type'],

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 产品运输端表格数据

product\_transportList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

cnName: null,

transportTypeKey: null,

statisticalYear: null,

},

// 表单参数

form: {},

// 表单校验

rules: {

},

// 导入参数

upload: {

// 是否显示弹出层

open: false,

// 弹出层标题

title: "",

// 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: "Bearer " + getToken()},

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/asm/product\_transport/importData"

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询产品运输端列表 \*/

getList() {

this.loading = true;

listProduct\_transport(this.queryParams).then(response => {

this.product\_transportList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

uuid: null,

orderNum: null,

cnName: null,

transportTypeKey: null,

transportTypeLable: null,

statisticalYear: null,

carbonEmissionIntensity: null,

remark: null,

createUser: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加产品运输端";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const uuid = row.uuid || this.ids

getProduct\_transport(uuid).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改产品运输端";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.uuid != null) {

updateProduct\_transport(this.form).then(response => {

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

addProduct\_transport(this.form).then(response => {

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const uuids = row.uuid || this.ids;

this.$modal.confirm('是否确认删除产品运输端编号为"' + uuids + '"的数据项？').then(function() {

return delProduct\_transport(uuids);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

this.download('asm/product\_transport/export', {

...this.queryParams

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

},

/\*\* 导入按钮操作 \*/

handleImport() {

this.upload.title = "导入";

this.upload.open = true;

},

/\*\* 下载模板操作 \*/

importTemplate() {

this.download('asm/product\_transport/importTemplate', {}, `product\_transport\_template\_${new Date().getTime()}.xlsx`)

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

},

// 文件上传成功处理

handleFileSuccess(response, file, fileList) {

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

this.$alert("<div style='overflow: auto;overflow-x: hidden;max-height: 70vh;padding: 10px 20px 0;'>" + response.msg + "</div>", "导入结果", {dangerouslyUseHTMLString: true});

this.getList();

},

// 提交上传文件

submitFileForm() {

this.$refs.upload.submit();

}

}

};

</script>

<template>

<div class="app-container">

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

<el-form-item label="产品名称" prop="proName">

<el-input

v-model="queryParams.proName"

placeholder="请输入产品名称"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

<el-form-item label="产品种类" prop="proTypeKey">

<el-select v-model="queryParams.proTypeKey" placeholder="请选择产品种类" clearable>

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="使用方式" prop="useTypeKey">

<el-select v-model="queryParams.useTypeKey" placeholder="请选择使用方式" clearable>

<el-option

v-for="dict in dict.type.product\_use\_type"

:key="dict.value"

:label="dict.label"

:value="dict.value"

/>

</el-select>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input

v-model="queryParams.statisticalYear"

placeholder="请输入使用端碳排放统计年份"

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"

v-hasPermi="['asm:product\_use:add']"

>新增</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="success"

plain

icon="el-icon-edit"

size="mini"

:disabled="single"

@click="handleUpdate"

v-hasPermi="['asm:product\_use:edit']"

>修改</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="danger"

plain

icon="el-icon-delete"

size="mini"

:disabled="multiple"

@click="handleDelete"

v-hasPermi="['asm:product\_use:remove']"

>删除</el-button>

</el-col>

<el-col :span="1.5">

<el-button

type="info"

plain

icon="el-icon-upload2"

size="mini"

@click="handleImport"

v-hasPermi="['asm:product\_use:import']"

>导入

</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:product\_use:export']"

>导出</el-button>

</el-col>

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

</el-row>

<el-table v-loading="loading" :data="product\_useList" @selection-change="handleSelectionChange">

<el-table-column type="selection" width="55" align="center" />

<!--<el-table-column label="ID" align="center" prop="uuid" />-->

<el-table-column label="序号" align="center" prop="orderNum" width="60"/>

<el-table-column label="产品名称" align="center" prop="proName" />

<el-table-column label="产品种类" align="center" prop="proTypeKey">

<template slot-scope="scope">

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

</template>

</el-table-column>

<el-table-column label="使用方式" align="center" prop="useTypeKey">

<template slot-scope="scope">

<dict-tag :options="dict.type.product\_use\_type" :value="scope.row.useTypeKey"/>

</template>

</el-table-column>

<el-table-column label="统计年份" align="center" prop="statisticalYear" />

<el-table-column label="碳排放强度" align="center" prop="carbonEmissionIntensity" />

<el-table-column label="备注" align="center" prop="remark" />

<!--<el-table-column label="创建者" align="center" prop="createBy" />-->

<el-table-column label="操作" align="center" class-name="small-padding fixed-width">

<template slot-scope="scope">

<el-button

size="mini"

type="text"

icon="el-icon-edit"

@click="handleUpdate(scope.row)"

v-hasPermi="['asm:product\_use:edit']"

>修改</el-button>

<el-button

size="mini"

type="text"

icon="el-icon-delete"

@click="handleDelete(scope.row)"

v-hasPermi="['asm:product\_use:remove']"

>删除</el-button>

</template>

</el-table-column>

</el-table>

<pagination

v-show="total>0"

:total="total"

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

<!-- 添加或修改产品使用端对话框 -->

<el-dialog :title="title" :visible.sync="open" width="500px" append-to-body>

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

<el-form-item label="序号" prop="orderNum">

<el-input-number v-model="form.orderNum" placeholder="请输入排序" controls-position="right" :min="0"/>

</el-form-item>

<el-form-item label="产品名称" prop="proName">

<el-input v-model="form.proName" placeholder="请输入产品名称" />

</el-form-item>

<el-form-item label="产品种类" prop="proTypeKey">

<el-select v-model="form.proTypeKey" placeholder="请选择产品种类">

<el-option

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

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="使用方式" prop="useTypeKey">

<el-select v-model="form.useTypeKey" placeholder="请选择使用方式">

<el-option

v-for="dict in dict.type.product\_use\_type"

:key="dict.value"

:label="dict.label"

:value="dict.value"

></el-option>

</el-select>

</el-form-item>

<el-form-item label="统计年份" prop="statisticalYear">

<el-input v-model="form.statisticalYear" placeholder="请输入使用端碳排放统计年份" />

</el-form-item>

<el-form-item label="碳排放强度" prop="carbonEmissionIntensity">

<el-input v-model="form.carbonEmissionIntensity" placeholder="请输入使用端碳排放强度" />

</el-form-item>

<el-form-item label="备注" prop="remark">

<el-input v-model="form.remark" type="textarea" placeholder="请输入内容" />

</el-form-item>

</el-form>

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

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

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

</div>

</el-dialog>

<!-- 导入对话框 -->

<el-dialog :title="upload.title" :visible.sync="upload.open" width="400px" append-to-body>

<el-upload

ref="upload"

:limit="1"

accept=".xlsx, .xls"

:headers="upload.headers"

:action="upload.url + '?updateSupport=' + upload.updateSupport"

:disabled="upload.isUploading"

:on-progress="handleFileUploadProgress"

:on-success="handleFileSuccess"

:auto-upload="false"

drag

>

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

<div class="el-upload\_\_text">将文件拖到此处，或<em>点击上传</em></div>

<div class="el-upload\_\_tip text-center" slot="tip">

<!--<div class="el-upload\_\_tip" slot="tip">

<el-checkbox v-model="upload.updateSupport"/>

是否更新已经存在的数据

</div>-->

<span>仅允许导入xls、xlsx格式文件。</span>

<el-link type="primary" :underline="false" style="font-size:12px;vertical-align: baseline;" @click="importTemplate">下载模板</el-link>

</div>

</el-upload>

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

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

<el-button @click="upload.open = false">取 消</el-button>

</div>

</el-dialog>

</div>

</template>

<script>

import { listProduct\_use, getProduct\_use, delProduct\_use, addProduct\_use, updateProduct\_use } from "@/api/asm/product\_use";

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

export default {

name: "Product\_use",

dicts: ['product\_type', 'product\_use\_type'],

data() {

return {

// 遮罩层

loading: true,

// 选中数组

ids: [],

// 非单个禁用

single: true,

// 非多个禁用

multiple: true,

// 显示搜索条件

showSearch: true,

// 总条数

total: 0,

// 产品使用端表格数据

product\_useList: [],

// 弹出层标题

title: "",

// 是否显示弹出层

open: false,

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

proName: null,

proTypeKey: null,

useTypeKey: null,

statisticalYear: null

},

// 表单参数

form: {},

// 表单校验

rules: {

},

// 导入参数

upload: {

// 是否显示弹出层

open: false,

// 弹出层标题

title: "",

// 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: "Bearer " + getToken()},

// 上传的地址

url: process.env.VUE\_APP\_BASE\_API + "/asm/product\_use/importData"

}

};

},

created() {

this.getList();

},

methods: {

/\*\* 查询产品使用端列表 \*/

getList() {

this.loading = true;

listProduct\_use(this.queryParams).then(response => {

this.product\_useList = response.rows;

this.total = response.total;

this.loading = false;

});

},

// 取消按钮

cancel() {

this.open = false;

this.reset();

},

// 表单重置

reset() {

this.form = {

uuid: null,

orderNum: null,

proName: null,

proTypeLable: null,

proTypeKey: null,

useTypeLable: null,

useTypeKey: null,

statisticalYear: null,

carbonEmissionIntensity: null,

remark: null,

createUser: null,

createBy: null,

createTime: null

};

this.resetForm("form");

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

this.queryParams.pageNum = 1;

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

// 多选框选中数据

handleSelectionChange(selection) {

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

this.single = selection.length!==1

this.multiple = !selection.length

},

/\*\* 新增按钮操作 \*/

handleAdd() {

this.reset();

this.open = true;

this.title = "添加产品使用端";

},

/\*\* 修改按钮操作 \*/

handleUpdate(row) {

this.reset();

const uuid = row.uuid || this.ids

getProduct\_use(uuid).then(response => {

this.form = response.data;

this.open = true;

this.title = "修改产品使用端";

});

},

/\*\* 提交按钮 \*/

submitForm() {

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

if (valid) {

if (this.form.uuid != null) {

updateProduct\_use(this.form).then(response => {

this.$modal.msgSuccess("修改成功");

this.open = false;

this.getList();

});

} else {

addProduct\_use(this.form).then(response => {

this.$modal.msgSuccess("新增成功");

this.open = false;

this.getList();

});

}

}

});

},

/\*\* 删除按钮操作 \*/

handleDelete(row) {

const uuids = row.uuid || this.ids;

this.$modal.confirm('是否确认删除产品使用端编号为"' + uuids + '"的数据项？').then(function() {

return delProduct\_use(uuids);

}).then(() => {

this.getList();

this.$modal.msgSuccess("删除成功");

}).catch(() => {});

},

/\*\* 导出按钮操作 \*/

handleExport() {

this.download('asm/product\_use/export', {

...this.queryParams

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

},

/\*\* 导入按钮操作 \*/

handleImport() {

this.upload.title = "导入";

this.upload.open = true;

},

/\*\* 下载模板操作 \*/

importTemplate() {

this.download('asm/product\_use/importTemplate', {}, `product\_use\_template\_${new Date().getTime()}.xlsx`)

},

// 文件上传中处理

handleFileUploadProgress(event, file, fileList) {

this.upload.isUploading = true;

},

// 文件上传成功处理

handleFileSuccess(response, file, fileList) {

this.upload.open = false;

this.upload.isUploading = false;

this.$refs.upload.clearFiles();

this.$alert("<div style='overflow: auto;overflow-x: hidden;max-height: 70vh;padding: 10px 20px 0;'>" + response.msg + "</div>", "导入结果", {dangerouslyUseHTMLString: true});

this.getList();

},

// 提交上传文件

submitFileForm() {

this.$refs.upload.submit();

}

}

};

</script>

### Java:

package com.jiexun.asm.controller;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import com.jiexun.common.utils.DateUtils;

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.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.asm.domain.AsmCrudeOilExploitation;

import com.jiexun.asm.service.IAsmCrudeOilExploitationService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* 原油开采端Controller

\* @author yzc

\* @date 2022-07-20

\*/

@RestController

@RequestMapping("/asm/oil\_exploitation")

public class AsmCrudeOilExploitationController extends BaseController {

@Autowired

private IAsmCrudeOilExploitationService asmCrudeOilExploitationService;

/\*\*

\* 查询原油开采端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:list')")

@GetMapping("/list")

public TableDataInfo list(AsmCrudeOilExploitation asmCrudeOilExploitation) {

startPage();

List<AsmCrudeOilExploitation> list = asmCrudeOilExploitationService.selectAsmCrudeOilExploitationList(asmCrudeOilExploitation);

return getDataTable(list);

}

/\*\*

\* 导入数据

\* @param file

\* @param updateSupport 替换已存在数据（未使用）

\* @return

\* @throws Exception

\*/

@Log(title = "原油开采端", businessType = BusinessType.IMPORT)

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:import')")

@PostMapping("/importData")

public AjaxResult importData(MultipartFile file, boolean updateSupport) throws Exception {

ExcelUtil<AsmCrudeOilExploitation> util = new ExcelUtil<AsmCrudeOilExploitation>(AsmCrudeOilExploitation.class);

List<AsmCrudeOilExploitation> dataList = util.importExcel(file.getInputStream());

String operName = getUsername();

Long operId = getUserId();

String message = asmCrudeOilExploitationService.importData(dataList, updateSupport, operName, operId);

return AjaxResult.success(message);

}

@PostMapping("/importTemplate")

public void importTemplate(HttpServletResponse response) {

ExcelUtil<AsmCrudeOilExploitation> util = new ExcelUtil<AsmCrudeOilExploitation>(AsmCrudeOilExploitation.class);

util.importTemplateExcel(response, "原油开采端模版数据");

}

/\*\*

\* 导出原油开采端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:export')")

@Log(title = "原油开采端", businessType = BusinessType.EXPORT)

@PostMapping("/export")

public void export(HttpServletResponse response, AsmCrudeOilExploitation asmCrudeOilExploitation) {

List<AsmCrudeOilExploitation> list = asmCrudeOilExploitationService.selectAsmCrudeOilExploitationList(asmCrudeOilExploitation);

list.forEach(t -> {

if (null != t.getEvaluateAcid()) {

t.setEvaluateAcidStr(String.valueOf(t.getEvaluateAcid()));

}

if (null != t.getEvaluateSulfur()) {

t.setEvaluateSulfurStr(String.valueOf(t.getEvaluateSulfur()));

}

if (null != t.getCarbonEmissionIntensity()) {

t.setCarbonEmissionIntensityStr(String.valueOf(t.getCarbonEmissionIntensity()));

}

if (null != t.getEvaluateTime()) {

t.setEvaluateTimeStr(DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD, t.getEvaluateTime()));

}

if (null != t.getRemark()) {

t.setRemarkInfo(t.getRemark());

}

});

ExcelUtil<AsmCrudeOilExploitation> util = new ExcelUtil<AsmCrudeOilExploitation>(AsmCrudeOilExploitation.class);

util.exportExcel(response, list, "原油开采端导出数据");

}

/\*\*

\* 获取原油开采端详细信息

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:query')")

@GetMapping(value = "/{uuid}")

public AjaxResult getInfo(@PathVariable("uuid") String uuid) {

return AjaxResult.success(asmCrudeOilExploitationService.selectAsmCrudeOilExploitationByUuid(uuid));

}

/\*\*

\* 新增原油开采端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:add')")

@Log(title = "原油开采端", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody AsmCrudeOilExploitation asmCrudeOilExploitation) {

asmCrudeOilExploitation.setCreateUser(getUserId());

asmCrudeOilExploitation.setCreateBy(getUsername());

return toAjax(asmCrudeOilExploitationService.insertAsmCrudeOilExploitation(asmCrudeOilExploitation));

}

/\*\*

\* 修改原油开采端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:edit')")

@Log(title = "原油开采端", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody AsmCrudeOilExploitation asmCrudeOilExploitation) {

return toAjax(asmCrudeOilExploitationService.updateAsmCrudeOilExploitation(asmCrudeOilExploitation));

}

/\*\*

\* 删除原油开采端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_exploitation:remove')")

@Log(title = "原油开采端", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult remove(@PathVariable String[] uuids) {

return toAjax(asmCrudeOilExploitationService.deleteAsmCrudeOilExploitationByUuids(uuids));

}

}

package com.jiexun.asm.controller;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import com.jiexun.common.utils.DateUtils;

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.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.asm.domain.AsmCrudeOilTransport;

import com.jiexun.asm.service.IAsmCrudeOilTransportService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* 原油运输端Controller

\* @author yzc

\* @date 2022-07-20

\*/

@RestController

@RequestMapping("/asm/oil\_transport")

public class AsmCrudeOilTransportController extends BaseController {

@Autowired

private IAsmCrudeOilTransportService asmCrudeOilTransportService;

/\*\*

\* 查询原油运输端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:list')")

@GetMapping("/list")

public TableDataInfo list(AsmCrudeOilTransport asmCrudeOilTransport) {

startPage();

List<AsmCrudeOilTransport> list = asmCrudeOilTransportService.selectAsmCrudeOilTransportList(asmCrudeOilTransport);

return getDataTable(list);

}

/\*\*

\* 导入数据

\* @param file

\* @param updateSupport 替换已存在数据（未使用）

\* @return

\* @throws Exception

\*/

@Log(title = "原油运输端", businessType = BusinessType.IMPORT)

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:import')")

@PostMapping("/importData")

public AjaxResult importData(MultipartFile file, boolean updateSupport) throws Exception {

ExcelUtil<AsmCrudeOilTransport> util = new ExcelUtil<AsmCrudeOilTransport>(AsmCrudeOilTransport.class);

List<AsmCrudeOilTransport> dataList = util.importExcel(file.getInputStream());

String operName = getUsername();

Long operId = getUserId();

String message = asmCrudeOilTransportService.importData(dataList, updateSupport, operName, operId);

return AjaxResult.success(message);

}

@PostMapping("/importTemplate")

public void importTemplate(HttpServletResponse response) {

ExcelUtil<AsmCrudeOilTransport> util = new ExcelUtil<AsmCrudeOilTransport>(AsmCrudeOilTransport.class);

util.importTemplateExcel(response, "原油运输端模版数据");

}

/\*\*

\* 导出原油运输端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:export')")

@Log(title = "原油运输端", businessType = BusinessType.EXPORT)

@PostMapping("/export")

public void export(HttpServletResponse response, AsmCrudeOilTransport asmCrudeOilTransport) {

List<AsmCrudeOilTransport> list = asmCrudeOilTransportService.selectAsmCrudeOilTransportList(asmCrudeOilTransport);

list.forEach(t -> {

if (null != t.getEvaluateAcid()) {

t.setEvaluateAcidStr(String.valueOf(t.getEvaluateAcid()));

}

if (null != t.getEvaluateSulfur()) {

t.setEvaluateSulfurStr(String.valueOf(t.getEvaluateSulfur()));

}

if (null != t.getCarbonEmissionIntensity()) {

t.setCarbonEmissionIntensityStr(String.valueOf(t.getCarbonEmissionIntensity()));

}

if (null != t.getEvaluateTime()) {

t.setEvaluateTimeStr(DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD, t.getEvaluateTime()));

}

if (null != t.getRemark()) {

t.setRemarkInfo(t.getRemark());

}

});

ExcelUtil<AsmCrudeOilTransport> util = new ExcelUtil<AsmCrudeOilTransport>(AsmCrudeOilTransport.class);

util.exportExcel(response, list, "原油运输端导出数据");

}

/\*\*

\* 获取原油运输端详细信息

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:query')")

@GetMapping(value = "/{uuid}")

public AjaxResult getInfo(@PathVariable("uuid") String uuid) {

return AjaxResult.success(asmCrudeOilTransportService.selectAsmCrudeOilTransportByUuid(uuid));

}

/\*\*

\* 新增原油运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:add')")

@Log(title = "原油运输端", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody AsmCrudeOilTransport asmCrudeOilTransport) {

asmCrudeOilTransport.setCreateUser(getUserId());

asmCrudeOilTransport.setCreateBy(getUsername());

return toAjax(asmCrudeOilTransportService.insertAsmCrudeOilTransport(asmCrudeOilTransport));

}

/\*\*

\* 修改原油运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:edit')")

@Log(title = "原油运输端", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody AsmCrudeOilTransport asmCrudeOilTransport) {

return toAjax(asmCrudeOilTransportService.updateAsmCrudeOilTransport(asmCrudeOilTransport));

}

/\*\*

\* 删除原油运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:oil\_transport:remove')")

@Log(title = "原油运输端", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult remove(@PathVariable String[] uuids) {

return toAjax(asmCrudeOilTransportService.deleteAsmCrudeOilTransportByUuids(uuids));

}

}

package com.jiexun.asm.controller;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import com.jiexun.asm.domain.\*;

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.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.asm.service.IAsmModelService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* 模型Controller

\* @author yzc

\* @date 2022-07-22

\*/

@RestController

@RequestMapping("/asm/model")

public class AsmModelController extends BaseController {

@Autowired

private IAsmModelService asmModelService;

/\*\*

\* 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);

}

/\*\*

\* 导出模型列表

\*/

@PreAuthorize("@ss.hasPermi('asm:model:export')")

@Log(title = "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");

}

/\*\*

\* 导入数据

\* @param file

\* @param updateSupport 替换已存在数据（未使用）

\* @return

\* @throws Exception

\*/

@Log(title = "模型", businessType = BusinessType.IMPORT)

@PreAuthorize("@ss.hasPermi('asm:model:import')")

@PostMapping("/importData")

public AjaxResult importData(MultipartFile file, Boolean updateSupport) throws Exception {

try {

ExcelUtil<AsmModel> modelUtil = new ExcelUtil<AsmModel>(AsmModel.class);

List<AsmModel> modelList = modelUtil.importExcel("模型表", file.getInputStream(), 0);

ExcelUtil<AsmModelBlend> modelBlendUtil = new ExcelUtil<AsmModelBlend>(AsmModelBlend.class);

List<AsmModelBlend> modelBlendList = modelBlendUtil.importExcel("油品调和表", file.getInputStream(), 0);

ExcelUtil<AsmModelMaterialBalance> modelBalanceUtil = new ExcelUtil<AsmModelMaterialBalance>(AsmModelMaterialBalance.class);

List<AsmModelMaterialBalance> modelBalanceList = modelBalanceUtil.importExcel("物料平衡表", file.getInputStream(), 0);

ExcelUtil<AsmModelDevice> deviceUtil = new ExcelUtil<AsmModelDevice>(AsmModelDevice.class);

List<AsmModelDevice> deviceList = deviceUtil.importExcel("装置表", file.getInputStream(), 0);

ExcelUtil<AsmModelDeviceInput> deviceInputUtil = new ExcelUtil<AsmModelDeviceInput>(AsmModelDeviceInput.class);

List<AsmModelDeviceInput> deviceInputList = deviceInputUtil.importExcel("装置投入物料表", file.getInputStream(), 0);

ExcelUtil<AsmModelDeviceOsbl> deviceOsblUtil = new ExcelUtil<AsmModelDeviceOsbl>(AsmModelDeviceOsbl.class);

List<AsmModelDeviceOsbl> deviceOsblList = deviceOsblUtil.importExcel("装置消耗表", file.getInputStream(), 0);

ExcelUtil<AsmModelDeviceOutput> deviceOutputUtil = new ExcelUtil<AsmModelDeviceOutput>(AsmModelDeviceOutput.class);

List<AsmModelDeviceOutput> deviceOutputList = deviceOutputUtil.importExcel("装置产出物料表", file.getInputStream(), 0);

String loginUserName = getUsername();

Long loginUserId = getUserId();

asmModelService.importData(modelList, modelBlendList, modelBalanceList, deviceList, deviceInputList,

deviceOsblList, deviceOutputList, updateSupport, loginUserName, loginUserId);

return AjaxResult.success();

} catch (Exception e) {

e.printStackTrace();

return AjaxResult.error(e.getMessage());

}

}

/\*\*

\* 获取模型详细信息

\*/

@PreAuthorize("@ss.hasPermi('asm:model:query')")

@GetMapping(value = "/{modelId}")

public AjaxResult getInfo(@PathVariable("modelId") Long modelId) {

return AjaxResult.success(asmModelService.selectAsmModelByModelId(modelId));

}

/\*\*

\* 新增模型

\*/

@PreAuthorize("@ss.hasPermi('asm:model:add')")

@Log(title = "模型", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody AsmModel asmModel) {

return toAjax(asmModelService.insertAsmModel(asmModel));

}

/\*\*

\* 修改模型

\*/

@PreAuthorize("@ss.hasPermi('asm:model:edit')")

@Log(title = "模型", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody AsmModel asmModel) {

return toAjax(asmModelService.updateAsmModel(asmModel));

}

/\*\*

\* 删除模型

\*/

@PreAuthorize("@ss.hasPermi('asm:model:remove')")

@Log(title = "模型", businessType = BusinessType.DELETE)

@DeleteMapping("/{modelIds}")

public AjaxResult remove(@PathVariable Long[] modelIds) {

return toAjax(asmModelService.deleteAsmModelByModelIds(modelIds));

}

}

package com.jiexun.asm.controller;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.asm.domain.AsmProductTransport;

import com.jiexun.asm.service.IAsmProductTransportService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* 产品运输端Controller

\* @author yzc

\* @date 2022-07-20

\*/

@RestController

@RequestMapping("/asm/product\_transport")

public class AsmProductTransportController extends BaseController {

@Autowired

private IAsmProductTransportService asmProductTransportService;

/\*\*

\* 查询产品运输端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:list')")

@GetMapping("/list")

public TableDataInfo list(AsmProductTransport asmProductTransport) {

startPage();

List<AsmProductTransport> list = asmProductTransportService.selectAsmProductTransportList(asmProductTransport);

return getDataTable(list);

}

/\*\*

\* 导入数据

\* @param file

\* @param updateSupport 替换已存在数据（未使用）

\* @return

\* @throws Exception

\*/

@Log(title = "产品运输端", businessType = BusinessType.IMPORT)

@PreAuthorize("@ss.hasPermi('asm:product\_transport:import')")

@PostMapping("/importData")

public AjaxResult importData(MultipartFile file, boolean updateSupport) throws Exception {

ExcelUtil<AsmProductTransport> util = new ExcelUtil<AsmProductTransport>(AsmProductTransport.class);

List<AsmProductTransport> dataList = util.importExcel(file.getInputStream());

String operName = getUsername();

Long operId = getUserId();

String message = asmProductTransportService.importData(dataList, updateSupport, operName, operId);

return AjaxResult.success(message);

}

@PostMapping("/importTemplate")

public void importTemplate(HttpServletResponse response) {

ExcelUtil<AsmProductTransport> util = new ExcelUtil<AsmProductTransport>(AsmProductTransport.class);

util.importTemplateExcel(response, "产品运输端模版数据");

}

/\*\*

\* 导出产品运输端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:export')")

@Log(title = "产品运输端", businessType = BusinessType.EXPORT)

@PostMapping("/export")

public void export(HttpServletResponse response, AsmProductTransport asmProductTransport) {

List<AsmProductTransport> list = asmProductTransportService.selectAsmProductTransportList(asmProductTransport);

ExcelUtil<AsmProductTransport> util = new ExcelUtil<AsmProductTransport>(AsmProductTransport.class);

list.forEach(t -> {

if (null != t.getRemark()) {

t.setRemarkInfo(t.getRemark());

}

});

util.exportExcel(response, list, "产品运输端导出数据");

}

/\*\*

\* 获取产品运输端详细信息

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:query')")

@GetMapping(value = "/{uuid}")

public AjaxResult getInfo(@PathVariable("uuid") String uuid) {

return AjaxResult.success(asmProductTransportService.selectAsmProductTransportByUuid(uuid));

}

/\*\*

\* 新增产品运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:add')")

@Log(title = "产品运输端", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody AsmProductTransport asmProductTransport) {

asmProductTransport.setCreateUser(getUserId());

asmProductTransport.setCreateBy(getUsername());

return toAjax(asmProductTransportService.insertAsmProductTransport(asmProductTransport));

}

/\*\*

\* 修改产品运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:edit')")

@Log(title = "产品运输端", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody AsmProductTransport asmProductTransport) {

return toAjax(asmProductTransportService.updateAsmProductTransport(asmProductTransport));

}

/\*\*

\* 删除产品运输端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_transport:remove')")

@Log(title = "产品运输端", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult remove(@PathVariable String[] uuids) {

return toAjax(asmProductTransportService.deleteAsmProductTransportByUuids(uuids));

}

}

package com.jiexun.asm.controller;

import java.util.List;

import javax.servlet.http.HttpServletResponse;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.PostMapping;

import org.springframework.web.bind.annotation.PutMapping;

import org.springframework.web.bind.annotation.DeleteMapping;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.RequestBody;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.RestController;

import com.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.asm.domain.AsmProductUse;

import com.jiexun.asm.service.IAsmProductUseService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

import org.springframework.web.multipart.MultipartFile;

/\*\*

\* 产品使用端Controller

\* @author yzc

\* @date 2022-07-20

\*/

@RestController

@RequestMapping("/asm/product\_use")

public class AsmProductUseController extends BaseController {

@Autowired

private IAsmProductUseService asmProductUseService;

/\*\*

\* 查询产品使用端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:list')")

@GetMapping("/list")

public TableDataInfo list(AsmProductUse asmProductUse) {

startPage();

List<AsmProductUse> list = asmProductUseService.selectAsmProductUseList(asmProductUse);

return getDataTable(list);

}

/\*\*

\* 导入数据

\* @param file

\* @param updateSupport 替换已存在数据（未使用）

\* @return

\* @throws Exception

\*/

@Log(title = "产品使用端", businessType = BusinessType.IMPORT)

@PreAuthorize("@ss.hasPermi('asm:product\_use:import')")

@PostMapping("/importData")

public AjaxResult importData(MultipartFile file, boolean updateSupport) throws Exception {

ExcelUtil<AsmProductUse> util = new ExcelUtil<AsmProductUse>(AsmProductUse.class);

List<AsmProductUse> dataList = util.importExcel(file.getInputStream());

String operName = getUsername();

Long operId = getUserId();

String message = asmProductUseService.importData(dataList, updateSupport, operName, operId);

return AjaxResult.success(message);

}

@PostMapping("/importTemplate")

public void importTemplate(HttpServletResponse response) {

ExcelUtil<AsmProductUse> util = new ExcelUtil<AsmProductUse>(AsmProductUse.class);

util.importTemplateExcel(response, "产品使用端模版数据");

}

/\*\*

\* 导出产品使用端列表

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:export')")

@Log(title = "产品使用端", businessType = BusinessType.EXPORT)

@PostMapping("/export")

public void export(HttpServletResponse response, AsmProductUse asmProductUse) {

List<AsmProductUse> list = asmProductUseService.selectAsmProductUseList(asmProductUse);

ExcelUtil<AsmProductUse> util = new ExcelUtil<AsmProductUse>(AsmProductUse.class);

list.forEach(t -> {

if (null != t.getRemark()) {

t.setRemarkInfo(t.getRemark());

}

});

util.exportExcel(response, list, "产品使用端导出数据");

}

/\*\*

\* 获取产品使用端详细信息

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:query')")

@GetMapping(value = "/{uuid}")

public AjaxResult getInfo(@PathVariable("uuid") String uuid) {

return AjaxResult.success(asmProductUseService.selectAsmProductUseByUuid(uuid));

}

/\*\*

\* 新增产品使用端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:add')")

@Log(title = "产品使用端", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody AsmProductUse asmProductUse) {

asmProductUse.setCreateUser(getUserId());

asmProductUse.setCreateBy(getUsername());

return toAjax(asmProductUseService.insertAsmProductUse(asmProductUse));

}

/\*\*

\* 修改产品使用端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:edit')")

@Log(title = "产品使用端", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody AsmProductUse asmProductUse) {

return toAjax(asmProductUseService.updateAsmProductUse(asmProductUse));

}

/\*\*

\* 删除产品使用端

\*/

@PreAuthorize("@ss.hasPermi('asm:product\_use:remove')")

@Log(title = "产品使用端", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult remove(@PathVariable String[] uuids) {

return toAjax(asmProductUseService.deleteAsmProductUseByUuids(uuids));

}

}