**炼化技术碳排放计算评价系统V1.0**

北京杰迅鸿翔信息技术有限公司

### **JavaScript**:

import Vue from 'vue'

import Cookies from 'js-cookie'

import Element from 'element-ui'

// 全局修改默认配置，点击空白处不能关闭弹窗

Element.Dialog.props.closeOnClickModal.default = false

import './assets/styles/element-variables.scss'

import '@/assets/styles/index.scss' // global css

import '@/assets/styles/jx.scss' // jx css

import App from './App'

import store from './store'

import router from './router'

import directive from './directive' // directive

import plugins from './plugins' // plugins

import { download } from '@/utils/request'

import './assets/icons' // icon

import './permission' // permission control

import { getDicts } from "@/api/system/dict/data";

import { getConfigKey } from "@/api/system/config";

import { parseTime, resetForm, addDateRange, selectDictLabel, selectDictLabels, handleTree } from "@/utils/jx";

// 分页组件

import Pagination from "@/components/Pagination";

// 自定义表格工具组件

import RightToolbar from "@/components/RightToolbar"

// 富文本组件

import Editor from "@/components/Editor/index"

import cmsEditor from "@/components/cmsEditor"

// 文件上传组件

import FileUpload from "@/components/FileUploadMinIO"

// 图片上传组件

import ImageUpload from "@/components/ImageUpload"

// 图片预览组件

import ImagePreview from "@/components/ImagePreview"

// 字典标签组件

import DictTag from '@/components/DictTag'

// 头部标签组件

import VueMeta from 'vue-meta'

// 字典数据组件

import DictData from '@/components/DictData'

import echarts from "echarts";

Vue.prototype.$echarts = echarts;

// 全局方法挂载

Vue.prototype.getDicts = getDicts

Vue.prototype.getConfigKey = getConfigKey

Vue.prototype.parseTime = parseTime

Vue.prototype.resetForm = resetForm

Vue.prototype.addDateRange = addDateRange

Vue.prototype.selectDictLabel = selectDictLabel

Vue.prototype.selectDictLabels = selectDictLabels

Vue.prototype.download = download

Vue.prototype.handleTree = handleTree

// 全局组件挂载

Vue.component('DictTag', DictTag)

Vue.component('Pagination', Pagination)

Vue.component('RightToolbar', RightToolbar)

Vue.component('Editor', Editor)

Vue.component('cmsEditor', cmsEditor)

Vue.component('FileUpload', FileUpload)

Vue.component('ImageUpload', ImageUpload)

Vue.component('ImagePreview', ImagePreview)

Vue.use(directive)

Vue.use(plugins)

Vue.use(VueMeta)

DictData.install()

/\*\*

\* If you don't want to use mock-server

\* you want to use MockJs for mock api

\* you can execute: mockXHR()

\*

\* Currently MockJs will be used in the production environment,

\* please remove it before going online! ! !

\*/

Vue.use(Element, {

size: Cookies.get('size') || 'medium' // set element-ui default size

})

Vue.config.productionTip = false

new Vue({

el: '#app',

router,

store,

render: h => h(App)

})

<template>

<div id="app">

<router-view />

</div>

</template>

<script>

export default {

name: 'App',

metaInfo() {

return {

title: this.$store.state.settings.dynamicTitle && this.$store.state.settings.title,

titleTemplate: title => {

return title ? `${title} - ${process.env.VUE\_APP\_TITLE}` : process.env.VUE\_APP\_TITLE

}

}

}

}

</script>

<style>

.v-modal{

z-index: 1003!important;

}

</style>

<template>

<div class="login">

<el-form ref="loginForm" :model="loginForm" :rules="loginRules" class="login-form">

<h3 class="title"></h3>

<el-form-item prop="username">

<el-input

v-model="loginForm.username"

type="text"

auto-complete="off"

placeholder="账号"

>

<svg-icon slot="prefix" icon-class="user" class="el-input\_\_icon input-icon" />

</el-input>

</el-form-item>

<el-form-item prop="password">

<el-input

v-model="loginForm.password"

type="password"

auto-complete="off"

placeholder="密码"

@keyup.enter.native="handleLogin"

>

<svg-icon slot="prefix" icon-class="password" class="el-input\_\_icon input-icon" />

</el-input>

</el-form-item>

<el-form-item prop="code" v-if="captchaEnabled">

<el-input

v-model="loginForm.code"

auto-complete="off"

placeholder="验证码"

style="width: 63%"

@keyup.enter.native="handleLogin"

>

<svg-icon slot="prefix" icon-class="validCode" class="el-input\_\_icon input-icon" />

</el-input>

<div class="login-code">

<img :src="codeUrl" @click="getCode" class="login-code-img"/>

</div>

</el-form-item>

<el-checkbox v-model="loginForm.rememberMe" style="margin:0px 0px 25px 0px;">记住密码</el-checkbox>

<el-form-item style="width:100%;">

<el-button

:loading="loading"

size="medium"

type="primary"

style="width:100%;"

@click.native.prevent="handleLogin"

>

<span v-if="!loading">登 录</span>

<span v-else>登 录 中...</span>

</el-button>

<div style="float: right;" v-if="register">

<router-link class="link-type" :to="'/register'">立即注册</router-link>

</div>

</el-form-item>

</el-form>

<!-- 底部 -->

<div class="el-login-footer">

<span>Copyright © 2021-2031 jxhx All Rights Reserved.</span>

</div>

</div>

</template>

<script>

import { getCodeImg } from "@/api/login";

import Cookies from "js-cookie";

import { encrypt, decrypt } from '@/utils/jsencrypt'

export default {

name: "Login",

data() {

return {

codeUrl: "",

loginForm: {

username: "",

password: "",

rememberMe: false,

code: "",

uuid: ""

},

loginRules: {

username: [

{ required: true, trigger: "blur", message: "请输入您的账号" }

],

password: [

{ required: true, trigger: "blur", message: "请输入您的密码" }

],

code: [{ required: true, trigger: "change", message: "请输入验证码" }]

},

loading: false,

// 验证码开关

captchaEnabled: true,

// 注册开关

register: false,

redirect: ''

};

},

watch: {

$route: {

handler: function(route) {

this.redirect = route.query && route.query.redirect;

},

immediate: true

}

},

created() {

this.getCode();

this.getCookie();

},

methods: {

getCode() {

getCodeImg().then(res => {

this.captchaEnabled = res.captchaEnabled === undefined ? true : res.captchaEnabled;

if (this.captchaEnabled) {

this.codeUrl = "data:image/gif;base64," + res.img;

this.loginForm.uuid = res.uuid;

}

});

},

getCookie() {

const username = Cookies.get("username");

const password = Cookies.get("password");

const rememberMe = Cookies.get('rememberMe')

this.loginForm = {

username: username === undefined ? this.loginForm.username : username,

password: password === undefined ? this.loginForm.password : decrypt(password),

rememberMe: rememberMe === undefined ? false : Boolean(rememberMe)

};

},

handleLogin() {

this.$refs.loginForm.validate(valid => {

if (valid) {

this.loading = true;

if (this.loginForm.rememberMe) {

Cookies.set("username", this.loginForm.username, { expires: 30 });

Cookies.set("password", encrypt(this.loginForm.password), { expires: 30 });

Cookies.set('rememberMe', this.loginForm.rememberMe, { expires: 30 });

} else {

Cookies.remove("username");

Cookies.remove("password");

Cookies.remove('rememberMe');

}

this.$store.dispatch("Login", this.loginForm).then(() => {

this.$router.push({ path: this.redirect || "/index" }).catch(()=>{});

}).catch(() => {

this.loading = false;

if (this.captchaEnabled) {

this.getCode();

}

});

}

});

}

}

};

</script>

<style rel="stylesheet/scss" lang="scss">

.login {

display: flex;

justify-content: center;

align-items: center;

height: 100%;

background-image: url("../assets/images/login-background.jpg");

background-size: cover;

}

.title {

margin: 0px auto 30px auto;

text-align: center;

color: #707070;

}

.login-form {

border-radius: 6px;

background: #ffffff;

width: 400px;

padding: 25px 25px 5px 25px;

.el-input {

height: 38px;

input {

height: 38px;

}

}

.input-icon {

height: 39px;

width: 14px;

margin-left: 2px;

}

}

.login-tip {

font-size: 13px;

text-align: center;

color: #bfbfbf;

}

.login-code {

width: 33%;

height: 38px;

float: right;

img {

cursor: pointer;

vertical-align: middle;

}

}

.el-login-footer {

height: 40px;

line-height: 40px;

position: fixed;

bottom: 0;

width: 100%;

text-align: center;

color: #fff;

font-family: Arial;

font-size: 12px;

letter-spacing: 1px;

}

.login-code-img {

height: 38px;

}

</style>

<template>

<el-main>

<router-view :key="key" />

</el-main>

</template>

<script>

export default {

name: 'ment',

computed: {

key() {

return this.$route.path

}

}

}

</script>

<style scoped>

.el-main {

padding: 10px 0;

min-height:880px;

overflow-y: hidden;

}

</style>

<template>

<div id="box">

<div id="title\_box" class="flex\_between">

<div class="flex-center" style="width:16%;">

<img src="../../../../assets/home/carbonFootprint/logo.png" style="width:30px;height: 30px;margin-right:2%;">

<h3 class="font14 white fat550">{{title}}</h3>

</div>

<div id="search" v-show="islogin==false">

<el-input placeholder="请输入您要查询的内容" suffix-icon="white el-icon-search" size="mini" style="width:60%;margin-right:3%;"/>

<el-button class="white" size="mini" @click="tologin"><span>注册</span></el-button>

<el-button class="white" size="mini" @click="tologin"><span>登录</span></el-button>

</div>

<el-dropdown class="avatar-container right-menu-item hover-effect" trigger="click" v-show="islogin">

<div class="avatar-wrapper">

<!-- <el-avatar class="user-avatar" :src="avatar" @error="errorHandler">-->

<!-- <i class="el-icon-s-custom" />-->

<!-- </el-avatar>-->

<el-input placeholder="请输入您要查询的内容" suffix-icon="white el-icon-search" size="mini" style="width:60%;margin-right:3%;"/>

<p class="avatar-Name">你好：{{nickName}}</p>

</div>

<el-dropdown-menu slot="dropdown">

<router-link target="\_blank" to="/index">

<el-dropdown-item>后台管理</el-dropdown-item>

</router-link>

<el-dropdown-item divided @click.native="logout">

<span>退出登录</span>

</el-dropdown-item>

</el-dropdown-menu>

</el-dropdown>

</div>

<div id="nav\_box" class="flex\_center">

<ul class="flex\_around li-pointer">

<li @click="goto('home')"><span><img src="../../../../assets/home/technology/home-logo.png" alt=""></span>首页</li>

<li @click="goto('library')"><span><img src="../../../../assets/home/technology/declare-logo.png" alt=""></span>技术库</li>

<li @click="goto('experts')"><span><img src="../../../../assets/home/technology/validation-logo.png" alt=""></span>专家库</li>

<li @click="goto('standard')"><span><img src="../../../../assets/home/technology/experts-logo.png" alt=""></span>标准评价</li>

<li @click="goto('instructions')"><span><img src="../../../../assets/home/technology/publicity-logo.png" alt=""></span>申报须知</li>

<!-- <li><span><img src="../../../../assets/home/carbonTechnology/saloon-logo.png" alt=""></span>低碳展示大厅</li>-->

<!-- <li><span><img src="../../../../assets/home/carbonTechnology/center-logo.png" alt=""></span>政策资源中心</li>-->

<!-- <li><span><img src="../../../../assets/home/carbonTechnology/about-logo.png" alt=""></span>关于平台</li>-->

</ul>

</div>

</div>

</template>

<script>

import {mapGetters} from "vuex";

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

export default {

name: 'navbar',

data(){

return{

title:"",

islogin:false,

queryParams:{

appId:4

},

}

},

computed: {

...mapGetters([

'avatar',

'name',

'nickName'

]),

},

created(){

// this.title=this.$route.query.title

this.login();

},

methods:{

goto(val){

if(val =='library'){

this.$router.push({

path: '/carbonTechnology/main/library',

});

}else if(val=='home'){

this.$router.push({

path: '/carbonTechnology/main/index',

});

}else if(val == 'experts'){

this.$router.push({

path: '/carbonTechnology/main/experts',

});

}else if(val == 'instructions'){

this.$router.push({

path: '/carbonTechnology/main/instructions',

});

}else if(val == 'standard'){

this.$router.push({

path: '/carbonTechnology/main/standard',

});

}

},

login() {

let token = getToken();

if (token == null || token == '') {

this.islogin = false;

} else {

this.islogin = true;

}

},

tologin() {

this.$router.push({

path: "/login",

query: {

redirect: this.$route.path+'?title='+this.$route.query.title

},

});

},

async logout() {

this.$confirm('确定注销并退出系统登录吗？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

this.$store.dispatch('LogOut').then(() => {

location.href = '/carbonTechnology';

})

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

}

}

}

</script>

<style scoped>

.avatar-container {

margin-right: 30px;

}

.avatar-Name {

margin-left: 10px;

cursor: pointer;

float: right;

font-size: 16px;

color: #ffffff;

white-space:nowrap;

}

#box{

background: url("../../../../assets/home/technology/nav.png") no-repeat;

background-size: 100% 100%;

}

#title\_box{

height: 70%;

}

.el-button{

background-color:rgb(255, 255, 255,0.3);

border: none;

}

#search{

width: 25%;

}

#search span{

color: white;

}

/deep/.el-input\_\_inner {

background-color: rgb(255, 255, 255,0.3) !important;

border: none;

}

/deep/input::-webkit-input-placeholder {

font-size: 12px;

}

#nav\_box{

height: 30%;

background: #467fda;

}

#nav\_box ul{

width: 60%;

list-style: none;

}

#nav\_box li{

color: white;

font-weight: 550;

font-size: 14px;

display: inline-block;

}

#nav\_box li img{

height: 1.1vh;

margin-right: 0.2vw;

vertical-align: middle;

}

.avatar-wrapper{

display: flex;

align-items: center;

}

</style>

<template>

<div id="box">

<NavBar></NavBar>

<keep-alive>

<Main></Main>

</keep-alive>

<Footer></Footer>

</div>

</template>

<script>

import NavBar from './NavBar.vue'

import Main from './main.vue'

import Footer from './Footer.vue'

import store from "@/store";

export default {

name: 'ment',

components: {

NavBar,

Main,

Footer

},

created(){

store.dispatch('settings/setTitle', '')

}

}

</script>

<style scoped>

#box{

width:100%;

height:100px;

}

</style>

<template>

<div id="box" class="flex\_center">

<span>版权所有&nbsp;©&nbsp;中国石化石油化工科学研究院&emsp;&emsp;地址：中国北京市海淀区学院路18号</span>

<span>邮编：100083&emsp;&emsp;ICP备案编号：京ICP备05037230</span>

</div>

</template>

<script>

export default {

name: 'navbar',

data(){

return{

}

}

}

</script>

<style scoped>

#box{

width:100%;

height:70px;

background: #e2e9ec;

font-size: 10px;

display: flex;

flex-direction: column;

}

span{

display: inline-block;

height: 20px;

color: #3e5b77;

}

</style>

<template>

<div id="declare">

<div class="header">低碳技术评价申报入口</div>

<div class="body">

<div class="body-left">

<ul>

<li @click="goto('0')">

<div>

<img src="../../../../../assets/home/technology/dtt-01.png" alt="" v-show="titleIndex!=0">

<img src="../../../../../assets/home/technology/dtt-h01.png" alt="" v-show="titleIndex==0">

</div>

<div :class="titleIndex==0?'active':''">申报流程</div>

</li>

<li @click="goto('1')">

<div>

<img src="../../../../../assets/home/technology/dtt-02.png" alt="" v-show="titleIndex!=1">

<img src="../../../../../assets/home/technology/dtt-h02.png" alt="" v-show="titleIndex==1">

</div>

<div :class="titleIndex==1?'active':''">材料准备</div>

</li>

<li @click="goto('2')">

<div>

<img src="../../../../../assets/home/technology/dtt-03.png" alt="" v-show="titleIndex!=2">

<img src="../../../../../assets/home/technology/dtt-h03.png" alt="" v-show="titleIndex==2">

</div>

<div :class="titleIndex==2?'active':''">材料提报</div>

</li>

<li @click="goto('3')">

<div>

<img src="../../../../../assets/home/technology/dtt-04.png" alt="" v-show="titleIndex!=3">

<img src="../../../../../assets/home/technology/dtt-h04.png" alt="" v-show="titleIndex==3">

</div>

<div :class="titleIndex==3?'active':''">进度及结果查询</div>

</li>

</ul>

</div>

<div class="body-right">

<div class="flex-between" v-show="titleIndex == 0">

<div class="captions">

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

<div class="title">申报流程文字说明</div>

<div class="captions-content">

<div v-html="flowDescription"></div>

<div class="footer">若需要提供流程建模服务，请联系<span>{{ contacts }}</span>,联系电话<span>{{ contactNumber }}</span>, 电子邮箱:<span>{{ contactEmail }}</span></div>

</div>

</el-card>

</div>

<div class='flowChart'>

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

<div class="title">申报流程流程图</div>

<div>

<img :src="imgUrl+flowChart" alt="">

</div>

</el-card>

</div>

</div>

<div class="flex-between" v-show="titleIndex == 1">

<div class="captions">

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

<div class="title">技术分类说明</div>

<div class="captions-content">

<div v-html="description"></div>

<!-- <div class="footer">若需要提供流程建模服务，请联系<span>{{ contacts }}</span>,联系电话<span>{{ contactNumber }}</span>, 电子邮箱:<span>{{ contactEmail }}</span></div>-->

</div>

</el-card>

</div>

<div class='template'>

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

<div class="title">材料清单及模板</div>

<div class="conditionsBox">

<el-form :inline="true" :model="form" class="demo-form-inline">

<el-form-item label="技术类别选择">

<el-select @change="choose" v-model="form.professionalField" placeholder="技术类别选择" clearable>

<el-option

v-for="dict in technicalClassList"

:key="dict.uuid"

:label="dict.name"

:value="dict.uuid"

></el-option>

</el-select>

</el-form-item>

</el-form>

</div>

<div class="templateList">

<div v-for="item in templateList" @click="download(item)"><i></i>{{ item.originalName }}</div>

</div>

</el-card>

</div>

</div>

<div v-show="titleIndex == 2">

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

<div class="provided" >

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

<el-row>

<el-col :span="14">

<el-form-item label="待评价技术名称" label-width="120px" prop="name">

<el-autocomplete

v-model="form.name"

:fetch-suggestions="querySearchAsync"

placeholder="请输入待评价技术名称"

@select="handleSelect"

></el-autocomplete>

<!-- <el-input v-model="form.name" placeholder="请输入待评价技术名称"></el-input>-->

</el-form-item>

</el-col>

<el-col :span="10">

<el-form-item label="关键词">

<el-input v-model="form.keyword" placeholder="请输入关键词" />

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="8">

<el-form-item label="技术领域">

<el-select v-model="form.field" placeholder="请选择技术领域">

<el-option

v-for="dict in fieldList"

:key="dict.dictValue"

:label="dict.dictLabel"

:value="dict.dictValue"

></el-option>

</el-select>

</el-form-item>

</el-col>

<el-col :span="8">

<el-form-item label="技术类别" prop="type">

<el-select v-model="form.type" placeholder="请选择技术类别">

<el-option

v-for="dict in technicalClassList"

:key="dict.uuid"

:label="dict.name"

:value="dict.uuid"

></el-option>

</el-select>

</el-form-item>

</el-col>

<el-col :span="8">

<el-form-item label="所属阶段" prop="stage">

<el-select v-model="form.stage" placeholder="请选择所属阶段">

<el-option

v-for="dict in technicalStageList"

:key="dict.dictValue"

:label="dict.dictLabel"

:value="dict.dictValue"

/>

</el-select>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

</el-col>

</el-row>

<el-row>

<el-col :span="24">

<el-form-item label="待评价技术简介" prop="remark" label-width="120px">

<el-input v-model="form.remark" type="textarea" maxlength="200" show-word-limit placeholder="请输入待评价技术简介" />

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="24">

<el-form-item label="类比技术名称" prop="analogyName" label-width="120px">

<el-input v-model="form.analogyName" placeholder="请输入类比技术名称" />

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="24">

<el-form-item label="类比技术简介" prop="analogyRemark" label-width="120px">

<el-input v-model="form.analogyRemark" type="textarea" placeholder="请输入类比技术简介" maxlength="200" show-word-limit/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="技术材料">

<file-upload v-model="form.technicalMaterials"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="第三方资质证明" label-width="120px">

<file-upload v-model="form.qualificationCertificate"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="专利数量" prop="patentNum">

<el-input v-model="form.patentNum" type="number" placeholder="请输入专利数量" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="专利文件">

<file-upload v-model="form.patentDoc"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="国家级获奖数量" prop="nationalAwardsNum" label-width="120px">

<el-input v-model="form.nationalAwardsNum" type="number" placeholder="请输入国家级获奖数量" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="国家级获奖文件" label-width="120px">

<file-upload v-model="form.nationalAwardsDoc"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="省部级获奖数量" prop="ministerialAwardsNum" label-width="120px">

<el-input v-model="form.ministerialAwardsNum" type="number" placeholder="请输入省部级获奖数量" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="省部级获奖文件" label-width="120px">

<file-upload v-model="form.ministerialAwardsDoc"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="工业化证明数量" prop="industrializationCertificateNum" label-width="120px">

<el-input v-model="form.industrializationCertificateNum" type="number" placeholder="请输入工业化证明数量" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="工业化证明文件" label-width="120px">

<file-upload v-model="form.industrializationCertificateDoc"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="24">

<el-form-item label="其它证明文件" label-width="120px">

<file-upload v-model="form.otherDoc"/>

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="姓名" prop="applyUser">

<el-input v-model="form.applyUser" placeholder="请输入姓名" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="单位" prop="applyCompany">

<el-input v-model="form.applyCompany" placeholder="请输入单位" />

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item label="联系电话" prop="applyPhone">

<el-input v-model="form.applyPhone" placeholder="请输入联系电话" />

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="电子邮箱" prop="applyEmail">

<el-input v-model="form.applyEmail" placeholder="请输入电子邮箱" />

</el-form-item>

</el-col>

</el-row>

<el-row>

<el-col :span="12">

<el-form-item>

<!-- <el-button type="primary" @click="onSubmit">立即创建</el-button>-->

<!-- <el-button>取消</el-button>-->

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item>

<el-button type="info" @click="draft">存为草稿</el-button>

<el-button type="primary" @click="onSubmit">立即提交</el-button>

</el-form-item>

</el-col>

</el-row>

</el-form>

</div>

</el-card>

</div>

<div v-show="titleIndex == 3">

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

<div class="resultsShow">

<div class="headers flex-around">

<div>

<span>选择查询的技术</span>

<el-select v-model="value" placeholder="请选择">

<el-option

v-for="item in technologyByLoginNameList"

:key="item.uuid"

:label="item.name"

:value="item.uuid">

</el-option>

</el-select>

</div>

</div>

<div class="schedule">

<div class="title">评价进度</div>

<div class="content">

<div class="progressBarBox">

<!-- 驳回颜色红色-->

<!-- -->

<el-steps :active="2" finish-status="success">

<el-step title="材料提交"></el-step>

<el-step title="材料审核退回" description="退回原因" status="error"></el-step>

<el-step title="专家评审"></el-step>

<el-step title="结果审核"></el-step>

<el-step title="结果发布"></el-step>

<el-step title="评价完成"></el-step>

</el-steps>

</div>

</div>

</div>

<div class="result">

<div class="title">评价结果</div>

<div class="content">

<div class="resultBox">

1、通过资料预审的申请资料，EESIA秘书处将组织评审专家组进行集中评审。专家将严格按照评价规则，针对申请单位提交的申请资料进行评审，并由评审专家组打分确定企业评价等级并向EESIA秘书处出具评价结论。需要补充解释性材料或部分内容修改的评价资料，由申请单位3个工作日内根据评审意见提交补充材料或修改内容后函复EESIA秘书处；对于无法出具评价等级的申请资料，EESIA在评审结论做出后3个工作日内通知申请单位。

</div>

<div class="downloadBtn">

<el-button type="primary" icon="el-icon-download">评价报告下载</el-button>

</div>

<div class="evaluationShow">

<span>2、该评价结果是否在平台进行展示？</span>

<el-radio v-model="radio" label="1">是</el-radio>

<el-radio v-model="radio" label="2">否</el-radio>

<el-button type="primary" size="mini">提交</el-button>

</div>

</div>

</div>

</div>

</el-card>

</div>

</div>

</div>

</div>

</template>

<script>

import {

getConfigurationList,

getTechnicalClassList,

gettTechnicalStageList,

addTechnology,

getTechnologyByLoginNameList

} from "@/api/carbonTechnology";

export default {

name: "declare",

data(){

return {

radio:'',

titleIndex:0,

flowDescription:'',

contacts:'',

contactNumber:'',

contactEmail:'',

flowChart:'',

imgUrl:process.env.VUE\_APP\_BASE\_API+ "/common/getMinIoFileById?id=",

description:'',

form:{},

rules:{

// name:[

// { required: true, type: 'integer', min: 1, message: '请正确选择客户', trigger: 'change' }

// ]

},

technicalClassList:[],

templateList:[],

fieldList:[],

scienceList:[],

technicalStageList:[],

value:'',

options:[],

active:3,

technologyByLoginNameList:[],

restaurants:[],

queryParams:{

reviewState:''

},

technicalScore:'',

economyScore:'',

socialScore:'',

comprehensiveSocre:''

}

},

mounted() {

this.getList()

},

methods:{

querySearchAsync(queryString, cb) {

var restaurants = this.restaurants;

var results = queryString ? restaurants.filter(this.createStateFilter(queryString)) : restaurants;

clearTimeout(this.timeout);

this.timeout = setTimeout(() => {

cb(results);

}, 1000 \* Math.random());

},

createStateFilter(queryString) {

return (state) => {

return (state.value.toLowerCase().indexOf(queryString.toLowerCase()) === 0);

};

},

handleSelect(item) {

this.form = item

this.form.name = item.name

},

draft(){

this.form.reviewState = '1'

this.form.name = this.form.name!==null?this.form.name.trim():''

if(this.form.name!=''){

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

if (valid) {

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

this.$modal.msgSuccess("提交成功");

this.reset()

});

}

});

}else {

this.$message({

message: '待评价技术名称不能为空',

type: 'warning'

});

}

},

onSubmit(){

this.form.name = this.form.name!==null?this.form.name.trim():''

if(this.form.name!=''){

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

if (valid) {

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

this.$modal.msgSuccess("提交成功");

this.reset()

});

}

});

}else {

this.$message({

message: '待评价技术名称不能为空',

type: 'warning'

});

}

},

// 表单重置

reset() {

this.form = {

uuid: null,

name: null,

type: null,

field: null,

stage: null,

remark: null,

analogyName: null,

analogyRemark: null,

platformDisplay: 0,

technicalMaterials: null,

qualificationCertificate: null,

patentNum: null,

patentDoc: null,

nationalAwardsNum: null,

nationalAwardsDoc: null,

ministerialAwardsNum: null,

ministerialAwardsDoc: null,

industrializationCertificateNum: null,

industrializationCertificateDoc: null,

otherDoc: null,

reviewState: null,

evaluateLevel: null,

advantage: null,

topLevel: null,

searchNum: null,

applyUser: null,

applyCompany: null,

applyPhone: null,

applyEmail: null,

applyTime: null,

createTime: null,

delState: null

};

this.resetForm("form");

},

download(item){

window.location.href=this.imgUrl+item.uuid

},

choose(val){

this.technicalClassList.filter((item)=>{

if(item.uuid == val){

this.templateList = item.files

}

})

},

goto(item){

this.titleIndex = item

if(item == '2'){

this.reset()

}

},

getList(){

this.queryParams.reviewState='1'

getConfigurationList('flowDescription').then(res => {

this.flowDescription = res.data

})

getConfigurationList('contacts').then(res => {

this.contacts = res.data

})

getConfigurationList('contactNumber').then(res => {

this.contactNumber = res.data

})

getConfigurationList('contactEmail').then(res => {

this.contactEmail = res.data

})

getConfigurationList('flowChart').then(res => {

this.flowChart = res.data

})

getConfigurationList('description').then(res => {

this.description = res.data

})

gettTechnicalStageList('low\_carbon\_pro\_field').then(res => {

this.fieldList = res.data

})

gettTechnicalStageList('technical\_stage').then(res => {

this.technicalStageList = res.data

})

//技术性指标得分

gettTechnicalStageList('technicalScore').then(res => {

this.technicalScore = res.data

})

//经济性指标得分

gettTechnicalStageList('economyScore').then(res => {

this.economyScore = res.data

})

//社会性指标得分

gettTechnicalStageList('socialScore').then(res => {

this.socialScore = res.data

})

//综合评价得分

gettTechnicalStageList('comprehensiveSocre').then(res => {

this.comprehensiveSocre = res.data

})

getTechnologyByLoginNameList(this.queryParams).then(res => {

res.rows.map(item => {

item.value= item.name

})

this.restaurants = res.rows

})

getTechnologyByLoginNameList().then(res => {

this.technologyByLoginNameList = res.rows

})

getTechnicalClassList().then(res => {

this.technicalClassList = res.rows

})

},

}

}

</script>

<style scoped>

#declare{

width: 60%;

margin: 0 auto;

}

.header{

width: 275px;

margin: 0 auto;

text-align: center;

color: #fff;

background: url("../../../../../assets/home/technology/dt01.png") no-repeat;

}

.body {

margin-top: 60px;

display: flex;

justify-content: space-between;

}

.body .body-left{

width: 120px;

padding-top: 30px;

}

.body .body-left ul li {

/\*height: 160px;\*/

margin-bottom: 60px;

}

.body .body-left ul li div{

text-align: center;

color: #666666;

font-size: 14px;

}

.body-right{

width: 85%;

min-height: 100%;

}

.body-right>div{

width: 100%;

height: 100%;

}

.captions,.flowChart,.template{

width: 45%;

height: 100%;

}

.captions .title,.flowChart .title,.template .title{

text-align: center;

font-weight: 600;

}

/deep/ .el-card{

height: 100%;

}

.body .body-left ul li div.active{

font-size: 16px;

font-weight: bold;

}

.footer span{

color: rgb(0, 87, 255);

}

.flowChart div img {

width: 100%;

}

.template .conditionsBox{

padding-top: 20px;

}

.template .templateList div i{

height: 5px;

width: 5px;

background-color: #3aa2eb;

border-radius: 50%;

display:inline-block;

margin-right: 10px;

line-height: 18px;

vertical-align: middle;

}

/\*.template .templateList div:after{\*/

/\* content: '';\*/

/\* margin: 3px;\*/

/\* !\*display: table;\*!\*/

/\* width: 6px;\*/

/\* height: 6px;\*/

/\* background: #2090ff;\*/

/\* border-radius: 50%;\*/

/\*}\*/

.template .templateList div {

color: #3aa2eb;

cursor: pointer;

overflow:hidden;

white-space: nowrap;

text-overflow: ellipsis;

-o-text-overflow:ellipsis;

}

.resultsShow .headers div>span{

margin-right: 10px;

}

.resultsShow .schedule .title,.resultsShow .result .title{

background: #e0ecfc;

padding-left: 10px;

font-size: 14px;

height: 30px;

line-height: 30px;

font-weight: bold;

}

.resultsShow .schedule {

margin-top: 30px;

}

.resultsShow .result{

border-left: 1px solid #e0ecfc;

border-bottom: 1px solid #e0ecfc;

border-right: 1px solid #e0ecfc;

}

.resultsShow .schedule .content{

width: 100%;

min-height: 200px;

display: flex;

justify-content: center;

align-items: center;

}

.resultsShow .result .content{

width: 100%;

min-height: 200px;

padding: 20px 40px 0;

}

.resultsShow .result .content .resultBox{

font-size: 14px;

font-family: "宋体", "微软雅黑";

line-height: 20px;

}

.resultsShow .result .downloadBtn{

margin: 10px 0;

}

.resultsShow .schedule .progressBarBox{

width: 90%;

}

</style>

<template>

<div id="expertsDetails">

</div>

</template>

<script>

export default {

name: "details"

}

</script>

<style scoped>

</style>

<template>

<div id="experts">

<!-- <div class="searchBox">-->

<!-- </div>-->

<div class="expertsBox">

<div class="expertsBox-left">

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

<el-row>

<el-col :span="9">

<el-form-item label="姓名" prop="name">

<el-input

v-model="queryParams.name"

placeholder="请输入姓名"

clearable

@keyup.enter.native="handleQuery"

/>

</el-form-item>

</el-col>

<el-col :span="9">

<el-form-item label="专业领域" prop="professionalField">

<el-select v-model="queryParams.professionalField" placeholder="请选择专业领域" clearable>

<el-option

v-for="dict in fieldList"

:key="dict.dictValue"

:label="dict.dictLabel"

:value="dict.dictValue"

></el-option>

</el-select>

</el-form-item>

</el-col>

<el-col :span="6">

<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-col>

</el-row>

</el-form>

<div>

<ul >

<el-row :gutter="12">

<el-col :span="24" v-for="(item,index) in expertsList" :key="index">

<el-card shadow="hover">

<div @click="gotoContent(item.id)">

<div class="photo">

<img :src="imgUrl+item.photo" alt="">

</div>

<div class="message">

<div class="expertsName">{{ item.name }}<span>{{ item.title | getTitleName }}</span><span>{{ item.industry }}</span></div>

<div class="companyName">{{ item.companyName }}</div>

<!-- <div class="titleKey"><span>从事行业：</span>{{ item.industry }}</div>-->

<div class="titleKey"><span>技术领域：</span>{{ getFieldName(item.professionalField) }}</div>

<div class="titleKey introduce"><span>简单介绍：</span><span>{{ item.briefIntroduction }}</span></div>

</div>

</div>

</el-card>

</el-col>

</el-row>

</ul>

<pagination

v-show="total>0"

:total="total"

:page-sizes='pageSize'

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

</div>

</div>

<div class="expertsBox-right">

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

<div class="technologyDistribution">

<div class="title">专家职称职务分布</div>

<div id="main" ref="main"></div>

</div>

<div class="dataAnalysis">

<div class="title">专家专业领域分布</div>

<div id="main1" ref="main1"></div>

</div>

</el-card>

</div>

</div>

</div>

</template>

<script>

import echarts from "echarts";

import {

getExpertList,

getJobDistributionAnalysisList,

getProfessionalFieldsDistributionList,

gettTechnicalStageList

} from "@/api/carbonTechnology";

var \_self;

export default {

name: "experts",

data(){

return{

// 显示搜索条件

showSearch: true,

total:0,

pageSize:['10'],

queryParams:{

pageNum: 1,

pageSize: 10,

},

expertsList:[],

titleList:[],

fieldList:[],

imgUrl:process.env.VUE\_APP\_BASE\_API+ "/common/getMinIoFileById?id=",

jobDistributionAnalysisList:[],

professionalFieldsDistributionList:[],

dataAnalysis:'',

distribution:''

}

},

beforeCreate() {

\_self = this;

},

filters:{

getTitleName(type){

let name = "无";

for (let obj of \_self.titleList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

},

mounted() {

this.getList()

this.getConditionsList()

},

methods:{

getList(){

getExpertList(this.queryParams).then(res => {

this.expertsList = res.rows

this.total = res.total

})

},

gotoContent(uuid){

const {href} = this.$router.resolve({

path: "/carbonTechnology/main/expertsContent",

query: {id: uuid}

});

window.open(href, "\_blank");

},

getConditionsList(){

gettTechnicalStageList('low\_carbon\_occupation\_name').then(res => {

this.titleList = res.data

})

gettTechnicalStageList('low\_carbon\_professional\_field').then(res => {

this.fieldList = res.data

})

getJobDistributionAnalysisList().then(res => {

this.jobDistributionAnalysisList = res.data

this.distribution = echarts.init(this.$refs.main);

this.$nextTick(() => {

this.init()

})

})

getProfessionalFieldsDistributionList().then(res => {

this.professionalFieldsDistributionList = res.data

this.dataAnalysis = echarts.init(this.$refs.main1);

this.$nextTick(() => {

this.init1()

})

})

},

/\*\* 搜索按钮操作 \*/

handleQuery() {

// this.queryParams.pageNum = 1;

this.expertsList = []

// console.log(this.queryParams)

this.getList();

},

/\*\* 重置按钮操作 \*/

resetQuery() {

this.resetForm("queryForm");

this.handleQuery();

},

getFieldName(fields){

let fieldObjs = this.fieldList;

if(!fieldObjs || !fields){

return '';

}

const fieldArr = fields.split(',');

let nameArr = [];

f1 :for (let id of fieldArr) {

f2 : for(let obj of fieldObjs){

if(obj.dictValue == id){

nameArr.push(obj.dictLabel);

break f2;

}

}

}

let res = nameArr.join(",");

return res;

},

init(){

this.distribution.setOption({

tooltip: {

trigger: 'item'

},

legend: {

top: '5%',

left: 'center'

},

// grid: {

// top: 80,

// left: '2%',

// right: '8%',

// bottom: '5%',

// containLabel: true

// },

series: [

{

name: '专家职称职务',

type: 'pie',

radius: ['40%', '70%'],

center: ['50%', '60%'],

avoidLabelOverlap: false,

itemStyle: {

borderRadius: 10,

borderColor: '#fff',

borderWidth: 2,

},

detail: {

// 调节数字位置

// offsetCenter: [-1,-15],

fontSize: 10,

fontFamily:'PangMenZhengDao' ,

color: '#D8E9FF',

},

label: {

show: false,

position: 'center'

},

emphasis: {

label: {

show: true,

fontSize: '14',

fontWeight: 'bold'

}

},

labelLine: {

show: false

},

data: this.jobDistributionAnalysisList

}

]

})

},

init1(){

this.dataAnalysis.setOption({

tooltip: {

trigger: 'item'

},

legend: {

top: '5%',

left: 'center'

},

// grid: {

// top: 80,

// left: '2%',

// right: '8%',

// bottom: '5%',

// containLabel: true

// },

series: [

{

name: '专家职称职务',

type: 'pie',

radius: ['40%', '70%'],

center: ['50%', '70%'],

avoidLabelOverlap: false,

itemStyle: {

borderRadius: 10,

borderColor: '#fff',

borderWidth: 2,

},

detail: {

// 调节数字位置

// offsetCenter: [-1,-15],

fontSize: 10,

fontFamily:'PangMenZhengDao' ,

color: '#D8E9FF',

},

label: {

show: false,

position: 'center'

},

emphasis: {

label: {

show: true,

fontSize: '14',

fontWeight: 'bold'

}

},

labelLine: {

show: false

},

data: this.professionalFieldsDistributionList

}

]

})

}

}

}

</script>

<style scoped>

#experts{

width: 70%;

margin: 0 auto;

}

.searchBox{

width: 100%;

}

.expertsBox {

width: 100%;

display: flex;

justify-content: space-between;

}

.expertsBox .expertsBox-left{

width: 69%;

}

.expertsBox .expertsBox-left ul li{

display: flex;

margin-top: 1vh;

}

ul /deep/ .el-card\_\_body > div{

display: flex;

cursor: pointer;

}

/deep/ .el-col-24{

margin-bottom: 10px;

}

ul /deep/ .el-row {

margin: 0;

margin-right: 10px !important;

}

.photo {

width: 120px;

}

.photo img{

width: 100%;

}

/deep/ .el-card\_\_body .message{

margin-left: 1vw;

}

.message .expertsName{

font-size: 1.8vh;

margin-bottom: 0.5vh;

color: #467fda;

}

.message .expertsName span{

margin-left: 1vw;

color: #000;

/\*font-size: 1.4vh;\*/

}

.message .companyName{

font-size: 1.6vh;

margin-bottom: 0.5vh;

}

.message .titleKey{

font-size: 1.6vh;

color: #666666;

}

.message .titleKey span:nth-child(1){

display: inline-block;

text-align: right;

}

.message> .introduce {

display: flex;

}.

.message> .introduce span:nth-child(1){

margin-left: 0.35vw;

}

.expertsBox .message> .introduce span:nth-child(2){

width: 80%;

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

display: -moz-box;

-moz-line-clamp: 2;

-moz-box-orient: vertical;

overflow-wrap: break-word;

word-break: break-all;

white-space: normal;

overflow: hidden;

}

.message {

width: 80%;

}

.expertsBox .expertsBox-right{

width: 355px;

}

.technologyDistribution .title,.dataAnalysis .title{

text-align: center;

line-height: 4vh;

font-weight: bold;

}

.dataAnalysis #main1{

height: 400px;

}

.technologyDistribution #main{

height: 400px;

}

</style>

<template>

<div id="expertsContent">

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

<div class="box">

<div class="box-left">

<img :src="imgUrl+expertInfoList.photo" alt="">

</div>

<div class="box-right">

<div class="name">{{ expertInfoList.name }}</div>

<div class="position">

<div class="postName">{{ expertInfoList.postName }}</div>

{{ expertInfoList.companyName }}

<span>|</span>

{{ expertInfoList.title | getTitleName }}

<span>|</span>

{{ expertInfoList.industry }}

</div>

<div class="field">

<div class="title"><img src="@/assets/home/technology/min-logo.png" alt="">技术领域：</div>

<div class="content">{{ getFieldName(expertInfoList.professionalField) }}</div>

</div>

<div class="detailed">

<div class="title"><img src="@/assets/home/technology/min-logo.png" alt="">详细信息：</div>

<div class="content" v-html="expertInfoList.detailedInfo"></div>

</div>

</div>

</div>

</el-card>

</div>

</template>

<script>

import {getExpertInfoList, gettTechnicalStageList} from "@/api/carbonTechnology";

var \_self;

export default {

name: "expertsContent",

data(){

return {

id:this.$route.query.id,

expertInfoList:{},

imgUrl:process.env.VUE\_APP\_BASE\_API+ "/common/getMinIoFileById?id=",

titleList:[],

fieldList:[],

}

},

beforeCreate() {

\_self = this;

},

filters:{

getTitleName(type){

let name = "无";

for (let obj of \_self.titleList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

},

mounted() {

this.getList()

},

methods:{

getList(){

gettTechnicalStageList('low\_carbon\_occupation\_name').then(res => {

this.titleList = res.data

})

gettTechnicalStageList('low\_carbon\_professional\_field').then(res => {

this.fieldList = res.data

})

getExpertInfoList(this.id).then(res => {

console.log(res)

this.expertInfoList = res.data

})

},

getFieldName(fields){

let fieldObjs = this.fieldList;

if(!fieldObjs || !fields){

return '';

}

const fieldArr = fields.split(',');

let nameArr = [];

f1 :for (let id of fieldArr) {

f2 : for(let obj of fieldObjs){

if(obj.dictValue == id){

nameArr.push(obj.dictLabel);

break f2;

}

}

}

let res = nameArr.join(",");

return res;

},

},

}

</script>

<style scoped>

#expertsContent {

width: 60%;

margin: 0 auto;

}

.box-card{

padding: 30px 0 0 30px;

}

.box{

display: flex;

}

.box-left {

width: 135px;

}

.box-left img{

width: 100%;

}

.box-right{

padding-left: 35px;

width: 85%;

}

.box-right .name{

font-size: 22px;

font-weight: 600;

line-height: 60px;

}

.box-right .position{

color: #999999;

}

.box-right .position .postName{

margin-bottom: 10px;

}

.box-right .position span{

margin: 0 5px;

}

.box-right .field .title,.box-right .detailed .title{

margin-top: 20px;

font-size: 15px;

font-weight: bold;

margin-bottom: 10px;

}

.box-right .field .title img,.box-right .detailed .title img{

vertical-align: middle;

margin-right: 5px;

}

.box-right .field .content{

text-indent: 20px;

color: #666666;

}

</style>

<template>

<div style="height: 100%">

<div id="box">

<div id="content">

<div id="initBox">

<el-carousel trigger="click" height="33.2vh">

<el-carousel-item v-for="(item,index) in imgurl" :key="index">

<img :src="imgUrl+item" alt="">

</el-carousel-item>

</el-carousel>

</div>

<div class="textList">

<div class="left">

<div style="height: 66vh">

<div class="font16 fat550 title flex\_between"><span>低碳技术展示</span><span @click="technologyMore">查看更多</span></div>

<div class="flex\_grow flex\_around iconList">

<div class="scrollBox">

<div v-for="(item,index) in topTechnologyList" :key="index" class="scroll\_boxs flex\_between top" @click="gotoLibrary(item.uuid)">

<div class="item-left">

<div class="top-img">

<img src="@/assets/home/technology/top.png" alt="">

</div>

<div class="content">

<div class="technicalTitle">{{ item.name }}</div>

<div class="technicalMessage">申请人：{{ item.applyUser }} 所属阶段：{{ item.stage | getTypeNames }} 技术类别：{{ item.type | getTypeName }}</div>

<div class="technicalAdvantage">{{ item.advantage }}</div>

<div class="remark">{{ item.remark }}</div>

</div>

</div>

<div class="item-right">

<div class="technicalRating">技术评级：{{ item.evaluateLevel | getLevelNames }}</div>

<div class="technicalDate">{{ parseTime(item.applyTime, '{y}-{m}-{d}') }}</div>

</div>

</div>

<vue-seamless-scroll :class-option="classOption" :data="TechnologyShowList" class="warp">

<div v-for="(item,index) in TechnologyShowList" :key="index" class="scroll\_boxs flex\_between" @click="gotoLibrary(item.uuid)">

<div class="item-left">

<div class="technicalTitle">{{ item.name }}</div>

<div class="technicalMessage">申请人：{{ item.applyUser }} 所属阶段：{{ item.stage | getTypeNames }} 技术类别：{{ item.type | getTypeName }}</div>

<div class="technicalAdvantage">{{ item.advantage }}</div>

<div class="remark">{{ item.remark }}</div>

</div>

<div class="item-right">

<div class="technicalRating">技术评级：{{ item.evaluateLevel | getLevelNames }}</div>

<div class="technicalDate">{{ parseTime(item.applyTime, '{y}-{m}-{d}') }}</div>

</div>

</div>

</vue-seamless-scroll>

</div>

<div style="width: 400px">

<div class="declareEntrance">

<img src="@/assets/home/carbonTechnology/rk.png" alt="" @click="gotoDeclare">

<!-- <div class="">低碳技术评价申报入口</div>-->

</div>

</div>

</div>

</div>

<div style="height:40%">

<div class="font16 fat550 title flex\_between"><span>专家介绍</span><span @click="expertsMore">查看更多</span></div>

<div class="flex\_grow review">

<ul class="ul\_pinsheng flex\_between">

<li class="expertsBox flex\_between" v-for="(item,index) in expertsList">

<div class="box-left">

<div class="expertsBox-left" @click="gotoExperts(item.id)">

<img :src="imgUrl+item.photo" alt="">

</div>

<div class="expertsBox-right">

<div class="expertsName">{{ item.name }}<span>{{ item.title | getTitleName }}</span></div>

<div class="titleKey"><span>单位：</span>{{ item.companyName }}</div>

<div class="titleKey"><span>从事行业：</span>{{ item.industry }}</div>

<div class="titleKey"><span>技术领域：</span>{{ getFieldName(item.professionalField) }}</div>

<div class="titleKey introduce"><span>简单介绍：</span><span>{{ item.briefIntroduction }}</span></div>

</div>

</div>

<div class="box-right li-pointer">

<div class="messageBtn" @click="gotoExperts(item.id)">查看详情</div>

</div>

</li>

</ul>

</div>

</div>

<div style="height:40%">

<!-- <div class="font16 fat550 title flex\_between"><span>评审公示</span><span @click="expertsMore">查看更多</span></div>-->

<div class="font16 fat550 title flex\_between"><span>评审公示</span><span @click="technologyMore">查看更多</span></div>

<div class="flex\_grow review">

<ul class="ul\_zhengce">

<li v-for="item in libraryList" class="flex\_between">

<span>·{{ item.name}}技术通过评审</span>

<span>{{ parseTime(item.applyTime, '{y}-{m}-{d}') }}</span>

</li>

</ul>

</div>

</div>

</div>

</div>

</div>

</div>

<div class="box\_ flex-center">

<div class="swiper-content flex-center">

<div id="nav">

<div v-for="(item,index) in slides" :key="index">

<div class="swiper-item li-pointer" @click="goto(item.route)">

<img :src="item.src" alt="">

<p>{{ item.name }}</p>

</div>

</div>

</div>

</div>

</div>

</div>

</template>

<script>

import {

getConfigurationList,

getExpertList,

getHotTechnologyList, getTechnicalClassList, getTechnologyList,

getTopTechnologyList,

gettTechnicalStageList

} from "@/api/carbonTechnology";

var \_self;

const echarts = require('echarts');

import "../../../../../../node\_modules/echarts/map/js/china.js";

import vueSeamlessScroll from 'vue-seamless-scroll'

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

export default{

components: {

vueSeamlessScroll

},

computed:{

classOption: function () {

return {

step: 0.1,

limitMoveNum: 4

}

},

},

data(){

return{

imgurl: [],

slides: [

{src: require('../../../../../assets/home/icon.png'), name: '',route:'home'},

{src: require('../../../../../assets/home/examination.png'), name: '',route:'carbonProducts'},

{src: require('../../../../../assets/home/carbonFootprint.png'), name: '',route:'skip'},

{src: require('../../../../../assets/home/policiesRegulations.png'), name: '',route:'carbonStandard'},

{src: require('../../../../../assets/home/icon.png'), name: '',route:'refinery'},

],

expertsList:[],

TechnologyShowList:[],

titleList:[],

fieldList:[],

imgUrl:process.env.VUE\_APP\_BASE\_API+ "/common/getMinIoFileById?id=",

technicalStageList:[],

scienceList:[],

evaluateLevel:[],

expertParams:{

pageNum: 1,

pageSize: 4,

},

queryParams:{

pageNum: 1,

pageSize: 6,

},

libraryList:[],

topTechnologyList:[]

}

},

beforeCreate() {

\_self = this;

},

filters:{

getTitleName(type){

let name = "无";

for (let obj of \_self.titleList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getTypeNames(type){

let name = "无";

for (let obj of \_self.technicalStageList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getTypeName(type){

let name = "无";

for (let obj of \_self.scienceList) {

if (obj.uuid == type) {

name = obj.name;

return name;

}

}

return name;

},

getLevelNames(type){

let name = "无";

for (let obj of \_self.evaluateLevel) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

},

created(){

this.getTechnologyShowList()

this.getExpertLists()

this.getList()

},

mounted(){

},

methods:{

getList(){

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

})

},

gotoDeclare(){

let token = getToken();

if (token == null || token == '') {

this.$confirm('申报需要登录账户，您确定要去登录吗？', '提示', {

confirmButtonText: '确定',

cancelButtonText: '取消',

type: 'warning'

}).then(() => {

this.$router.push({

path: "/login",

query: {

redirect: this.$route.path+'?title='+this.$route.query.title

},

});

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

} else {

const {href} = this.$router.resolve({

path: "/carbonTechnology/main/declare",

});

window.open(href, "\_blank");

}

},

gotoLibrary(id){

const {href} = this.$router.resolve({

path: "/carbonTechnology/main/libraryContent",

query: {id: id}

});

window.open(href, "\_blank");

},

gotoExperts(id){

const {href} = this.$router.resolve({

path: "/carbonTechnology/main/expertsContent",

query: {id: id}

});

window.open(href, "\_blank");

},

getFieldName(fields){

let fieldObjs = this.fieldList;

if(!fieldObjs || !fields){

return '';

}

const fieldArr = fields.split(',');

let nameArr = [];

f1 :for (let id of fieldArr) {

f2 : for(let obj of fieldObjs){

if(obj.dictValue == id){

nameArr.push(obj.dictLabel);

break f2;

}

}

}

let res = nameArr.join(",");

return res;

},

getTechnologyShowList(){

getConfigurationList('titleImage').then(res => {

this.imgurl = res.data.split(',')

})

getTopTechnologyList().then(res => {

this.topTechnologyList = res.rows

})

getHotTechnologyList().then(res => {

this.TechnologyShowList = res.rows

})

},

//获取专家列表

getExpertLists(){

getExpertList(this.expertParams).then(res => {

this.expertsList = res.rows

})

gettTechnicalStageList('low\_carbon\_occupation\_name').then(res => {

this.titleList = res.data

})

gettTechnicalStageList('low\_carbon\_professional\_field').then(res => {

this.fieldList = res.data

})

gettTechnicalStageList('technical\_stage').then(res => {

this.technicalStageList = res.data

})

getTechnicalClassList().then(res=>{

this.scienceList = res.rows

})

gettTechnicalStageList('evaluate\_level').then(res => {

this.evaluateLevel = res.data

})

},

expertsMore(){

this.$router.push({

path: "/carbonTechnology/main/experts",

});

},

technologyMore(){

this.$router.push({

path: "/carbonTechnology/main/library",

});

},

goto(item){

if(item=='carbonStandard'){

const { href } = this.$router.resolve({

path: "/carbonStandard/main/index",

});

window.open(href, "\_blank");

}else if(item=='refinery'){

const { href } = this.$router.resolve({

path: "/refinery/main/index",

})

window.open(href, "\_blank");

}else if(item=='carbonProducts'){

const { href } = this.$router.resolve({

path: "/carbonProducts/main/productsIndex",

});

window.open(href, "\_blank");

}else if(item=='skip'){

const { href } = this.$router.resolve({

path: "/carbonFootprint/main/index",

});

window.open(href, "\_blank");

}else if(item=='home'){

const { href } = this.$router.resolve({path: "/home/main/cmsIndex"});

window.open(href, "\_blank");

}

},

}

}

</script>

<style scoped>

#box{

height: 90%;

width: 100%;

display: flex;

justify-content: center;

}

#content{

height: 100%;

width: 70%;

}

#initBox{

/\*border: 1px solid #f0f0f0;\*/

/\*padding: 10px 0;\*/

height:320px;

}

#initBox img{

width: 100%;

}

.textList{

/\*height: 445px;\*/

padding: 10px 0;

/\*display: flex;\*/

}

.textList>div{

width: 100%;

height: 100%;

}

.textList .title{

height: 30px;

display: flex;

align-items: center;

border-bottom: 1px solid #ededed;

}

.textList .title>span:nth-child(1){

display: inline-block;

padding: 0 5px;

height: 100%;

line-height: 200%;

border-bottom:2px solid #4982de;

}

.textList .title>span:nth-child(2){

font-weight: normal;

margin-right: 1vw;

cursor: pointer;

}

.left>div{

margin-bottom: 1vh;

}

.left .iconList{

height: 90%;

margin-top: 1vh;

}

.iconList>.scrollBox{

width: 100%;

height: 100%;

overflow: hidden;

}

.iconList>.scrollBox .scroll\_boxs{

padding: 1vh 1vw;

align-items: flex-start;

/\*margin-bottom: 1vh;\*/

border-bottom: 1px solid #eeeeee;

cursor: pointer;

}

.iconList>.scrollBox .scroll\_boxs .technicalTitle{

font-family: "宋体", "微软雅黑";

font-size: 1.8vh;

font-weight: 600;

}

.iconList>.scrollBox .scroll\_boxs .technicalMessage,.iconList>.scrollBox .scroll\_boxs .remark{

font-size: 1.4vh;

color: #707070;

margin-bottom: 0.2vh;

}

.iconList>.scrollBox .top-img img{

width: 15px;

margin-top: 10px;

margin-right: 10px;

}

.iconList>.scrollBox .top .item-left{

display: flex;

width: 75%;

}

.iconList>.scrollBox .warp .item-left{

width: 75%;

}

.iconList>.scrollBox .top .item-left>.content{

width: 95%;

}

.iconList>.scrollBox .warp .item-left>\*{

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.iconList>.scrollBox .item-left>.content>\*{

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.iconList>.scrollBox .item-right{

width: 130px;

}

.item-right .technicalRating{

line-height: 5vh;

margin-bottom: 1vh;

}

.technicalAdvantage{

font-size: 14px;

color: #467fda;

}

.declareEntrance{

text-align: center;

}

.declareEntrance img {

width: 80%;

cursor: pointer;

}

.expertsBox .box-left{

display: flex;

}

.expertsBox .box-right{

width: 9vw;

}

.expertsBox .box-right .messageBtn{

width: 80%;

height: 3vh;

/\*margin: 0 auto;\*/

border: 1px solid #467fda;

border-radius: 5px;

text-align: center;

font-size: 1.6vh;

line-height: 3vh;

white-space: nowrap;

color: #467fda;

}

.ul\_pinsheng{

flex-wrap: wrap;

}

.ul\_pinsheng li{

width: 50%;

margin-bottom: 2vh;

}

.ul\_pinsheng li .expertsBox-left{

width: 145px;

}

.ul\_pinsheng li .expertsBox-left img{

width: 100%;

}

.ul\_pinsheng li .expertsBox-right {

width: 18vw;

margin-left: 1vw;

flex-wrap: wrap;

}

.ul\_pinsheng li .expertsBox-right> div{

overflow:hidden;

white-space: nowrap;

text-overflow: ellipsis;

}

.ul\_pinsheng li .expertsBox-right .expertsName{

font-size: 1.8vh;

}

.ul\_pinsheng li .expertsBox-right .expertsName span{

font-size: 1.4vh;

}

.ul\_pinsheng li .expertsBox-right .titleKey span:nth-child(1){

display: inline-block;

width: 4vw;

text-align: right;

}

.ul\_pinsheng li .expertsBox-right> .introduce {

display: flex;

}.ul\_pinsheng li .expertsBox-right> .introduce span:nth-child(1){

margin-left: 0.35vw;

}

.ul\_pinsheng li .expertsBox-right> .introduce span:nth-child(2){

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

display: -moz-box;

-moz-line-clamp: 2;

-moz-box-orient: vertical;

overflow-wrap: break-word;

word-break: break-all;

white-space: normal;

overflow: hidden;

}

/\*.left .iconList>div{\*/

/\* width: 16%;\*/

/\* height: 100%;\*/

/\* display: flex;\*/

/\* align-items: center;\*/

/\*}\*/

.left .iconItem{

display:inline-block;

text-align: center;

}

.left span{

display: inline-block;

padding: 0 5px;

height: 100%;

line-height: 200%;

}

/\*.left img{\*/

/\* width: 45px;\*/

/\* height: 45px;\*/

/\*}\*/

.review{

overflow: hidden;

padding: 10px 0 0 0;

}

.review ul{

list-style: none;

font-size: 12px;

height: 100%;

overflow: hidden;

}

/\*.review li span:last-child{\*/

/\* float: right;\*/

/\* color: #9a9a9a;\*/

/\* font-size: 12px;\*/

/\*}\*/

.ul\_zhengce li{

font-size: 1.6vh;

/\*height: 20%;\*/

padding: 0 1vw;

}

.ul\_zhuanjia li{

height: 20%;

}

.right .iconList{

width: 100%;

height:30%;

background: #e2ebfa;

}

.right .iconList>div{

width: 30%;

height: 100%;

display: flex;

align-items: center;

justify-content: center;

}

.right .iconItem{

height: 100%;

display: inline-block;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

.right img{

height: 40px;

}

.right .icons{

height: 65%;

}

.icons img{

width: 45px;

height: 45px;

}

.contact{

background: url("../../../../../assets/home/technology/contactus.png") no-repeat;

background-size: 100% 100%;

height: 35%;

}

.swiper-content{

width: 70%;

height: 10vh;

/\*background-color: #e7f7fe;\*/

/\*padding-left: 3vw;\*/

position: relative;

}

.swiper-content>div{

white-space: nowrap;

overflow-x: auto;

-webkit-touch-callout:none; /\*系统默认菜单被禁用\*/

-webkit-user-select:none; /\*webkit浏览器\*/

-khtml-user-select:none; /\*早期浏览器\*/

-moz-user-select:none;/\*火狐\*/

-ms-user-select:none; /\*IE10\*/

user-select:none;

}

.swiper-slide{

width: 17.36vw!important;

height: 6vh;

}

.swiper-slide .card {

/\*width: 340px;\*/

/\*height: 320px;\*/

width: 14.07vw;

height: 6vh;

/\*background: url("https://img2.baidu.com/it/u=1473738496,3480051138&fm=253&fmt=auto&app=138&f=JPEG?w=500&h=500") no-repeat;\*/

position: relative;

cursor: pointer;

}

.swiper-slide:nth-child(odd) .swiper-item{

background: url("../../../../../assets/home/swiper-item.png") no-repeat;

background-size: 340px 320px;

}

.swiper-slide:nth-child(even) .swiper-item{

background: url("../../../../../assets/home/bg.png") no-repeat;

background-size: 340px 320px;

}

.swiper-content .swiper-item{

width: 13vw;

height: 6vh;

display: flex;

padding-left: 1.8vw;

align-items:center;

/\*margin-right: 2.8vw;\*/

/\*margin-top: 2vh;\*/

padding-right: 1vw;

white-space: normal;

}

.swiper-content .swiper-item img{

width: 1.5vw;

height: 2.6vh;

margin-right: 0.75vw;

}

.swiper-content .swiper-item p{

color: #FFFFFF;

font-size: 1.6vh;

}

.swiper-button-prev,.swiper-button-next{

width: 2vw;

height: 5vh;

}

.swiper-button-prev:after, .swiper-container-rtl .swiper-button-next:after{

content:'';

/\*background: url("../../../assets/home/left.png");\*/

}

.swiper-button-next:after, .swiper-container-rtl .swiper-button-prev:after{

content: '';

/\*background: url("../../../assets/home/right.png");\*/

}

.swiper-content #nav{

display: flex;

width: 100%;

justify-content: space-around;

}

.swiper-content #nav div:nth-child(odd) .swiper-item{

background: url("../../../../../assets/home/swiper-item.png") no-repeat;

}

.swiper-content #nav div:nth-child(even) .swiper-item{

background: url("../../../../../assets/home/bg.png") no-repeat;

}

</style>

<template>

<div id="instructions">

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

<div class="content" v-html="list"></div>

</el-card>

</div>

</template>

<script>

import {getConfigurationList} from "@/api/carbonTechnology";

export default {

name: "instructions",

data(){

return {

list:''

}

},

mounted() {

this.getList()

},

methods:{

getList(){

getConfigurationList('declarationInstructions').then(res => {

this.list = res.data

})

}

}

}

</script>

<style scoped>

#instructions {

width: 60%;

margin: 0 auto;

}

</style>

<template>

<div id="library">

<div class="library-left">

<div class="filterBox">

<div class="filter-item">

<div class="conditionsName">所属阶段:</div>

<ul class="li-pointer">

<li :class="phaseKey==null?'active':''" @click="choosePhase()">全部</li>

<li :class="phaseKey==index?'active':''" v-for="(item,index) in technicalStageList" :key="index" @click="choosePhase(index,item)">{{item.dictLabel}}</li>

</ul>

</div>

<div class="filter-item">

<div class="conditionsName">技术分类:</div>

<ul class="li-pointer">

<li :class="scienceKey==null?'active':''" @click="chooseScience()">全部</li>

<li :class="scienceKey==index?'active':''" @click="chooseScience(index,item)" v-for="(item,index) in scienceList" :key="index">{{ item.name }}</li>

</ul>

</div>

</div>

<div class="filterContent">

<div class="filterContent-left">

<el-row :gutter="12">

<el-col :span="24" v-for="(item,index) in libraryList" :key="index">

<el-card shadow="hover">

<div @click="gotoContent(item.uuid)" class="item">

<div class="item-left">

<!-- <p style='width: 8%'>{{ item.model\_zone }}</p>-->

<div class="technicalTitle">{{ item.name }}</div>

<!-- <p style='width: 10%'>{{ item.model\_state }}</p>-->

<div class="technicalMessage">申请人：{{ item.applyUser }} 所属阶段：

<!-- <dict-tag :options="dict.type.technical\_stage" :value="item.stage"/>-->

{{ item.stage | getTypeNames }}

技术类别：{{ item.type | getTypeName }}</div>

<div class="technicalAdvantage">{{ item.advantage }}</div>

<div >{{ item.remark }}</div>

</div>

<div class="item-right">

<div class="technicalRating">技术评级：{{ item.evaluateLevel | getLevelNames }}

<!-- <dict-tag :options="dict.type.evaluate\_level" :value="item.evaluateLevel"/>-->

</div>

<div class="technicalDate">{{ parseTime(item.applyTime, '{y}-{m}-{d}') }}</div>

</div>

</div>

</el-card>

</el-col>

</el-row>

<pagination

v-show="total>0"

:total="total"

:page-sizes='pageSize'

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

</div>

<div class="library-right">

<el-card class="box-card" v-if="hotTechnologyList.length>0">

<div class="title">热点技术</div>

<ul>

<li v-for="(item,index) in hotTechnologyList" :key="index"><span></span>{{ item.name }}</li>

</ul>

</el-card>

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

<div class="technologyDistribution">

<div class="title">技术应用程度分析</div>

<div id="main" ref="main"></div>

</div>

<div class="dataAnalysis">

<div class="title">技术评价等级分析</div>

<div id="main1" ref="main1"></div>

</div>

</el-card>

</div>

</div>

</div>

</div>

</template>

<script>

import echarts from "echarts";

import {

getAppDegreeAnalysisList, getAppLevelAnalysisList, getHotTechnologyList,

getTechnicalClassList,

getTechnologyList,

gettTechnicalStageList

} from "@/api/carbonTechnology";

var \_self;

export default {

name: "library",

data(){

return {

phaseKey:null,

phaseList:[

'全部','初期阶段','中期阶段','成熟应用'

],

scienceKey:null,

scienceList:[],

libraryList:[],

total:0,

pageSize:['10'],

queryParams: {

pageNum: 1,

pageSize: 10,

type: null,

stage: null,

},

distribution:'',

dataAnalysis:'',

technicalStageList:[],

evaluateLevel:[],

series:[],

series1:[],

hotTechnologyList:[]

}

},

created() {

this.getTechnicalClassLists()

this.getInitList()

},

mounted() {

this.getList()

},

beforeCreate() {

\_self = this;

},

filters:{

getTypeNames(type){

let name = "无";

for (let obj of \_self.technicalStageList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getLevelNames(type){

let name = "无";

for (let obj of \_self.evaluateLevel) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getTypeName(type){

let name = "无";

for (let obj of \_self.scienceList) {

if (obj.uuid == type) {

name = obj.name;

return name;

}

}

return name;

},

},

methods:{

gotoContent(uuid){

const {href} = this.$router.resolve({

path: "/carbonTechnology/main/libraryContent",

query: {id: uuid}

});

window.open(href, "\_blank");

},

getTechnicalClassLists(){

getTechnicalClassList().then(res=>{

this.scienceList = res.rows

})

gettTechnicalStageList('technical\_stage').then(res => {

this.technicalStageList = res.data

})

gettTechnicalStageList('evaluate\_level').then(res => {

this.evaluateLevel = res.data

})

getHotTechnologyList().then(res => {

this.hotTechnologyList = res.rows

})

},

getInitList(){

getAppDegreeAnalysisList().then(res => {

let obj={}

res.data.map(item => {

item.hj = item.cj+item.zj+item.cs

obj[item.name]=[]

obj[item.name].push(item.cj)

obj[item.name].push(item.zj)

obj[item.name].push(item.cs)

obj[item.name].push(item.hj)

});

for(let key in obj){

this.series.push({

name: key,

type: 'bar',

stack: 'Ad',

emphasis: {

focus: 'series'

},

label: {

show: true,

position: 'inside'

},

data: obj[key]

})

}

this.distribution = echarts.init(this.$refs.main);

this.$nextTick(() => {

this.init()

})

})

getAppLevelAnalysisList().then(res => {

let obj={}

res.data.map(item => {

obj[item.name]=[]

obj[item.name].push(item.aa)

obj[item.name].push(item.a)

obj[item.name].push(item.b)

obj[item.name].push(item.c)

obj[item.name].push(item.d)

});

for(let key in obj){

this.series1.push({

name: key,

type: 'bar',

stack: 'Ad',

emphasis: {

focus: 'series'

},

label: {

show: true,

position: 'inside'

},

data: obj[key]

})

}

this.dataAnalysis = echarts.init(this.$refs.main1);

this.$nextTick(() => {

this.init1()

})

})

},

getList(){

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

this.total = res.total

})

},

choosePhase(index,item){

this.phaseKey = index

if(item!=undefined){

this.queryParams.stage = item.dictValue

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

this.total = res.total

})

}else {

this.queryParams.stage = null

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

this.total = res.total

})

}

},

chooseScience(index,item){

if(item!=undefined){

this.queryParams.type = item.uuid

this.scienceKey = index!=undefined?index:null

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

this.total = res.total

})

}else {

this.queryParams.type = null

this.scienceKey = index!=undefined?index:null

getTechnologyList(this.queryParams).then(res => {

this.libraryList = res.rows

this.total = res.total

})

}

},

init(){

this.distribution.setOption({

tooltip: {

trigger: 'axis',

axisPointer: { // 坐标轴指示器，坐标轴触发有效

type: 'shadow' // 默认为直线，可选为：'line' | 'shadow'

}

},

legend: {},

grid: {

top: 80,

left: '2%',

right: '8%',

bottom: '5%',

containLabel: true

},

xAxis: [{

type: 'category',

data: ['初级应用','中级应用','成熟应用','合计'],

axisTick: {

alignWithLabel: true

},

axisLine: { // 坐标轴 轴线

show: false, // 是否显示

lineStyle: {

color: '#eeeeee',

width: 2,

type: 'solid'

}

},

axisLabel:{

interval:0,

rotate:40,

textStyle:{

color:'#808080'

}

}

}],

yAxis: [{

show: true,

type: 'value',

name: '',

scale:true,

axisTick: {

show: false

},

// min:82,

// max: 92,

minInterval:2,

axisLine: { // 坐标轴 轴线

show: false, // 是否显示

lineStyle: {

color: '#666666',

width: 1,

type: 'solid'

}

},

}],

series: this.series

});

},

init1(){

this.dataAnalysis.setOption({

tooltip: {

trigger: 'axis',

axisPointer: { // 坐标轴指示器，坐标轴触发有效

type: 'shadow' // 默认为直线，可选为：'line' | 'shadow'

}

},

legend: {},

grid: {

top: 80,

left: '2%',

right: '8%',

bottom: '5%',

containLabel: true

},

xAxis: [{

type: 'category',

data: ['A+级技术','A级技术','B级技术','C级技术','D级技术'],

axisTick: {

alignWithLabel: true

},

axisLine: { // 坐标轴 轴线

show: false, // 是否显示

lineStyle: {

color: '#eeeeee',

width: 2,

type: 'solid'

}

},

axisLabel:{

interval:0,

rotate:40,

textStyle:{

color:'#808080'

}

}

}],

yAxis: [{

show: true,

type: 'value',

name: '',

scale:true,

axisTick: {

show: false

},

// min:82,

// max: 92,

minInterval:2,

axisLine: { // 坐标轴 轴线

show: false, // 是否显示

lineStyle: {

color: '#666666',

width: 1,

type: 'solid'

}

},

}],

series: this.series1

});

}

}

}

</script>

<style scoped>

#library{

width: 70%;

margin: 0 auto;

display: flex;

justify-content: space-between;

}

#library .library-left{

width: 100%;

}

.filterBox .filter-item{

display: flex;

padding: 1vh 0;

}

.filterBox .filter-item ul {

display: flex;

margin-left: 1vw;

}

.filterBox .filter-item ul li{

padding: 0px 10px;

}

.filterBox .filter-item ul li.active{

background: #4982de;

color: #fff;

}

.filterContent{

display: flex;

justify-content: space-between;

padding: 2vh 0 ;

}

.filterContent .item{

width: 95%;

margin-bottom: 2vh;

cursor: pointer;

}

.filterContent .item-left{

width: 75%;

}

.filterContent .item-left>\*{

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.filterContent .item-right{

width: 140px;

}

.filterContent .item-left .technicalTitle{

font-family: "宋体", "微软雅黑";

font-size: 1.8vh;

font-weight: 600;

}

.filterContent .item-left .technicalMessage{

font-size: 1.4vh;

color: #707070;

margin-bottom: 0.2vh;

}

.filterContent .item-left .technicalMessage div{

display: inline;

}

.filterContent .item-left .technicalAdvantage{

color: #467fda;

}

.filterContent .item-right .technicalRating{

line-height: 5vh;

margin-bottom: 1vh;

}

.library-right{

width: 400px;

}

.library-right .el-card:nth-child(1){

margin-bottom: 20px;

}

.technologyDistribution{

width: 100%;

}

.technologyDistribution .title{

text-align: center;

line-height: 4vh;

font-weight: bold;

}

.dataAnalysis #main1{

height: 300px;

}

.dataAnalysis{

width: 100%;

}

.dataAnalysis .title{

text-align: center;

line-height: 4vh;

font-weight: bold;

}

.technologyDistribution #main{

/\*background: #0000;\*/

/\*min-height: 25vh;\*/

height: 300px;

}

/deep/ .el-card\_\_body > .item{

display: flex;

justify-content: space-between;

}

/deep/ .el-col-24{

margin-bottom: 10px;

}

.filterContent /deep/ .el-row {

margin: 0;

margin-right: 10px !important;

}

.filterContent-left {

width: 68%;

}

.box-card .title{

text-align: center;

line-height: 3vh;

font-weight: bold;

}

.box-card ul li:before{

content: '';

display: inline-block;

width: 4px;

height: 4px;

background-color: #000;

vertical-align: middle;

margin-right: 7px;

border-radius: 4px;

}

</style>

<template>

<div id="libraryContent">

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

<div class="contentBoxHeader">

<div class="header flex-between">

<div class="header-left">

<div class="title">

<div class="titleName">{{ libraryObj.name }}</div>

<div class="level">{{ libraryObj.evaluateLevel | getLevelNames }}</div>

</div>

<div class="advantage">

<ul>

<li v-for="item in advantageList">{{ item }}</li>

</ul>

</div>

</div>

<div class="header-right">

<div class="time">{{ parseTime(libraryObj.applyTime, '{y}-{m}-{d}') }}</div>

</div>

</div>

<div class="message">

<div class="applicant">

<p class="title"><span>关键词</span></p>

<div class="content">{{ libraryObj.keyword }}</div>

</div>

<div class="applicant">

<div class="applicant">

<p class="title"><span>申请人</span></p>

<div class="content">{{ libraryObj.applyUser }}</div>

</div>

<p class="title"><span>单位名称</span></p>

<div class="content">{{ libraryObj.applyCompany }}</div>

</div>

<div class="applicant">

<p class="title"><span>所属阶段</span></p>

<div class="content">{{ libraryObj.stage | getTypeNames }}</div>

</div>

<div class="applicant">

<p class="title"><span>技术类型</span></p>

<div class="content">{{ libraryObj.type | getTypeName }}</div>

</div>

<div class="applicant">

<p class="title"><span>技术领域</span></p>

<div class="content">{{ getFieldName(libraryObj.field) }}</div>

</div>

<div class="introduce">

<p class="title"><span>技术简介</span></p>

<div class="content">{{ libraryObj.remark }}</div>

</div>

<!-- <div class="applicant">-->

<!-- <p class="title"><span>类比技术</span></p>-->

<!-- <div class="content" style="text-indent: 10px">类比技术名称：{{ libraryObj.analogyName }}</div>-->

<!-- <div class="content" style="text-indent: 10px">类比技术简介：<span style="text-indent: 20px;" v-html="libraryObj.analogyRemark"></span></div>-->

<!-- </div>-->

</div>

</div>

</el-card>

</div>

</template>

<script>

import {getLibraryList, getTechnicalClassList, gettTechnicalStageList} from "@/api/carbonTechnology";

var \_self;

export default {

name: "libraryContent",

data(){

return{

uuid:this.$route.query.id,

libraryObj:{},

evaluateLevel:[],

advantageList:[],

technicalStageList:[],

scienceList:[],

fieldList:[]

}

},

beforeCreate() {

\_self = this;

},

filters:{

getLevelNames(type){

let name = "无";

for (let obj of \_self.evaluateLevel) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getTypeNames(type){

let name = "无";

for (let obj of \_self.technicalStageList) {

if (obj.dictValue == type) {

name = obj.dictLabel;

return name;

}

}

return name;

},

getTypeName(type){

let name = "无";

for (let obj of \_self.scienceList) {

if (obj.uuid == type) {

name = obj.name;

return name;

}

}

return name;

},

},

mounted() {

this.getList()

},

methods:{

getFieldName(fields){

let fieldObjs = this.fieldList;

if(!fieldObjs || !fields){

return '';

}

const fieldArr = fields.split(',');

let nameArr = [];

f1 :for (let id of fieldArr) {

f2 : for(let obj of fieldObjs){

if(obj.dictValue == id){

nameArr.push(obj.dictLabel);

break f2;

}

}

}

let res = nameArr.join(",");

return res;

},

getList(){

gettTechnicalStageList('evaluate\_level').then(res => {

this.evaluateLevel = res.data

})

gettTechnicalStageList('technical\_stage').then(res => {

this.technicalStageList = res.data

})

getTechnicalClassList().then(res=>{

this.scienceList = res.rows

})

gettTechnicalStageList('low\_carbon\_pro\_field').then(res => {

this.fieldList = res.data

})

getLibraryList(this.uuid).then(res => {

this.libraryObj = res.data

this.advantageList = res.data.advantage.split('、')

})

},

},

}

</script>

<style scoped>

#libraryContent {

width: 60%;

margin: 0 auto;

}

.contentBoxHeader .header .header-left {

width: 90%;

}

.contentBoxHeader .header .header-right {

width: 100px;

}

.contentBoxHeader .header .title{

display: flex;

align-items: center;

}

.contentBoxHeader .titleName{

/\*text-align: center;\*/

font-size: 28px;

}

.contentBoxHeader .header .title .level{

width: 36px;

height: 20px;

margin-left: 20px;

text-align: center;

line-height: 22px;

color: #fff;

background: url("../../../../../assets/home/technology/level.png") no-repeat;

}

.contentBoxHeader .header .advantage ul{

display: flex;

flex-wrap: wrap;

margin-top: 10px;

}

.contentBoxHeader .header .advantage ul li{

border: 1px solid #97be6c;

color: #97be6c;

border-radius: 5px;

font-size: 12px;

padding: 3px 4px;

margin-right: 10px;

margin-bottom: 10px;

}

.contentBoxHeader .message{

padding: 0 20px;

}

.contentBoxHeader .message .title span{

border-bottom: 2px solid #4982de;

padding-bottom: 5px;

font-weight: bold;

font-size: 15px;

}

.contentBoxHeader .message .content{

color: #989898;

font-size: 14px;

text-indent: 30px;

}

/deep/ .contentBoxHeader .message .content p span{

color: #989898!important;

}

</style>

<template>

<div style="height: 100%">

<!-- <pdf-->

<!-- v-for="i in numPages"-->

<!-- :key="i"-->

<!-- :src="src"-->

<!-- :page="i"-->

<!-- style="display: inline-block; width: 70%;margin: 0 auto"-->

<!-- ></pdf>-->

<!-- <iframe ref="mainiframe" :src="src" style="height:100%;width:100%" ></iframe>-->

<embed disposition="inline" type="application/pdf" :src="src" width="100%" height="100%">

</div>

</template>

<script>

import pdf from "vue-pdf";

export default {

name: "pdfContent",

components: { pdf },

data(){

return {

numPages:undefined,

src: '',

url: process.env.VUE\_APP\_BASE\_API + "/lowCarbonApi/openPdf/"

}

},

created() {

let urls = this.url + this.$route.query.name

this.src = urls

// this.src = pdf.createLoadingTask(urls)

},

mounted() {

// this.src.promise.then(pdf => {

// this.numPages = pdf.numPages;

// });

}

}

</script>

<style scoped>

</style>

<template>

<div id="standard">

<div class="standard-left">

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

<ul class="li-pointer">

<li v-for="(item,index) in dataList" @click="gotoContent(item)" :key="index" class="flex-between">

<div class="name">

<div class="circle">·</div>

<div class="fileName">{{ item.fileName }}</div>

</div>

<div class="time">{{ parseTime(item.createTime, '{m}-{d}') }}</div>

</li>

</ul>

<pagination

v-show="total>0"

:total="total"

:page-sizes='pageSize'

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="getList"

/>

</el-card>

</div>

<div class="standard-right">

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

<div class="title">低碳技术业发展动态</div>

</el-card>

</div>

</div>

</template>

<script>

import {getEvaluationCriteriaList} from "@/api/carbonTechnology";

export default {

name: "standard",

data(){

return {

dataList:[],

queryParams: {

pageNum: 1,

pageSize: 15,

},

total:0,

pageSize:['15'],

}

},

mounted() {

this.getList()

},

methods:{

gotoContent(item){

const { href } = this.$router.resolve({

path: '/pdfContent',

query:{name:item.fileName}

});

// let href = this.url+item.fileId

window.open(href, "\_blank");

},

getList(){

getEvaluationCriteriaList(this.queryParams).then(res => {

console.log(res)

this.total = res.total

this.dataList = res.rows

})

}

}

}

</script>

<style scoped>

#standard{

width: 60%;

min-height: 880px;

margin: 0 auto;

display: flex;

justify-content: space-between;

}

.standard-left {

width: 70%;

min-height: 880px;

}

.standard-right{

width: 28%;

}

.standard-left ul li{

line-height: 4.4vh;

border-bottom: 1px solid #eeeeee;

padding-left: 1vw;

}

.standard-left ul li:hover{

background: #eeeeee;

}

.standard-left ul li .name{

display: flex;

}

.standard-left ul li .fileName {

overflow:hidden;

text-overflow: ellipsis;

-o-text-overflow:ellipsis;

}

.standard-left ul li .time{

margin-right: 10px;

white-space: nowrap;

}

.standard-left .circle {

display: inline-block;

margin-right: 5px;

font-size: 4vh;

}

.standard-right .title{

text-align: center;

font-weight: bold;

font-size: 18px;

}

/deep/ .el-card\_\_body{

min-height: 880px;

}

</style>

### Java:

package com.jiexun.api.constants;

/\*\*

\* cms 常量 . <br>

\* @author YanZhiCheng

\* date: 2022/7/27上午9:33

\*/

public class CmsConstants {

/\*\* 删除状态 0 正常，1 已删除 \*/

public static final Integer DEL\_STATE\_FALSE = 0;

public static final Integer DEL\_STATE\_TRUE = 1;

/\*\* 内容审核状态 0 未审核，1 审核通过，2 审核退回 \*/

public static final Integer AUDIT\_STATE\_NONE = 0;

public static final Integer AUDIT\_STATE\_PASS = 1;

public static final Integer AUDIT\_STATE\_RETURN = 2;

/\*\* 内容发布状态 0 草稿，1 已发布 \*/

public static final Integer RELEASE\_STATE\_FALSE = 0;

public static final Integer RELEASE\_STATE\_TRUE = 1;

/\*\* 内容公开状态 0 否，1 是 \*/

public static final Integer CHARGE\_FALSE = 0;

public static final Integer CHARGE\_TRUE = 1;

/\*\* 内容是否是轮播图 0 否，1 是 \*/

public static final Integer ROTATION\_CHART\_FALSE = 0;

public static final Integer ROTATION\_CHART\_TRUE = 1;

/\*\* 内容的阅读权限 1完全公开（不涉及知识产权的），2会员公开（日后用于会员发布，包含一些专家观点等），3内部公开（仅供院内有权限的人看） \*/

public static final Integer PERMISSION\_TYPE\_ALL = 1;

public static final Integer PERMISSION\_TYPE\_MEMBER = 2;

public static final Integer PERMISSION\_TYPE\_ROLE = 3;

/\*\* 非置顶的内容 \*/

public static final Integer NOT\_TOP = 0;

/\*\* 内容 索引 \*/

public static final String CONTENT\_INDEX = "cms\_content";

/\*\* 内容 索引 \*/

public static final String CONTENT\_INDEX\_OLD = "cms\_content\_new";

/\*\* 内容 映射 \*/

public static final String CONTENT\_MAPPING = "{\"properties\":{\"appId\":{\"type\":\"keyword\"},\"auditStatus\":{\"type\":\"keyword\"}," +

"\"author\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"channelId\":{\"type\":\"keyword\"},\"charge\":{\"type\":\"keyword\"}," +

"\"content\":{\"type\":\"text\",\"analyzer\":\"ik\_max\_word\"},\"contentTempId\":{\"type\":\"keyword\"}," +

"\"createBy\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"createTime\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"createUser\":{\"type\":\"keyword\"},\"delState\":{\"type\":\"keyword\"},\"deptId\":{\"type\":\"long\"}," +

"\"id\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"isHot\":{\"type\":\"keyword\"},\"isTop\":{\"type\":\"keyword\"},\"memberGroupId\":{\"type\":\"long\"}," +

"\"otherFiles\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":256}}}," +

"\"pictures\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":256}}}," +

"\"readCount\":{\"type\":\"keyword\"},\"releaseStatus\":{\"type\":\"keyword\"},\"releaseTime\":{\"type\":\"text\"," +

"\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}},\"rotationChart\":{\"type\":\"keyword\"}," +

"\"source\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":256}}}," +

"\"subTitle\":{\"type\":\"text\",\"analyzer\":\"ik\_max\_word\"},\"permissionType\":{\"type\":\"keyword\"}," +

"\"tags\":{\"type\":\"keyword\"}," +

"\"title\":{\"type\":\"text\",\"analyzer\":\"ik\_max\_word\"},\"roleId\":{\"type\":\"keyword\"}," +

"\"titleImg\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":256}}}," +

"\"toTopEndTime\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"updateBy\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"updateTime\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":64}}}," +

"\"uuid\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above\":32}}}}}";

/\*\* 搜索词 索引 \*/

public static final String SEARCH\_WORD\_INDEX = "cms\_search\_words";

/\*\* 搜索词 映射 \*/

public static final String SEARCH\_WORD\_MAPPING = "{\"properties\":{\"id\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\"" +

":\"keyword\",\"ignore\_above\":32}}},\"appId\":{\"type\":\"keyword\"},\"channelId\":{\"type\":\"keyword\"},\"searchText" +

"\":{\"type\":\"keyword\"},\"createBy\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\",\"ignore\_above" +

"\":64}}},\"createTime\":{\"type\":\"text\",\"fields\":{\"keyword\":{\"type\":\"keyword\"," +

"\"ignore\_above\":64}}}}}";

}

package com.jiexun.api.constants;

/\*\*

\* 常量 . <br>

\* @author YanZhiCheng

\* date: 2022/8/29下午5:19

\*/

public class LowCarbonConstants {

/\*\* 低碳产品验证阶段

\* 1 草稿

\* 3 提交

\* 5 材料审核中

\* 7 材料审核通过

\* 9 材料审核退回

\* 11 专家评审中

\* 13 专家评审完成

\* 15 结果审核中

\* 17 结果审核中完成

\* 19 结果发布中

\* 25 结果发布完成

\*/

public static final Integer PRO\_STATE\_DRAFT = 1;

public static final Integer PRO\_STATE\_SUBMIT = 3;

public static final Integer PRO\_STATE\_MATERIAL\_REVIEW\_ING = 5;

public static final Integer PRO\_STATE\_MATERIAL\_REVIEW\_PASS = 7;

public static final Integer PRO\_STATE\_MATERIAL\_REVIEW\_BACK = 9;

public static final Integer PRO\_STATE\_EXPERT\_REVIEW\_ING = 11;

public static final Integer PRO\_STATE\_EXPERT\_REVIEW\_END = 13;

public static final Integer PRO\_STATE\_RESULT\_REVIEW\_ING = 15;

public static final Integer PRO\_STATE\_RESULT\_REVIEW\_END = 17;

public static final Integer PRO\_STATE\_RESULTS\_RELEASE\_ING = 19;

public static final Integer PRO\_STATE\_RESULTS\_RELEASE\_END = 25;

}

package com.jiexun.api.controller;

import com.jiexun.api.service.ILargeScreenService;

import com.jiexun.api.vo.DeviceOsblVO;

import com.jiexun.asm.domain.\*;

import com.jiexun.asm.service.\*;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.utils.DateUtils;

import com.jiexun.common.utils.SecurityUtils;

import com.jiexun.common.utils.StringUtils;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.RequestMapping;

import org.springframework.web.bind.annotation.ResponseBody;

import java.util.\*;

/\*\*

\* 大屏机迁移 . <br>

\* @author YanZhiCheng

\* date: 2022/7/21下午5:09

\*/

@Controller

@RequestMapping("/largeScreen")

public class Apico2eController {

@Autowired

IAsmModelService asmModelService;

@Autowired

IAsmCompanyService asmCompanyService;

@Autowired

IAsmModelBlendService asmModelBlendService;

@Autowired

IAsmModelDeviceOutputService asmModelDeviceOutputService;

@Autowired

IAsmModelDeviceOsblService asmModelDeviceOsblService;

@Autowired

IAsmOsblDictService asmOsblDictService;

@Autowired

ILargeScreenService largeScreenService;

@GetMapping()

@ResponseBody

public AjaxResult index(Integer year) {

Map<String, Object> resultMap = new HashMap<>();

List<Integer> years = getYears();

int currentYear = years.get(0);

if (year != null) {

currentYear = year;

}

Double totalOil = largeScreenService.selectTotalOil(currentYear);

Double totalCo2 = largeScreenService.selectTotalCO2(currentYear);

Double co2Factor = 0.0;

if (totalOil != null && totalCo2 != null) {

co2Factor = totalCo2 / totalOil;

}

Double totalCompanyOil = largeScreenService.selectUsedTotalOil(currentYear);

List<AsmCompany> companyList = asmCompanyService.selectUsedCompanyList();

List<Map<String, Double>> zoneCo2 = largeScreenService.selectZoneCo2(currentYear);

List<Map<String, Object>> modelCo2Factor = largeScreenService.selectCo2Factor(currentYear);

List<Map<String, Object>> listCo2 = largeScreenService.selectListCo2(currentYear);

List<Map<String, Object>> listCo2Factor = largeScreenService.selectListCo2Factor(currentYear);

List<Map<String, Object>> listProductCo2 = largeScreenService.selectListProductCo2(currentYear);

List<Map<String, Object>> listKeyProductCo2 = largeScreenService.selectKeyProduct(currentYear);

Map<String, Object> productCo2 = largeScreenService.selectTotalProduct(currentYear);

if (productCo2 == null) {

productCo2 = new HashMap<>();

}

Double totalProductCo2 = productCo2.get("co2") == null ? 0.0 : ((Double) productCo2.get("co2"));

Double totalProductPrice = productCo2.get("output\_value") == null ? 0.0 : ((Double) productCo2.get("output\_value"));

double productCo2Factor = 0.0;

if (totalProductPrice > 0.0) {

productCo2Factor = totalProductCo2 / totalProductPrice;

}

//未登陆数据加密

if (!SecurityUtils.loggedIn()) {

if (null != modelCo2Factor && !modelCo2Factor.isEmpty()) {

for (Map<String, Object> map :modelCo2Factor) {

map.put("co2\_factor", "\*未登录\*");

}

}

}

resultMap.put("years", years);

resultMap.put("current\_year", currentYear);

resultMap.put("totalOil", StringUtils.doubleToString("#0.00", (totalOil == null ? 0.0 : totalOil) / 10000.0));

resultMap.put("totalCo2", StringUtils.doubleToString("#0.00", (totalCo2 == null ? 0.0 : totalCo2) / 10000.0));

resultMap.put("co2Factor", StringUtils.doubleToString("#0.00", co2Factor));

resultMap.put("totalCompanyOil", StringUtils.doubleToString("#0.00", totalCompanyOil));

resultMap.put("comanyDictList", companyList);

resultMap.put("zoneCo2", zoneCo2);

resultMap.put("modelCo2Factor", modelCo2Factor);

resultMap.put("list\_key\_proudct\_co2", listKeyProductCo2);

resultMap.put("product\_co2\_factor", StringUtils.doubleToString("#0.00", productCo2Factor));

resultMap.put("min\_co2", getMapByIndex(listCo2, false, "model\_company\_name", "", "device\_co2", 0.0, null, null, null, null));

resultMap.put("max\_co2", getMapByIndex(listCo2, true, "model\_company\_name", "", "device\_co2", 2000000.0, null, null, null, null));

resultMap.put("min\_co2\_factor", getMapByIndex(listCo2Factor, false, "model\_company\_name", "", "device\_co2\_factor", 0.0, null, null, null, null));

resultMap.put("max\_co2\_factor", getMapByIndex(listCo2Factor, true, "model\_company\_name", "", "device\_co2\_factor", 10.0, null, null, null, null));

resultMap.put("min\_product\_factor", getMapByIndex(listProductCo2, false, "model\_company\_name", "", "output\_co2\_factor", 0.0, "output\_value", 0.0, "output\_co2", 0.0));

resultMap.put("max\_product\_factor", getMapByIndex(listProductCo2, true, "model\_company\_name", "", "output\_co2\_factor", 1, "output\_value", 1, "output\_co2", 1));

return AjaxResult.success(resultMap);

}

@GetMapping("/map")

@ResponseBody

public AjaxResult geoMap() {

Map<String, Object> resMap = new HashMap<>();

List<AsmCompany> companyList = asmCompanyService.selectAsmCompanyList(new AsmCompany());

resMap.put("comanyDictList", companyList);

// return prefix + "/map";

return AjaxResult.success(resMap);

}

@GetMapping("/model")

@ResponseBody

public Map<String, Object> viewModel(Integer year, String modelCompanyName) {

List<Integer> years = getYears();

int currentYear = years.get(0);

if (year != null) {

currentYear = year;

}

Map<String, Object> resultMap = new HashMap<>();

Double totalCo2 = largeScreenService.selectCo2ByComanyName(modelCompanyName, currentYear);

Double totalInput = largeScreenService.selectInputByComanyName(modelCompanyName, currentYear);

List<Map<String, Object>> productCo2 = largeScreenService.selectListProductCo2ByComanyName(modelCompanyName, currentYear);

Map<String, Object> totalProductCo2 = largeScreenService.selectProductCo2FactorByComanyName(modelCompanyName, currentYear);

if (totalCo2 == null) {

totalCo2 = 0.0;

}

double co2Factor;

if (totalInput == null) {

co2Factor = 0.0;

} else {

co2Factor = totalCo2 / totalInput;

}

resultMap.put("success", true);

resultMap.put("total\_co2", totalCo2);

resultMap.put("totalProductCo2", getTotalProductCo2(totalProductCo2));

resultMap.put("co2Factor", co2Factor);

resultMap.put("product\_co2", productCo2);

return resultMap;

}

@ResponseBody

@GetMapping("/viewmodel")

public AjaxResult getModelDatas(Integer year, String modelCompanyName) {

Map<String, Object> resultMap = new HashMap<>();

List<Map<String, Object>> productCo2 = new ArrayList<>();

List<Map<String, Object>> listCo2 = new ArrayList<>();

List<Map<String, Object>> listCo2Factor = new ArrayList<>();

List<Map<String, Object>> listProductCo2 = new ArrayList<>();

List<Integer> years = new ArrayList<>();

List<DeviceOsblVO> deviceOsblVOList = new ArrayList<>();

Map<String, Object> totalProductCo2 = null;

double co2Factor = 0.0;

double keo = 0.0;

double ecd = 0.0;

double ecu = 0.0;

double ect = 0.0;

double ecs = 0.0;

double eck = 0.0;

double ecr = 0.0;

double ecz = 0.0;

double ecj = 0.0;

Double totalCo2 = 0.0;

try {

years = getYears();

int currentYear = years.get(0);

if (year != null) {

currentYear = year;

}

Date beginDate = DateUtils.getYearFirst(currentYear);

Date endDate = DateUtils.getYearLast(currentYear);

AsmModel search = new AsmModel();

search.setModelCompany(modelCompanyName);

search.setBeginTime(beginDate);

search.setEndTime(endDate);

List<AsmModel> modelTempList = asmModelService.selectAsmModelList(search);

List<AsmOsblDict> dicts = asmOsblDictService.selectAsmOsblDictList(new AsmOsblDict());

Map<String, AsmOsblDict> osblDictMap = new HashMap<>();

for (AsmOsblDict dict : dicts) {

osblDictMap.put(dict.getDictCode(), dict);

}

Map<String, DeviceOsblVO> deviceOsblVOMap = new HashMap<>();

double modelDeviceCo2 = 0;

for (AsmModel model : modelTempList) {

AsmModel item = asmModelService.selectAsmModelByModelId(model.getModelId());

modelDeviceCo2 += item.getTotalFactoryCo2();

List<AsmModelDevice> devices = item.getDevices();

if (null == devices || devices.isEmpty()) {

continue;

}

for (AsmModelDevice device : devices) {

if (device.getIsVirtual() == 1) {

continue;

}

DeviceOsblVO vo = deviceOsblVOMap.get(device.getDeviceCode());

if (vo == null) {

vo = new DeviceOsblVO();

vo.setDeviceName(device.getDeviceName());

vo.setDeviceCode(device.getDeviceCode());

deviceOsblVOMap.put(device.getDeviceCode(), vo);

}

vo.setDeviceCo2(vo.getDeviceCo2() + device.getDeviceCo2());

if (modelDeviceCo2 == 0.0) {

vo.setDeviceCo2Percent(0.0);

} else {

vo.setDeviceCo2Percent((vo.getDeviceCo2() / modelDeviceCo2) \* 100.0);

}

vo.setDeviceProcess(vo.getDeviceProcess() + device.getDeviceProcess());

Map<String, AsmModelDeviceOsbl> osblMap = new HashMap<>();

for (AsmModelDeviceOsbl osbl : device.getOsbls()) {

osblMap.put(osbl.getOsblCode(), osbl);

AsmOsblDict dict = osblDictMap.get(osbl.getOsblCode());

keo += (osbl.getConsumption() == null ? 0 : osbl.getConsumption()) \* (dict.getFoldCoefficient() == null ? 0 : dict.getFoldCoefficient());

ecd += osbl.getCarbonEmission();

}

ecu = (osblMap.get("ECC") == null ? 0.0 : osblMap.get("ECC").getCarbonEmission())

+ (osblMap.get("CCK") == null ? 0.0 : osblMap.get("CCK").getCarbonEmission())

+ (osblMap.get("CKK") == null ? 0.0 : osblMap.get("CKK").getCarbonEmission())

+ (osblMap.get("EOO") == null ? 0.0 : osblMap.get("EOO").getCarbonEmission())

+ (osblMap.get("UFL") == null ? 0.0 : osblMap.get("UFL").getCarbonEmission())

+ (osblMap.get("GAS") == null ? 0.0 : osblMap.get("GAS").getCarbonEmission())

+ (osblMap.get("DIO") == null ? 0.0 : osblMap.get("DIO").getCarbonEmission())

+ (osblMap.get("LPS") == null ? 0.0 : osblMap.get("LPS").getCarbonEmission())

+ (osblMap.get("LTG") == null ? 0.0 : osblMap.get("LTG").getCarbonEmission())

+ (osblMap.get("UGS") == null ? 0.0 : osblMap.get("UGS").getCarbonEmission())

+ (osblMap.get("GAG") == null ? 0.0 : osblMap.get("GAG").getCarbonEmission())

+ (osblMap.get("NAG") == null ? 0.0 : osblMap.get("NAG").getCarbonEmission());

ect = osblMap.get("CKE") == null ? 0.0 : osblMap.get("CKE").getCarbonEmission();

ecs = (osblMap.get("10M") == null ? 0.0 : osblMap.get("10M").getCarbonEmission())

+ (osblMap.get("05M") == null ? 0.0 : osblMap.get("05M").getCarbonEmission())

+ (osblMap.get("3D5") == null ? 0.0 : osblMap.get("3D5").getCarbonEmission())

+ (osblMap.get("2D5") == null ? 0.0 : osblMap.get("2D5").getCarbonEmission())

+ (osblMap.get("1D5") == null ? 0.0 : osblMap.get("1D5").getCarbonEmission())

+ (osblMap.get("1DM") == null ? 0.0 : osblMap.get("1DM").getCarbonEmission())

+ (osblMap.get("D7M") == null ? 0.0 : osblMap.get("D7M").getCarbonEmission())

+ (osblMap.get("D5M") == null ? 0.0 : osblMap.get("D5M").getCarbonEmission())

+ (osblMap.get("LD3") == null ? 0.0 : osblMap.get("LD3").getCarbonEmission());

eck = (osblMap.get("KWS") == null ? 0.0 : osblMap.get("KWS").getCarbonEmission())

+ (osblMap.get("KWH") == null ? 0.0 : osblMap.get("KWH").getCarbonEmission());

ecr = (osblMap.get("WAT") == null ? 0.0 : osblMap.get("WAT").getCarbonEmission())

+ (osblMap.get("CWT") == null ? 0.0 : osblMap.get("CWT").getCarbonEmission())

+ (osblMap.get("SWT") == null ? 0.0 : osblMap.get("SWT").getCarbonEmission())

+ (osblMap.get("DAW") == null ? 0.0 : osblMap.get("DAW").getCarbonEmission())

+ (osblMap.get("DOW") == null ? 0.0 : osblMap.get("DOW").getCarbonEmission())

+ (osblMap.get("COT") == null ? 0.0 : osblMap.get("COT").getCarbonEmission())

+ (osblMap.get("COH") == null ? 0.0 : osblMap.get("COH").getCarbonEmission())

+ (osblMap.get("NNS") == null ? 0.0 : osblMap.get("NNS").getCarbonEmission());

ecz = ecu + ect;

ecj = ecr + eck + ecs;

keo += vo.getOsbls().get("keo") == null ? 0.0 : vo.getOsbls().get("keo");

ecd += vo.getOsbls().get("ecd") == null ? 0.0 : vo.getOsbls().get("ecd");

ecu += vo.getOsbls().get("ecu") == null ? 0.0 : vo.getOsbls().get("ecu");

ect += vo.getOsbls().get("ect") == null ? 0.0 : vo.getOsbls().get("ect");

ecs += vo.getOsbls().get("ecs") == null ? 0.0 : vo.getOsbls().get("ecs");

eck += vo.getOsbls().get("eck") == null ? 0.0 : vo.getOsbls().get("eck");

ecr += vo.getOsbls().get("ecr") == null ? 0.0 : vo.getOsbls().get("ecr");

ecz += vo.getOsbls().get("ecz") == null ? 0.0 : vo.getOsbls().get("ecz");

ecj += vo.getOsbls().get("ecj") == null ? 0.0 : vo.getOsbls().get("ecj");

vo.getOsbls().put("keo", keo);

vo.getOsbls().put("ecd", ecd);

vo.getOsbls().put("ecu", ecu);

vo.getOsbls().put("ect", ect);

vo.getOsbls().put("ecs", ecs);

vo.getOsbls().put("eck", eck);

vo.getOsbls().put("ecr", ecr);

vo.getOsbls().put("ecz", ecz);

vo.getOsbls().put("ecj", ecj);

}

}

totalCo2 = largeScreenService.selectCo2ByComanyName(modelCompanyName, currentYear);

Double totalInput = largeScreenService.selectInputByComanyName(modelCompanyName, currentYear);

totalProductCo2 = largeScreenService.selectProductCo2FactorByComanyName(modelCompanyName, currentYear);

listCo2 = largeScreenService.selectListCo2(currentYear);

listCo2Factor = largeScreenService.selectListCo2Factor(currentYear);

listProductCo2 = largeScreenService.selectListProductCo2(currentYear);

if (totalCo2 == null) {

totalCo2 = 0.0;

}

if (totalInput == null) {

co2Factor = 0.0;

} else {

co2Factor = totalCo2 / totalInput;

}

deviceOsblVOList.addAll(deviceOsblVOMap.values());

deviceOsblVOList.sort(Comparator.comparing(DeviceOsblVO::getDeviceCo2).reversed());

productCo2 = largeScreenService.selectListProductCo2ByComanyName(modelCompanyName, currentYear);

for (DeviceOsblVO vo : deviceOsblVOList) {

ecz += vo.getOsbls().get("ecz");

ecj += vo.getOsbls().get("ecj");

ecd += vo.getOsbls().get("ecd");

ecu += vo.getOsbls().get("ecu");

ect += vo.getOsbls().get("ect");

ecs += vo.getOsbls().get("ecs");

eck += vo.getOsbls().get("eck");

ecr += vo.getOsbls().get("ecr");

vo.setDeviceCo2Factor(vo.getDeviceCo2() / vo.getDeviceProcess());

}

resultMap.put("years", years);

resultMap.put("currentYear", currentYear);

resultMap.put("modelCompanyName", modelCompanyName);

resultMap.put("devices", deviceOsblVOList);

resultMap.put("ecz", ecz);

resultMap.put("ecj", ecj);

resultMap.put("ecd", ecd);

resultMap.put("ecu", ecu);

resultMap.put("ect", ect);

resultMap.put("ecs", ecs);

resultMap.put("eck", eck);

resultMap.put("ecr", ecr);

resultMap.put("totalCo2", totalCo2);

resultMap.put("totalProductCo2", getTotalProductCo2(totalProductCo2));

resultMap.put("co2Factor", co2Factor);

resultMap.put("productCo2", productCo2);

resultMap.put("min\_co2", getMapByIndex(listCo2, false, "model\_company\_name", "", "device\_co2", 0.0, null, null, null, null));

resultMap.put("max\_co2", getMapByIndex(listCo2, true, "model\_company\_name", "", "device\_co2", 2000000.0, null, null, null, null));

resultMap.put("min\_co2\_factor", getMapByIndex(listCo2Factor, false, "model\_company\_name", "", "device\_co2\_factor", 0.0, null, null, null, null));

resultMap.put("max\_co2\_factor", getMapByIndex(listCo2Factor, true, "model\_company\_name", "", "device\_co2\_factor", 10.0, null, null, null, null));

resultMap.put("min\_product\_factor", getMapByIndex(listProductCo2, false, "model\_company\_name", "", "output\_co2\_factor", 0.0, "output\_value", 0.0, "output\_co2", 0.0));

resultMap.put("max\_product\_factor", getMapByIndex(listProductCo2, true, "model\_company\_name", "", "output\_co2\_factor", 1, "output\_value", 1, "output\_co2", 1));

return AjaxResult.success(resultMap);

} catch (Exception e) {

resultMap.put("years", years);

resultMap.put("currentYear", year);

resultMap.put("modelCompanyName", modelCompanyName);

resultMap.put("devices", deviceOsblVOList);

resultMap.put("ecz", ecz);

resultMap.put("ecj", ecj);

resultMap.put("ecd", ecd);

resultMap.put("ecu", ecu);

resultMap.put("ect", ect);

resultMap.put("ecs", ecs);

resultMap.put("eck", eck);

resultMap.put("ecr", ecr);

resultMap.put("totalCo2", totalCo2);

resultMap.put("totalProductCo2", getTotalProductCo2(totalProductCo2));

resultMap.put("co2Factor", 0.0);

resultMap.put("productCo2", productCo2);

resultMap.put("min\_co2", getMapByIndex(listCo2, false, "model\_company\_name", "", "device\_co2", 0.0, null, null, null, null));

resultMap.put("max\_co2", getMapByIndex(listCo2, true, "model\_company\_name", "", "device\_co2", 2000000.0, null, null, null, null));

resultMap.put("min\_co2\_factor", getMapByIndex(listCo2Factor, false, "model\_company\_name", "", "device\_co2\_factor", 0.0, null, null, null, null));

resultMap.put("max\_co2\_factor", getMapByIndex(listCo2Factor, true, "model\_company\_name", "", "device\_co2\_factor", 10.0, null, null, null, null));

resultMap.put("min\_product\_factor", getMapByIndex(listProductCo2, false, "model\_company\_name", "", "output\_co2\_factor", 0.0, "output\_value", 0.0, "output\_co2", 0.0));

resultMap.put("max\_product\_factor", getMapByIndex(listProductCo2, true, "model\_company\_name", "", "output\_co2\_factor", 1, "output\_value", 1, "output\_co2", 1));

return AjaxResult.error("获取部分数据失败，请联系管理员", resultMap);

}

}

/\*\*

\* 获取模型中的最大和最小年份

\* @return

\*/

private List<Integer> getYears() {

Map<String, Date> dateRange = asmModelService.selectDateRange();

int currYear = DateUtils.getCurrYear();

int minYear = currYear - 2;

int maxYear = currYear;

if (null != dateRange) {

Date minDate = dateRange.get("min\_date");

Date maxDate = dateRange.get("max\_date");

if (null != minDate) {

minYear = DateUtils.getYear(minDate);

}

if (null != maxDate) {

maxYear = DateUtils.getYear(maxDate);

}

}

List<Integer> years = new ArrayList<>();

for (int y = maxYear; y >= minYear; y--) {

years.add(y);

}

return years;

}

private Double getTotalProductCo2(Map<String, Object> totalProductCo2) {

if (totalProductCo2 == null) {

return 0.0;

} else {

Object outputFacotr = totalProductCo2.get("output\_facotr");

if (outputFacotr == null) {

return 0.0;

} else {

return (double) outputFacotr;

}

}

}

/\*\*

\* 根据list 取数据

\* @param list

\* @param max 是否是最大 true 取下标为list.size()-1的，false取index为0的

\* @param k1 map的默认key

\* @param v1 map的默认值

\* @param k2

\* @param v2

\* @param k3

\* @param v3

\* @param k4

\* @param v4

\* @return

\*/

private Map<String, Object> getMapByIndex(List<Map<String, Object>> list,

Boolean max,

String k1, Object v1,

String k2, Object v2,

String k3, Object v3,

String k4, Object v4) {

Map<String, Object> map = new HashMap<>();

if (list == null || list.isEmpty()) {

map.put(k1, v1);

map.put(k2, v2);

if (k3 != null) {

map.put(k3, v3);

}

if (k4 != null) {

map.put(k4, v4);

}

} else {

map = list.get(max ? (list.size() - 1) : 0);

}

return map;

}

}

package com.jiexun.api.controller;

import com.alibaba.fastjson2.JSONObject;

import com.jiexun.api.constants.CmsConstants;

import com.jiexun.api.vo.ContentVO;

import com.jiexun.api.vo.RotationChartContentVO;

import com.jiexun.cms.domain.\*;

import com.jiexun.cms.service.\*;

import com.jiexun.common.constant.HttpStatus;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.core.domain.model.LoginUser;

import com.jiexun.common.core.page.PageDomain;

import com.jiexun.common.core.page.TableDataInfo;

import com.jiexun.common.core.page.TableSupport;

import com.jiexun.common.service.ElasticsearchService;

import com.jiexun.common.utils.DateUtils;

import com.jiexun.common.utils.StringUtils;

import com.jiexun.common.utils.uuid.UUID;

import com.jiexun.system.domain.SysFileInfo;

import com.jiexun.system.service.ISysFileInfoService;

import com.jiexun.system.service.ISysRoleService;

import org.apache.commons.logging.Log;

import org.apache.commons.logging.LogFactory;

import org.elasticsearch.action.get.GetResponse;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.io.IOException;

import java.util.Collections;

import java.util.List;

import java.util.stream.Collectors;

/\*\*

\* 内容管理Api

\* @author yzc

\* @date 2022-07-26

\*/

@RestController

@RequestMapping("/cmsApi")

public class CmsApiController extends BaseController {

@Autowired

private ICmsContentService cmsContentService;

@Autowired

private ElasticsearchService elasticsearchService;

@Autowired

private ICmsChannelService cmsChannelService;

@Autowired

private ICmsTemplateService cmsTemplateService;

@Autowired

private ICmsMemberGroupService cmsMemberGroupService;

@Autowired

private ICmsCommentService cmsCommentService;

@Autowired

private ISysRoleService roleService;

@Autowired

private ISysFileInfoService sysFileInfoService;

private final Log logger = LogFactory.getLog(CmsApiController.class);

/\*\*

\* 查询栏目列表

\*/

@GetMapping("/getListChannel")

public TableDataInfo getListChannel(CmsChannel cmsChannel) {

cmsChannel.setDelState(CmsConstants.DEL\_STATE\_FALSE);

cmsChannel.setParentId(0L);

List<CmsChannel> list = cmsChannelService.selectCmsChannelList(cmsChannel);

return getDataTable(list);

}

/\*\*

\* 查询子栏目列表

\*/

@GetMapping("/getChildChannel/{channelId}")

public TableDataInfo getChildChannel(@PathVariable Long channelId) {

CmsChannel cmsChannel = new CmsChannel();

cmsChannel.setDelState(CmsConstants.DEL\_STATE\_FALSE);

cmsChannel.setParentId(channelId);

List<CmsChannel> list = cmsChannelService.selectCmsChannelList(cmsChannel);

return getDataTable(list);

}

/\*\*

\* 查询内容列表

\*/

@GetMapping("/getListContent")

public TableDataInfo list(CmsContent cmsContent) {

PageDomain pageDomain = TableSupport.buildPageRequest();

Integer pageNum = pageDomain.getPageNum();

Integer pageSize = pageDomain.getPageSize();

cmsContent.setReleaseStatus(CmsConstants.RELEASE\_STATE\_TRUE);

cmsContent.setAuditStatus(CmsConstants.AUDIT\_STATE\_PASS);

cmsContent.setDelState(CmsConstants.DEL\_STATE\_FALSE);

Long channelId = cmsContent.getChannelId();

if (null != channelId) {

CmsChannel cmsChannel = cmsChannelService.selectCmsChannelById(channelId);

Integer channelMemberGroupId = cmsChannel.getMemberGroupId();

if (null != channelMemberGroupId && channelMemberGroupId > 0) {

LoginUser loginUser;

try {

loginUser = getLoginUser();

} catch (Exception e) {

return getUnLoginTable();

}

if (null == loginUser

|| loginUser.getUser() == null

|| StringUtils.isEmpty(loginUser.getUsername())

|| null == loginUser.getUserId()) {

return getUnLoginTable();

}

try {

CmsMemberGroup channelMemberGroup = cmsMemberGroupService.selectCmsMemberGroupById(channelMemberGroupId);

CmsMemberGroup param = new CmsMemberGroup();

param.setAppId(cmsChannel.getAppId());

param.setUserId(loginUser.getUserId());

Integer userLevel = cmsMemberGroupService.getLevel(param);

Integer channelLevel = channelMemberGroup.getLevel();

if (channelLevel != null && userLevel < channelLevel) {

return getErrorTable();

}

} catch (Exception e) {

return getErrorTable();

}

}

}

if (StringUtils.isNotEmpty(cmsContent.getGlobalQuery())) {

//添加查询日志

SearchWord searchWord = new SearchWord();

String dateToStr = DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD\_HH\_MM\_SS, DateUtils.getNowDate());

String uuid = UUID.getUuid();

searchWord.setId(uuid);

searchWord.setAppId(cmsContent.getAppId());

searchWord.setChannelId(cmsContent.getChannelId());

try {

searchWord.setCreateBy(getUsername());

} catch (Exception e) {

logger.debug("用户未登陆！");

}

searchWord.setCreateTime(dateToStr);

searchWord.setSearchText(cmsContent.getGlobalQuery());

try {

elasticsearchService.addDocument(searchWord, CmsConstants.SEARCH\_WORD\_INDEX, uuid);

} catch (IOException e) {

e.printStackTrace();

logger.info("添加搜索记录失败！");

}

}

TableDataInfo result = cmsContentService.selectCmsContentPageList(cmsContent, pageNum, pageSize);

List<CmsContent> rows = (List<CmsContent>) result.getRows();

if (!rows.isEmpty()) {

List<ContentVO> voList = rows.stream().map(c -> ContentVO.convert(c)).collect(Collectors.toList());

result.setRows(voList);

}

return result;

}

/\*\*

\* 查询热词列表

\* @param appId

\* @param channelId

\* @return arr

\*/

@GetMapping("/getHotWords")

public AjaxResult getHotWords(Long appId, Long channelId) {

List<JSONObject> hotWords = elasticsearchService.getHotWords(CmsConstants.SEARCH\_WORD\_INDEX, appId, channelId);

if (null == hotWords) {

return AjaxResult.success(Collections.EMPTY\_LIST);

}

return AjaxResult.success(hotWords);

}

/\*\*

\* 查询轮播图内容列表

\*/

@GetMapping("/getListRotationChart")

public TableDataInfo listRotationChart(RotationChartContentVO searchParam) {

TableDataInfo info = new TableDataInfo();

info.setMsg("参数错误！");

info.setCode(500);

if (null == searchParam) {

return info;

}

Long[] channelIds = searchParam.getChannelIds();

if (null == channelIds || channelIds.length == 0) {

return info;

}

Integer userLevel = 0;

if (null != channelIds) {

List<CmsChannel> cmsChannels = cmsChannelService.selectCmsChannelByIds(channelIds);

List<CmsMemberGroup> cmsMemberGroups = cmsMemberGroupService.selectCmsMemberGroupList(new CmsMemberGroup());

if (null != cmsMemberGroups && !cmsMemberGroups.isEmpty()) {

for (CmsChannel cmsChannel : cmsChannels) {

Integer channelMemberGroupId = cmsChannel.getMemberGroupId();

if (null != channelMemberGroupId && channelMemberGroupId > 0) {

Integer groupLevel = getGroupLevel(cmsMemberGroups, channelMemberGroupId);

LoginUser loginUser;

try {

loginUser = getLoginUser();

CmsMemberGroup param = new CmsMemberGroup();

param.setAppId(cmsChannel.getAppId());

param.setUserId(loginUser.getUserId());

userLevel = cmsMemberGroupService.getLevel(param);

} catch (Exception e) {

return getUnLoginTable();

}

if (null == loginUser

|| loginUser.getUser() == null

|| StringUtils.isEmpty(loginUser.getUsername())

|| null == loginUser.getUserId()) {

return getUnLoginTable();

}

if (groupLevel != null && userLevel < groupLevel) {

return getErrorTable();

}

}

}

}

}

searchParam.setReleaseStatus(CmsConstants.RELEASE\_STATE\_TRUE);

searchParam.setAuditStatus(CmsConstants.AUDIT\_STATE\_PASS);

searchParam.setDelState(CmsConstants.DEL\_STATE\_FALSE);

searchParam.setRotationChart(CmsConstants.ROTATION\_CHART\_TRUE);

info = cmsContentService.listRotationChart(searchParam);

return info;

}

/\*\*

\* 获取内容管理详细信息

\*/

@GetMapping(value = "/getContent/{uuid}")

public AjaxResult getContent(@PathVariable("uuid") String uuid) {

try {

GetResponse document = elasticsearchService.getDocument(CmsConstants.CONTENT\_INDEX, uuid);

CmsContent content = JSONObject.parseObject(JSONObject.toJSONString(document.getSourceAsMap()),

CmsContent.class);

content.setUuid(uuid);

// 验证权限

AjaxResult permissionResult = checkPermissions(content.getPermissionType(), content.getAppId(), content.getMemberGroupId(), content.getRoleId());

if (!permissionResult.get(AjaxResult.CODE\_TAG).equals(HttpStatus.SUCCESS)) {

return permissionResult;

}

String[] fileIds = content.getOtherFiles();

if (null != fileIds && fileIds.length > 0 ) {

List<SysFileInfo> fileList = sysFileInfoService.selectFileListByIds(fileIds);

if (null != fileList) {

content.setOtherFileList(fileList);

}

}

//阅读次数加1

Long readCount = content.getReadCount();

content.setReadCount(null == readCount ? 1L : (readCount + 1));

try {

elasticsearchService.updateDocument(CmsConstants.CONTENT\_INDEX, content.getUuid(), content);

} catch (Exception e) {

//修改阅读次数失败

e.printStackTrace();

}

return AjaxResult.success(content);

} catch (IOException e) {

e.printStackTrace();

}

return AjaxResult.error();

}

/\*\*

\* 根据ContentId获取模版详细信息

\*/

@GetMapping(value = "/getTempByContentId/{uuid}")

public AjaxResult getTempByContentId(@PathVariable("uuid") String uuid) {

GetResponse document = null;

try {

document = elasticsearchService.getDocument(CmsConstants.CONTENT\_INDEX, uuid);

} catch (IOException e) {

e.printStackTrace();

}

CmsContent content = JSONObject.parseObject(JSONObject.toJSONString(document.getSourceAsMap()),

CmsContent.class);

// 验证权限

AjaxResult permissionResult = checkPermissions(content.getPermissionType(), content.getAppId(), content.getMemberGroupId(), content.getRoleId());

if (!permissionResult.get(AjaxResult.CODE\_TAG).equals(HttpStatus.SUCCESS)) {

return permissionResult;

}

CmsTemplate contentTemplate = cmsTemplateService.selectCmsTemplateById(content.getContentTempId());

if (null == contentTemplate) {

CmsChannel cmsChannel = cmsChannelService.selectCmsChannelById(content.getChannelId());

if (null != cmsChannel) {

contentTemplate = cmsTemplateService.selectCmsTemplateById(cmsChannel.getContentTempId());

}

}

if (null == contentTemplate) {

return AjaxResult.error("获取模版信息失败！");

}

return AjaxResult.success(contentTemplate);

}

/\*\*

\* 新增评论

\*/

@PostMapping("/comment/add")

public AjaxResult addComment(@RequestBody CmsComment cmsComment) {

CmsComment parentComment = cmsCommentService.selectCmsCommentById(cmsComment.getParentId());

String loginName = null;

try {

LoginUser loginUser = getLoginUser();

if (null == loginUser) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

loginName = loginUser.getUsername();

} catch (Exception e) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

if (null != parentComment) {

String createBy = parentComment.getCreateBy();

cmsComment.setpCreateBy(createBy);

cmsComment.setLevel(parentComment.getLevel() + 1);

if (parentComment.getLevel() <= 1) {

cmsComment.setBlogId(parentComment.getId());

} else {

cmsComment.setBlogId(parentComment.getBlogId());

}

if (null != createBy && createBy.equals(loginName)) {

return AjaxResult.error("不能回复自己的评论！");

}

} else {

cmsComment.setLevel(1);

cmsComment.setBlogId(cmsComment.getMainId());

}

cmsComment.setCreateBy(loginName);

cmsComment.setUserId(getUserId());

return toAjax(cmsCommentService.insertCmsComment(cmsComment));

}

/\*\*

\* 查询评论列表

\*/

@GetMapping("/comment/list")

public TableDataInfo getCommentList(CmsComment cmsComment) {

startPage();

LoginUser loginUser = null;

try {

cmsComment.setCreateBy(getUsername());

cmsComment.setUserId(getUserId());

loginUser = getLoginUser();

} catch (Exception e) {

}

List<CmsComment> parentList = cmsCommentService.selectList(cmsComment);

TableDataInfo dataTable = getDataTable(parentList);

if (!parentList.isEmpty()) {

for (CmsComment comment : parentList) {

cmsCommentService.addChildList(comment, loginUser);

}

}

return dataTable;

}

@GetMapping("/comment/delete/{id}")

public AjaxResult removeComment(@PathVariable String id) {

CmsComment cmsComment = cmsCommentService.selectCmsCommentById(id);

if (null == cmsComment) {

return AjaxResult.error("参数错误");

}

String loginUserName = null;

try {

loginUserName = getLoginUser().getUsername();

} catch (Exception e) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

if (null == loginUserName) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

if (cmsComment.getCreateBy().equals(loginUserName)) {

return toAjax(cmsCommentService.deleteById(id));

} else {

return AjaxResult.error("不是自己的评论不能删除！");

}

}

/\*\*

\* 首页新增点赞

\*/

@GetMapping("/commentLike/add/{commentId}")

public AjaxResult addCmsCommentLike(@PathVariable String commentId) {

String loginUserName = null;

try {

loginUserName = getLoginUser().getUsername();

} catch (Exception e) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

if (null == loginUserName) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

CmsCommentLike cmsCommentLike = new CmsCommentLike();

cmsCommentLike.setCommentId(commentId);

cmsCommentLike.setUserId(getUserId());

cmsCommentLike.setCreateBy(getUsername());

cmsCommentLike.setCreateTime(DateUtils.getNowDate());

return toAjax(cmsCommentService.addCmsCommentLike(cmsCommentLike));

}

@GetMapping("/commentLike/delete/{commentId}")

public AjaxResult delCmsCommentLike(@PathVariable String commentId) {

CmsCommentLike cmsCommentLike = new CmsCommentLike();

Long loginUserId = null;

try {

loginUserId = getLoginUser().getUserId();

} catch (Exception e) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

if (null == loginUserId) {

return AjaxResult.error(402, "您的操作需要登陆！");

}

cmsCommentLike.setCommentId(commentId);

cmsCommentLike.setUserId(loginUserId);

return toAjax(cmsCommentService.delCmsCommentLike(cmsCommentLike));

}

private TableDataInfo getUnLoginTable() {

TableDataInfo dataInfo = new TableDataInfo();

dataInfo.setCode(402);

dataInfo.setMsg("您需要登陆访问！");

return dataInfo;

}

private TableDataInfo getErrorTable() {

TableDataInfo dataInfo = new TableDataInfo();

dataInfo.setCode(403);

dataInfo.setMsg("您没有访问该内容的权限！");

return dataInfo;

}

private AjaxResult unLoginAjaxResult() {

return AjaxResult.error(402, "非公开信息，需要登陆访问！");

}

private AjaxResult errorAjaxResult() {

return AjaxResult.error(403, "您没有访问该内容的权限！");

}

/\*\*

\* 根据权限验证类型验证当前用户是否有权限

\* @param permissionTypes 权限验证类型[]

\* @param appId 应用id

\* @param memberGroupId 授权的会员组id

\* @param roleIds 授权的角色id

\* @return

\*/

private AjaxResult checkPermissions(Integer[] permissionTypes, Long appId, Integer memberGroupId, Long[] roleIds) {

if (null != permissionTypes && permissionTypes.length > 0

&& !hasValue(permissionTypes, CmsConstants.PERMISSION\_TYPE\_ALL)) {

//验证用户权限

LoginUser loginUser;

try {

loginUser = getLoginUser();

} catch (Exception e) {

return unLoginAjaxResult();

}

if (null == loginUser

|| loginUser.getUser() == null

|| StringUtils.isEmpty(loginUser.getUsername())

|| null == loginUser.getUserId()) {

return unLoginAjaxResult();

}

try {

//验证会员等级和角色是否可以查看内容

boolean hasPermission = false;

if (hasValue(permissionTypes, CmsConstants.PERMISSION\_TYPE\_MEMBER)) {

CmsMemberGroup contentMemberGroup = cmsMemberGroupService.selectCmsMemberGroupById(memberGroupId);

CmsMemberGroup param = new CmsMemberGroup();

param.setAppId(appId);

param.setUserId(loginUser.getUserId());

Integer userLevel = cmsMemberGroupService.getLevel(param);

userLevel = userLevel == null ? 0 : userLevel;

Integer contentLevel = contentMemberGroup.getLevel();

if (contentLevel != null && userLevel >= contentLevel) {

//当前用户的会员等级可以看内容

hasPermission = true;

}

}

//会员等级不足时再验证角色

if (!hasPermission && hasValue(permissionTypes, CmsConstants.PERMISSION\_TYPE\_ROLE)) {

List<Long> userRoleIds = roleService.selectRoleListByUserId(loginUser.getUserId());

if (!userRoleIds.isEmpty() && roleIds.length > 0) {

for (Long roleId : roleIds) {

if (userRoleIds.contains(roleId)) {

hasPermission = true;

break;

}

}

}

}

if (!hasPermission) {

return errorAjaxResult();

}

} catch (Exception e) {

return errorAjaxResult();

}

}

return AjaxResult.success();

}

/\*\*

\* 获取会员组的等级

\* @param list

\* @param groupId

\* @return

\*/

private Integer getGroupLevel(List<CmsMemberGroup> list, Integer groupId) {

for (CmsMemberGroup group : list) {

if (group.getId().equals(groupId)) {

return group.getLevel();

}

}

return 0;

}

/\*\*

\* 判断数组是否包含某个值

\* @param arr

\* @param value

\* @return

\*/

private boolean hasValue(Integer[] arr, Integer value) {

if (arr != null && arr.length > 0) {

for (Integer i : arr) {

if (i.equals(value)) {

return true;

}

}

}

return false;

}

}

package com.jiexun.api.controller;

import com.alibaba.fastjson2.JSON;

import com.alibaba.fastjson2.JSONObject;

import com.jiexun.api.constants.CmsConstants;

import com.jiexun.api.vo.TechnicalClassificationVO;

import com.jiexun.cms.domain.CmsConfig;

import com.jiexun.cms.service.ICmsConfigService;

import com.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.core.domain.entity.SysDictData;

import com.jiexun.common.core.page.TableDataInfo;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.common.utils.StringUtils;

import com.jiexun.lowcarbon.domain.Expert;

import com.jiexun.lowcarbon.domain.TechnicalClassification;

import com.jiexun.lowcarbon.domain.Technology;

import com.jiexun.lowcarbon.service.IExpertService;

import com.jiexun.lowcarbon.service.ITechnicalClassificationService;

import com.jiexun.lowcarbon.service.ITechnologyService;

import com.jiexun.system.domain.SysFileInfo;

import com.jiexun.system.service.ISysDictTypeService;

import com.jiexun.system.service.ISysFileInfoService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.ArrayList;

import java.util.List;

/\*\*

\* api . <br>

\*

\* @author YanZhiCheng

\* date: 2022/8/29下午5:19

\*/

@RestController

@RequestMapping("/lowCarbonApi")

public class LowCarbonApiController extends BaseController {

@Autowired

private ITechnicalClassificationService technicalClassificationService;

@Autowired

private ITechnologyService technologyService;

@Autowired

private IExpertService expertService;

@Autowired

private ISysDictTypeService dictTypeService;

@Autowired

private ICmsConfigService cmsConfigService;

@Autowired

private ISysFileInfoService sysFileInfoService;

/\*\*

\* 获取低碳技术验证的配置接口

\*/

@GetMapping("/getConfig/{lable}")

public AjaxResult getConfig(@PathVariable String lable) {

CmsConfig cmsConfig = cmsConfigService.selectCmsConfigByAppId(1L);

JSONObject jsonObject = JSONObject.parseObject(JSON.toJSONString(cmsConfig));

return AjaxResult.success(jsonObject.get(lable));

}

/\*\*

\* 根据字典类型查询字典数据信息

\*/

@GetMapping(value = "/dictList/{type}")

public AjaxResult dictType(@PathVariable String type) {

List<SysDictData> data = dictTypeService.selectDictDataByType(type);

if (StringUtils.isNull(data)) {

data = new ArrayList<SysDictData>();

}

return AjaxResult.success(data);

}

/\*\*

\* 查询专家库列表

\*/

@GetMapping("/listExpert")

public TableDataInfo listExpert(Expert expert) {

startPage();

List<Expert> list = expertService.selectExpertList(expert);

return getDataTable(list);

}

/\*\*

\* 获取专家库详细信息

\*/

@GetMapping(value = "/getExpertInfo/{id}")

public AjaxResult getExpertInfo(@PathVariable("id") Long id) {

return AjaxResult.success(expertService.selectExpertById(id));

}

/\*\*

\* 获取专家职务分布统计

\*/

@GetMapping("/jobDistributionAnalysis")

public AjaxResult jobDistributionAnalysis() {

List<JSONObject> analysisData = expertService.jobDistributionAnalysis();

return AjaxResult.success(analysisData);

}

/\*\*

\* 获取专家专业领域分布统计

\*/

@GetMapping("/professionalFieldsDistribution")

public AjaxResult professionalFieldsDistribution() {

List<JSONObject> analysisData = expertService.professionalFieldsDistribution();

return AjaxResult.success(analysisData);

}

/\*\*

\* 查询技术分类列表

\*/

@GetMapping("/listTechnicalClass")

public TableDataInfo listTechnicalClass(TechnicalClassification technicalClassification) {

technicalClassification.setDelState(CmsConstants.DEL\_STATE\_FALSE);

List<TechnicalClassification> list = technicalClassificationService.selectTechnicalClassificationList(technicalClassification);

List<TechnicalClassificationVO> voList = new ArrayList<>();

for (TechnicalClassification obj: list) {

if(StringUtils.isNotEmpty(obj.getFiles())) {

String[] fileIds = obj.getFiles().split(",");

if (fileIds.length > 0) {

List<SysFileInfo> fileInfos = sysFileInfoService.selectFileListByIds(fileIds);

if (null != fileInfos) {

obj.setFileInfoList(fileInfos);

}

}

}

TechnicalClassificationVO vo = TechnicalClassificationVO.convert(obj);

voList.add(vo);

}

return getDataTable(voList);

}

/\*\*

\* 获取应用程度分析柱状图数据

\*/

@GetMapping("/getAppDegreeAnalysis")

public AjaxResult getAppDegreeAnalysis() {

List<JSONObject> analysisData = technicalClassificationService.getAppDegreeAnalysis();

return AjaxResult.success(analysisData);

}

/\*\*

\* 获取应用评价等级分析柱状图数据

\*/

@GetMapping("/getAppLevelAnalysis")

public AjaxResult getAppLevelAnalysis() {

List<JSONObject> analysisData = technicalClassificationService.getAppLevelAnalysis();

return AjaxResult.success(analysisData);

}

/\*\*

\* 查询技术库列表

\*

\* @param technology

\* @return

\*/

@GetMapping("/listTechnology")

public TableDataInfo listTechnology(Technology technology) {

startPage();

technology.setDelState(CmsConstants.DEL\_STATE\_FALSE);

List<Technology> list = technologyService.selectTechnologyList(technology);

return getDataTable(list);

}

/\*\*

\* 查询自己发布的技术库列表

\* @param technology

\* @return

\*/

@GetMapping("/listTechnologyByLoginName")

public TableDataInfo listTechnologyByLoginName(Technology technology) {

technology.setDelState(CmsConstants.DEL\_STATE\_FALSE);

technology.setCreateBy(this.getUsername());

List<Technology> list = technologyService.selectMyTechnologyList(technology);

return getDataTable(list);

}

/\*\*

\* 查询置顶的技术库

\*

\* @return

\*/

@GetMapping("/listTopTechnology")

public TableDataInfo listTopTechnology() {

List<Technology> list = technologyService.selectTopTechnologyList();

return getDataTable(list);

}

/\*\*

\* 查询按搜索量排序的技术库

\*

\* @return

\*/

@GetMapping("/listHotTechnology")

public TableDataInfo listHotTechnology() {

List<Technology> list = technologyService.selectHotTechnologyList();

return getDataTable(list);

}

/\*\*

\* 新增技术库

\*

\* @param technology

\* @return

\*/

@Log(title = "技术库", businessType = BusinessType.INSERT)

@PostMapping("/addTechnology")

public AjaxResult addTechnology(@RequestBody Technology technology) {

try {

technology.setCreateBy(getUsername());

} catch (Exception e) {

return AjaxResult.error("未登陆！");

}

return toAjax(technologyService.insertTechnology(technology));

}

/\*\*

\* 修改技术库

\*

\* @param technology

\* @return

\*/

@PutMapping("/updateTechnology")

public AjaxResult updateTechnology(@RequestBody Technology technology) {

return toAjax(technologyService.updateTechnology(technology));

}

/\*\*

\* 删除技术库

\*

\* @param uuids

\* @return

\*/

@Log(title = "技术库", businessType = BusinessType.DELETE)

@DeleteMapping("/deleteTechnology/{uuids}")

public AjaxResult deleteTechnology(@PathVariable String[] uuids) {

return toAjax(technologyService.deleteTechnologyByUuids(uuids));

}

/\*\*

\* 获取技术库详细信息

\*

\* @param uuid

\* @return

\*/

@GetMapping(value = "/getTechnologyInfo/{uuid}")

public AjaxResult getTechnologyInfo(@PathVariable("uuid") String uuid) {

return AjaxResult.success(technologyService.selectTechnologyByUuid(uuid));

}

}

package com.jiexun.api.mapper;

import org.apache.ibatis.annotations.Param;

import org.springframework.stereotype.Repository;

import java.util.Date;

import java.util.List;

import java.util.Map;

@Repository

public interface BigDataMapper {

public Double selectTotalOil(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Double selectTotalCo2(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Double selectUsedTotalOil(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Double>> selectZoneCo2(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectCo2Factor(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectListCo2(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectListCo2Factor(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Map<String, Object> selectTotalProduct(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectKeyProduct(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectListProductCo2(@Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Double selectCo2ByComanyName(@Param("modelCompanyName") String modelCompanyName, @Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Double selectInputByComanyName(@Param("modelCompanyName") String modelCompanyName, @Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public List<Map<String, Object>> selectListProductCo2ByComanyName(@Param("modelCompanyName") String modelCompanyName, @Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

public Map<String, Object> selectProductCo2FactorByComanyName(@Param("modelCompanyName") String modelCompanyName, @Param("beginDate") Date beginDate, @Param("endDate") Date endDate);

}

package com.jiexun.api.service.impl;

import com.jiexun.api.mapper.BigDataMapper;

import com.jiexun.api.service.ILargeScreenService;

import com.jiexun.common.utils.DateUtils;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.Date;

import java.util.List;

import java.util.Map;

@Service

public class LargeScreenServiceImpl implements ILargeScreenService {

@Autowired

private BigDataMapper mapper;

@Override

public Double selectTotalOil(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectTotalOil(beginDate, endDate);

}

@Override

public Double selectTotalCO2(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectTotalCo2(beginDate, endDate);

}

@Override

public Double selectUsedTotalOil(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectUsedTotalOil(beginDate, endDate);

}

@Override

public List<Map<String, Double>> selectZoneCo2(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectZoneCo2(beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectCo2Factor(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectCo2Factor(beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectListCo2(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectListCo2(beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectListCo2Factor(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectListCo2Factor(beginDate, endDate);

}

@Override

public Map<String, Object> selectTotalProduct(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectTotalProduct(beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectKeyProduct(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectKeyProduct(beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectListProductCo2(int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectListProductCo2(beginDate, endDate);

}

@Override

public Double selectCo2ByComanyName(String modelCompanyName, int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectCo2ByComanyName(modelCompanyName, beginDate, endDate);

}

@Override

public Double selectInputByComanyName(String modelCompanyName, int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectInputByComanyName(modelCompanyName, beginDate, endDate);

}

@Override

public List<Map<String, Object>> selectListProductCo2ByComanyName(String modelCompanyName, int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectListProductCo2ByComanyName(modelCompanyName, beginDate, endDate);

}

@Override

public Map<String, Object> selectProductCo2FactorByComanyName(String modelCompanyName, int year) {

Date beginDate = DateUtils.getYearFirst(year);

Date endDate = DateUtils.getYearLast(year);

return mapper.selectProductCo2FactorByComanyName(modelCompanyName, beginDate, endDate);

}

}

package com.jiexun.api.service;

import java.util.List;

import java.util.Map;

public interface ILargeScreenService {

public Double selectTotalOil(int year);

public Double selectTotalCO2(int year);

public Double selectUsedTotalOil(int year);

public List<Map<String, Double>> selectZoneCo2(int year);

public List<Map<String, Object>> selectCo2Factor(int year);

public List<Map<String, Object>> selectListCo2(int year);

public List<Map<String, Object>> selectListCo2Factor(int year);

public Map<String, Object> selectTotalProduct(int year);

public List<Map<String, Object>> selectKeyProduct(int year);

public List<Map<String, Object>> selectListProductCo2(int year);

public Double selectCo2ByComanyName(String modelCompanyName, int year);

public Double selectInputByComanyName(String modelCompanyName, int year);

public List<Map<String, Object>> selectListProductCo2ByComanyName(String modelCompanyName, int year);

public Map<String, Object> selectProductCo2FactorByComanyName(String modelCompanyName, int year);

}

package com.jiexun.api.task;

import com.alibaba.fastjson2.JSONArray;

import com.alibaba.fastjson2.JSONObject;

import com.jiexun.api.constants.CmsConstants;

import com.jiexun.cms.domain.CmsContent;

import com.jiexun.cms.service.ICmsContentService;

import com.jiexun.common.core.page.TableDataInfo;

import com.jiexun.common.service.ElasticsearchService;

import com.jiexun.common.utils.DateUtils;

import com.jiexun.common.utils.StringUtils;

import com.jiexun.common.utils.uuid.UUID;

import org.elasticsearch.action.bulk.BulkRequest;

import org.elasticsearch.action.bulk.BulkResponse;

import org.elasticsearch.action.update.UpdateRequest;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.common.xcontent.XContentType;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Component;

import javax.annotation.Resource;

import java.io.IOException;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

/\*\*

\* ContentTask . <br>

\* @author YanZhiCheng

\* date: 2022/8/1下午5:31

\*/

@Component("ContentTask")

public class ContentTask {

@Resource

private ElasticsearchService elasticsearchService;

@Autowired

private ICmsContentService cmsContentService;

public static int isPublic = 1;

public static void main(String[] args) {

// grabAreaData("政策法规", 2, 4L, 4L);

}

/\*\*

\* 定时任务，取消过期的置顶的内容

\*/

public void cancelExpiredTopping(){

CmsContent cmsContent = new CmsContent();

cmsContent.setReleaseStatus(CmsConstants.RELEASE\_STATE\_TRUE);

cmsContent.setAuditStatus(CmsConstants.AUDIT\_STATE\_PASS);

cmsContent.setDelState(CmsConstants.DEL\_STATE\_FALSE);

cmsContent.setToTopEndTime(DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD, DateUtils.getNowDate()));

TableDataInfo tableDataInfo = cmsContentService.selectCmsContentPageList(cmsContent, 1, 1000);

long total = tableDataInfo.getTotal();

if (total > 0) {

List<CmsContent> rows = (List<CmsContent>) tableDataInfo.getRows();

if (null != rows && !rows.isEmpty()) {

BulkRequest bulkRequest = new BulkRequest();

for (CmsContent content : rows) {

bulkRequest.add(

new UpdateRequest(CmsConstants.CONTENT\_INDEX, content.getUuid()).id(content.getUuid())

.doc(XContentType.JSON,

"isTop",CmsConstants.NOT\_TOP));

}

try {

BulkResponse bulk = elasticsearchService.bulk(bulkRequest, RequestOptions.DEFAULT);

System.out.println(bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

}

}

/\*\*

\* 获取内容定时任务

\* @param k 关键字

\* @param p 共几页

\* @param appId 应用

\* @param channelId 栏目

\* @return

\*/

public Map<String, Object> getContentData(,String url,String k, Integer p, Long appId, Long channelId) {

Map<String, Object> result = new HashMap<>();

result.put("msg", "成功");

result.put("code", 200);

url += "?callback=jsonpCallback&pageSize=10&siteId=11200&sort=1&isInclude=1&keyword=";

url += k;

if (null == p) {

return result;

}

for (int i = 1; i < p; i++) {

getSearchRes(url + "&pageNo=" + i, channelId, appId);

}

return result;

}

private void getSearchRes(String url, Long channelId, Long appId) {

Document connect = connect(url);

String s = connect.select("body").text();

String replace = s.replace("jsonpCallback(", "");

String substring = replace.substring(0, replace.length() - 2);

JSONObject jsonObject = JSONObject.parseObject(substring);

if (null == jsonObject || (jsonObject.get("code") != null && jsonObject.get("code").equals("200"))) {

return;

}

Map<String, JSONArray> content = (Map<String, JSONArray>) jsonObject.get("content");

if (null == content) {

return;

}

JSONArray result = content.get("result");

if (null == result || result.isEmpty()) {

return;

}

for (Object row : result) {

getContent(channelId, appId, (JSONObject) row);

}

}

private void getContent(Long channelId, Long appId, JSONObject row) {

List<String> urls = (List<String>) row.get("originUrl");

if (null == urls || urls.isEmpty()) {

return;

}

String s = urls.get(0);

if (StringUtils.isEmpty(s)) {

return;

}

Document connect = connect(s);

Element element = connect.select(".channel-page").get(0);

String title = element.select(".titles").text();

if (StringUtils.isEmpty(title)) {

return;

}

String releaseTime = element.select(".times").text().replace("发布时间：", "") + " 00:00:00";

String author = element.select(".author").text().replace("来源：", "");

String content = element.select("div.article-content").text();

String dateToStr = DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD\_HH\_MM\_SS, DateUtils.getNowDate());

CmsContent cmsContent = new CmsContent();

String uuid = UUID.getUuid();

cmsContent.setUuid(uuid);

cmsContent.setChannelId(channelId);

cmsContent.setAppId(appId);

cmsContent.setAuthor(author);

cmsContent.setReleaseTime(releaseTime);

cmsContent.setContentTempId(7);

cmsContent.setSource(author);

cmsContent.setTitle(title);

cmsContent.setSubTitle(title);

cmsContent.setTitleImg("");

cmsContent.setContent(content);

cmsContent.setIsTop(0);

cmsContent.setToTopEndTime(dateToStr);

cmsContent.setDelState(0);

cmsContent.setAuditStatus(1);

cmsContent.setReleaseStatus(1);

cmsContent.setRotationChart(1);

cmsContent.setCharge(1);

cmsContent.setIsHot(0);

cmsContent.setReadCount(0L);

cmsContent.setCreateUser(1L);

cmsContent.setCreateBy("system");

cmsContent.setUpdateBy("system");

cmsContent.setCreateTime(dateToStr);

cmsContent.setUpdateTime(dateToStr);

cmsContent.setDeptId(1L);

try {

elasticsearchService.addDocument(cmsContent, CmsConstants.CONTENT\_INDEX, uuid);

} catch (Exception e) {

e.printStackTrace();

}

}

/\*\*

\* 建立连接

\*/

private static Document connect(String url) {

if (url == null || url.isEmpty()) {

throw new IllegalArgumentException("无效的url");

}

try {

return Jsoup.connect(url).timeout(200 \* 2000).get();

} catch (IOException e) {

System.out.println(url + "地址不存在");

return null;

}

}

}

package com.jiexun.api.util;

import net.sourceforge.pinyin4j.PinyinHelper;

import net.sourceforge.pinyin4j.format.HanyuPinyinCaseType;

import net.sourceforge.pinyin4j.format.HanyuPinyinOutputFormat;

import net.sourceforge.pinyin4j.format.HanyuPinyinToneType;

import net.sourceforge.pinyin4j.format.exception.BadHanyuPinyinOutputFormatCombination;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

/\*\*

\*

\* @ClassName: ChineseAndEnglishUtil

\* @Description: 判断是中文还是英文

\* @author: Hepengyu

\* @date: May 15, 2020 8:00:08 PM

\* @Copyright:

\*/

public class ChineseAndEnglishUtil {

/\*\*

\*

\* @Title: isChinese

\* @Description: 是否是中文

\* @param c

\* @return: boolean

\* @throws

\*/

public static boolean isChinese(char c) {

Character.UnicodeBlock ub = Character.UnicodeBlock.of(c);

if (ub == Character.UnicodeBlock.CJK\_UNIFIED\_IDEOGRAPHS

|| ub == Character.UnicodeBlock.CJK\_COMPATIBILITY\_IDEOGRAPHS

|| ub == Character.UnicodeBlock.CJK\_UNIFIED\_IDEOGRAPHS\_EXTENSION\_A

|| ub == Character.UnicodeBlock.GENERAL\_PUNCTUATION

|| ub == Character.UnicodeBlock.CJK\_SYMBOLS\_AND\_PUNCTUATION

|| ub == Character.UnicodeBlock.HALFWIDTH\_AND\_FULLWIDTH\_FORMS) {

return true;

}

return false;

}

/\*\*

\*

\* @Title: isEnglish

\* @Description: 是否是英文

\* @param charaString

\* @return: boolean

\* @throws

\*/

public static boolean isEnglish(String charaString) {

return charaString.matches("^[a-zA-Z][0-9]\*");

}

/\*\*

\*

\* @Title: isChinese

\* @Description: 是否是中文

\* @param str

\* @return

\* @return: boolean

\* @throws

\*/

public static boolean isChinese(String str) {

String regEx = "[\\u4e00-\\u9fa5]+";

Pattern p = Pattern.compile(regEx);

Matcher m = p.matcher(str);

if (m.find()) {

return true;

} else {

return false;

}

}

public static String toHanyuPinyinFirst(String str) throws BadHanyuPinyinOutputFormatCombination {

HanyuPinyinOutputFormat format = new HanyuPinyinOutputFormat();

//拼音小写

format.setCaseType(HanyuPinyinCaseType.UPPERCASE);

//不带声调

format.setToneType(HanyuPinyinToneType.WITHOUT\_TONE);

String code = "";

for(int i=0; i<str.length(); i++){

String subString = str.substring(i, i+1);

if(ChineseAndEnglishUtil.isEnglish(subString)){

code += subString;

}else{

code += PinyinHelper.toHanYuPinyinString(subString, format, "", true).substring(0, 1);

}

}

return code;

}

}

package com.jiexun.api.vo;

import com.alibaba.fastjson2.JSON;

import com.fasterxml.jackson.annotation.JsonFormat;

import com.jiexun.cms.domain.CmsContent;

import java.io.Serializable;

/\*\*

\* ContentVO . <br>

\*

\* @author YanZhiCheng

\* date: 2022/9/5上午9:27

\*/

public class ContentVO implements Serializable {

private String uuid;

private Long channelId;

private String title;

private String subTitle;

private String titleImg;

private Integer contentTempId;

private String author;

private String content;

private Integer isHot;

private Integer isTop;

private Long readCount;

@JsonFormat(pattern="yyyy-MM-dd", timezone="GMT+8")

private String releaseTime;

public static ContentVO convert(CmsContent c) {

ContentVO vo = new ContentVO();

vo.setUuid(c.getUuid());

vo.setAuthor(c.getAuthor());

vo.setContent(c.getContent());

vo.setContentTempId(c.getContentTempId());

vo.setChannelId(c.getChannelId());

vo.setReadCount(c.getReadCount());

vo.setIsTop(c.getIsTop());

vo.setTitle(c.getTitle());

vo.setReleaseTime(c.getReleaseTime());

vo.setSubTitle(c.getSubTitle());

vo.setTitleImg(c.getTitleImg());

return vo;

}

public String getUuid() {

return uuid;

}

public void setUuid(String uuid) {

this.uuid = uuid;

}

public Long getChannelId() {

return channelId;

}

public void setChannelId(Long channelId) {

this.channelId = channelId;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getSubTitle() {

return subTitle;

}

public void setSubTitle(String subTitle) {

this.subTitle = subTitle;

}

public String getTitleImg() {

return titleImg;

}

public void setTitleImg(String titleImg) {

this.titleImg = titleImg;

}

public Integer getContentTempId() {

return contentTempId;

}

public void setContentTempId(Integer contentTempId) {

this.contentTempId = contentTempId;

}

public String getAuthor() {

return author;

}

public void setAuthor(String author) {

this.author = author;

}

public String getContent() {

return content;

}

public void setContent(String content) {

this.content = content;

}

public Integer getIsHot() {

return isHot;

}

public void setIsHot(Integer isHot) {

this.isHot = isHot;

}

public Integer getIsTop() {

return isTop;

}

public void setIsTop(Integer isTop) {

this.isTop = isTop;

}

public Long getReadCount() {

return readCount;

}

public void setReadCount(Long readCount) {

this.readCount = readCount;

}

public String getReleaseTime() {

return releaseTime;

}

public void setReleaseTime(String releaseTime) {

this.releaseTime = releaseTime;

}

@Override

public String toString() {

return JSON.toJSONString(this);

}

}

package com.jiexun.api.vo;

import java.util.HashMap;

import java.util.Map;

public class DeviceOsblVO {

private String deviceName;

private String deviceCode;

private Double deviceCo2 = 0.0;

private Double deviceProcess = 0.0;

private Double deviceCo2Factor = 0.0;

private Double deviceCo2Percent = 0.0;

private Map<String, Double> osbls = new HashMap<>();

public Double getDeviceCo2() {

return deviceCo2;

}

public void setDeviceCo2(Double deviceCo2) {

this.deviceCo2 = deviceCo2;

}

public Map<String, Double> getOsbls() {

return osbls;

}

public void setOsbls(Map<String, Double> osbls) {

this.osbls = osbls;

}

public DeviceOsblVO() {

}

public String getDeviceName() {

return deviceName;

}

public void setDeviceName(String deviceName) {

this.deviceName = deviceName;

}

public String getDeviceCode() {

return deviceCode;

}

public void setDeviceCode(String deviceCode) {

this.deviceCode = deviceCode;

}

public Double getDeviceProcess() {

return deviceProcess;

}

public void setDeviceProcess(Double deviceProcess) {

this.deviceProcess = deviceProcess;

}

public Double getDeviceCo2Factor() {

return deviceCo2Factor;

}

public void setDeviceCo2Factor(Double deviceCo2Factor) {

this.deviceCo2Factor = deviceCo2Factor;

}

public Double getDeviceCo2Percent() {

return deviceCo2Percent;

}

public void setDeviceCo2Percent(Double deviceCo2Percent) {

this.deviceCo2Percent = deviceCo2Percent;

}

}

package com.jiexun.api.vo;

import java.io.Serializable;

/\*\*

\* 获取轮播图的实体类 . <br>

\* @author YanZhiCheng

\* date: 2022/8/8下午4:21

\*/

public class RotationChartContentVO implements Serializable {

private Long[] channelIds;

private Integer appId;

private Integer pageSize;

private Integer pageNum;

private Integer rotationChart;

private Integer releaseStatus;

private Integer auditStatus;

private Integer delState;

public Long[] getChannelIds() {

return channelIds;

}

public void setChannelIds(Long[] channelIds) {

this.channelIds = channelIds;

}

public Integer getAppId() {

return appId;

}

public void setAppId(Integer appId) {

this.appId = appId;

}

public Integer getPageSize() {

return pageSize == null ? 10 : pageSize;

}

public void setPageSize(Integer pageSize) {

this.pageSize = pageSize;

}

public Integer getPageNum() {

return pageNum == null ? 1 : pageNum;

}

public void setPageNum(Integer pageNum) {

this.pageNum = pageNum;

}

public Integer getRotationChart() {

return rotationChart;

}

public void setRotationChart(Integer rotationChart) {

this.rotationChart = rotationChart;

}

public Integer getReleaseStatus() {

return releaseStatus;

}

public void setReleaseStatus(Integer releaseStatus) {

this.releaseStatus = releaseStatus;

}

public Integer getAuditStatus() {

return auditStatus;

}

public void setAuditStatus(Integer auditStatus) {

this.auditStatus = auditStatus;

}

public Integer getDelState() {

return delState;

}

public void setDelState(Integer delState) {

this.delState = delState;

}

}

package com.jiexun.api.vo;

import com.jiexun.lowcarbon.domain.TechnicalClassification;

import com.jiexun.system.domain.SysFileInfo;

import java.io.Serializable;

import java.util.List;

/\*\*

\* TODO . <br>

\*

\* @author YanZhiCheng

\* date: 2022/9/7下午2:38

\*/

public class TechnicalClassificationVO implements Serializable {

/\*\* uuid \*/

private String uuid;

/\*\* 技术类别 \*/

private String name;

private List<SysFileInfo> files;

public static TechnicalClassificationVO convert(TechnicalClassification obj){

TechnicalClassificationVO vo = new TechnicalClassificationVO();

vo.setUuid(obj.getUuid());

vo.setName(obj.getName());

vo.setFiles(obj.getFileInfoList());

return vo;

}

public String getUuid() {

return uuid;

}

public void setUuid(String uuid) {

this.uuid = uuid;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public List<SysFileInfo> getFiles() {

return files;

}

public void setFiles(List<SysFileInfo> files) {

this.files = files;

}

}

package com.jiexun.cms.controller;

import java.io.IOException;

import java.util.\*;

import java.util.stream.Collectors;

import javax.servlet.http.HttpServletResponse;

import com.alibaba.fastjson2.JSONObject;

import com.jiexun.api.constants.CmsConstants;

import com.jiexun.common.core.page.PageDomain;

import com.jiexun.common.core.page.TableSupport;

import com.jiexun.common.service.ElasticsearchService;

import com.jiexun.common.utils.DateUtils;

import com.jiexun.common.utils.StringUtils;

import com.jiexun.common.utils.uuid.UUID;

import com.jiexun.system.domain.SysFileInfo;

import com.jiexun.system.service.ISysFileInfoService;

import org.elasticsearch.action.bulk.BulkRequest;

import org.elasticsearch.action.bulk.BulkResponse;

import org.elasticsearch.action.get.GetResponse;

import org.elasticsearch.action.update.UpdateRequest;

import org.elasticsearch.action.update.UpdateResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.common.xcontent.XContentType;

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.cms.domain.CmsContent;

import com.jiexun.cms.service.ICmsContentService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

/\*\*

\* 内容管理Controller

\* @author yzc

\* @date 2022-07-26

\*/

@RestController

@RequestMapping("/cms/cmsContent")

public class CmsContentController extends BaseController {

@Autowired

private ICmsContentService cmsContentService;

@Autowired

private ElasticsearchService<CmsContent> elasticsearchService;

@Autowired

private ISysFileInfoService sysFileInfoService;

/\*\*

\* 查询内容管理列表

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:list')")

@GetMapping("/list")

public TableDataInfo list(CmsContent cmsContent) {

PageDomain pageDomain = TableSupport.buildPageRequest();

Integer pageNum = pageDomain.getPageNum();

Integer pageSize = pageDomain.getPageSize();

return cmsContentService.selectCmsContentPageList(cmsContent, pageNum, pageSize);

}

/\*\*

\* 导出内容管理列表

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:export')")

@Log(title = "内容管理", businessType = BusinessType.EXPORT)

@PostMapping("/export")

@SuppressWarnings("unchecked")

public void export(HttpServletResponse response, CmsContent cmsContent) {

List<CmsContent> list = (List<CmsContent>) cmsContentService

.selectCmsContentPageList(cmsContent, 1, 999999).getRows();

ExcelUtil<CmsContent> util = new ExcelUtil<CmsContent>(CmsContent.class);

util.exportExcel(response, list, "内容管理数据");

}

/\*\*

\* 获取内容管理详细信息

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:query')")

@GetMapping(value = "/{uuid}")

public AjaxResult getInfo(@PathVariable("uuid") String uuid) {

try {

GetResponse document = elasticsearchService.getDocument(CmsConstants.CONTENT\_INDEX, uuid);

CmsContent content = JSONObject.parseObject(JSONObject.toJSONString(document.getSourceAsMap()),

CmsContent.class);

content.setUuid(uuid);

String[] fileIds = content.getOtherFiles();

if (null != fileIds && fileIds.length > 0 ) {

List<SysFileInfo> fileList = sysFileInfoService.selectFileListByIds(fileIds);

if (null != fileList) {

content.setOtherFileList(fileList);

}

}

return AjaxResult.success(content);

} catch (IOException e) {

e.printStackTrace();

}

return AjaxResult.error();

}

/\*\*

\* 新增内容管理

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:add')")

@Log(title = "内容管理", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody CmsContent cmsContent) {

String uuid = UUID.getUuid();

String dateToStr = DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD\_HH\_MM\_SS, DateUtils.getNowDate());

String username = getUsername();

if (null == cmsContent.getIsTop()) {

cmsContent.setIsTop(0);

}

if (null == cmsContent.getIsHot()) {

cmsContent.setIsHot(0);

}

if (null == cmsContent.getAuditStatus()) {

cmsContent.setAuditStatus(0);

}

if (null == cmsContent.getReleaseStatus()) {

cmsContent.setReleaseStatus(0);

}

if (null == cmsContent.getPermissionType()) {

cmsContent.setPermissionType(new Integer[]{1});

}

if (null == cmsContent.getCharge()) {

cmsContent.setCharge(1);

}

if (null == cmsContent.getRotationChart()) {

cmsContent.setRotationChart(0);

}

cmsContent.setDelState(0);

cmsContent.setReadCount(0L);

cmsContent.setCreateUser(getUserId());

cmsContent.setCreateBy(username);

cmsContent.setCreateTime(dateToStr);

cmsContent.setUpdateBy(username);

cmsContent.setUpdateTime(dateToStr);

cmsContent.setDeptId(getDeptId());

List<SysFileInfo> otherFileList = cmsContent.getOtherFileList();

if (null != otherFileList && !otherFileList.isEmpty()) {

List<String> collect = otherFileList.stream().map(e -> e.getUuid()).collect(Collectors.toList());

cmsContent.setOtherFiles((String[]) collect.toArray());

}

cmsContent.setOtherFileList(null);

try {

boolean b = elasticsearchService.addDocument(cmsContent, CmsConstants.CONTENT\_INDEX, uuid);

return toAjax(b);

} catch (IOException e) {

e.printStackTrace();

}

return AjaxResult.error();

}

/\*\*

\* 修改内容管理

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:edit')")

@Log(title = "内容管理", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody CmsContent cmsContent) {

try {

String dateToStr = DateUtils.parseDateToStr(DateUtils.YYYY\_MM\_DD\_HH\_MM\_SS, DateUtils.getNowDate());

String username = getUsername();

cmsContent.setUpdateBy(username);

cmsContent.setUpdateTime(dateToStr);

if (cmsContent.getAuditStatus()!=null && cmsContent.getAuditStatus().equals(CmsConstants.AUDIT\_STATE\_PASS)) {

cmsContent.setAuditStatus(CmsConstants.AUDIT\_STATE\_NONE);

}

Integer memberGroupId = cmsContent.getMemberGroupId();

if (null == memberGroupId) {

cmsContent.setMemberGroupId(0);

}

UpdateResponse updateResponse = elasticsearchService

.updateDocument(CmsConstants.CONTENT\_INDEX, cmsContent.getUuid(), cmsContent);

boolean res = updateResponse.status().getStatus() == 200;

return toAjax(res);

} catch (IOException e) {

e.printStackTrace();

}

return AjaxResult.error();

}

/\*\*

\* 逻辑删除内容管理

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:remove')")

@Log(title = "内容管理", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult remove(@PathVariable String[] uuids) {

BulkRequest request = new BulkRequest();

if (null != uuids && uuids.length > 0) {

for (String uuid : uuids) {

request.add(new UpdateRequest(

CmsConstants.CONTENT\_INDEX, uuid).id(uuid)

.doc(XContentType.JSON, "delState", CmsConstants.DEL\_STATE\_TRUE));

}

try {

BulkResponse bulk = elasticsearchService.bulk(request, RequestOptions.DEFAULT);

return toAjax(!bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

return AjaxResult.error();

}

/\*\*

\* 审核通过

\* @param uuidStr

\* @return

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:audit')")

@Log(title = "内容管理", businessType = BusinessType.APPROVED)

@GetMapping("/approved")

public AjaxResult approvedCmsChannelByIds(String uuidStr) {

BulkRequest bulkRequest = new BulkRequest();

String[] uuids = uuidStr.split(",");

if (null != uuids && uuids.length > 0) {

for (String uuid : uuids) {

if (StringUtils.isEmpty(uuid)) {

continue;

}

bulkRequest.add(

new UpdateRequest(CmsConstants.CONTENT\_INDEX, uuid).id(uuid)

.doc(XContentType.JSON,

"auditStatus",CmsConstants.AUDIT\_STATE\_PASS));

}

try {

BulkResponse bulk = elasticsearchService.bulk(bulkRequest, RequestOptions.DEFAULT);

return toAjax(!bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

return AjaxResult.error();

}

/\*\*

\* 审核退回

\* @param uuidStr

\* @return

\*/

@PreAuthorize("@ss.hasPermi('cms:cmsContent:audit')")

@Log(title = "内容管理", businessType = BusinessType.REVIEW\_RETURN)

@GetMapping("/reviewReturn")

public AjaxResult reviewReturnCmsChannelByIds(String uuidStr) {

BulkRequest bulkRequest = new BulkRequest();

String[] uuids = uuidStr.split(",");

if (null != uuids && uuids.length > 0) {

for (String uuid : uuids) {

if (StringUtils.isEmpty(uuid)) {

continue;

}

bulkRequest.add(

new UpdateRequest(CmsConstants.CONTENT\_INDEX, uuid).id(uuid)

.doc(XContentType.JSON,

"auditStatus",CmsConstants.AUDIT\_STATE\_RETURN));

}

try {

BulkResponse bulk = elasticsearchService.bulk(bulkRequest, RequestOptions.DEFAULT);

return toAjax(!bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

return AjaxResult.error();

}

}

package com.jiexun.cms.controller;

import com.jiexun.api.constants.CmsConstants;

import com.jiexun.cms.domain.CmsContent;

import com.jiexun.cms.service.ICmsContentService;

import com.jiexun.common.annotation.Log;

import com.jiexun.common.core.controller.BaseController;

import com.jiexun.common.core.domain.AjaxResult;

import com.jiexun.common.core.page.PageDomain;

import com.jiexun.common.core.page.TableDataInfo;

import com.jiexun.common.core.page.TableSupport;

import com.jiexun.common.enums.BusinessType;

import com.jiexun.common.service.ElasticsearchService;

import org.elasticsearch.action.bulk.BulkRequest;

import org.elasticsearch.action.bulk.BulkResponse;

import org.elasticsearch.action.delete.DeleteRequest;

import org.elasticsearch.action.update.UpdateRequest;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.common.xcontent.XContentType;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.security.access.prepost.PreAuthorize;

import org.springframework.web.bind.annotation.\*;

import java.io.IOException;

/\*\*

\* 内容回收站管理 . <br>

\* @author YanZhiCheng

\* date: 2022/7/27上午9:39

\*/

@RestController

@RequestMapping("/cms/dustbin")

public class CmsDustbinContentController extends BaseController {

@Autowired

private ICmsContentService cmsContentService;

@Autowired

private ElasticsearchService<CmsContent> elasticsearchService;

/\*\*

\* 查询回收站内容管理列表

\* @param cmsContent

\* @return

\*/

@PreAuthorize("@ss.hasPermi('cms:dustbinContent:list')")

@GetMapping("/list")

public TableDataInfo dustbinList(CmsContent cmsContent) {

cmsContent.setDelState(CmsConstants.DEL\_STATE\_TRUE);

PageDomain pageDomain = TableSupport.buildPageRequest();

Integer pageNum = pageDomain.getPageNum();

Integer pageSize = pageDomain.getPageSize();

TableDataInfo result = cmsContentService.selectCmsContentPageList(cmsContent, pageNum, pageSize);

return result;

}

/\*\*

\* 回收站内容数据放回原处 并修改审核状态为未审核

\* @param uuids

\* @return

\*/

@PreAuthorize("@ss.hasPermi('cms:dustbinContent:putBack')")

@Log(title = "内容回收站管理", businessType = BusinessType.PUT\_BACK)

@GetMapping("/putBak/{uuids}")

public AjaxResult putBak(@PathVariable String[] uuids) {

BulkRequest request = new BulkRequest();

if (null != uuids && uuids.length > 0) {

for (String uuid : uuids) {

request.add(

new UpdateRequest(CmsConstants.CONTENT\_INDEX, uuid).id(uuid)

.doc(XContentType.JSON,

"delState", CmsConstants.DEL\_STATE\_FALSE,

"auditStatus",CmsConstants.AUDIT\_STATE\_NONE));

}

try {

BulkResponse bulk = elasticsearchService.bulk(request, RequestOptions.DEFAULT);

return toAjax(!bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

return AjaxResult.error();

}

/\*\*

\* 删除回收站内容数据

\*/

@PreAuthorize("@ss.hasPermi('cms:dustbinContent:remove')")

@Log(title = "内容回收站管理", businessType = BusinessType.DELETE)

@DeleteMapping("/{uuids}")

public AjaxResult delete(@PathVariable String[] uuids) {

BulkRequest request = new BulkRequest();

if (null != uuids && uuids.length > 0) {

for (String uuid : uuids) {

request.add(new DeleteRequest(CmsConstants.CONTENT\_INDEX, uuid).id(uuid));

}

try {

BulkResponse bulk = elasticsearchService.bulk(request, RequestOptions.DEFAULT);

return toAjax(!bulk.hasFailures());

} catch (IOException e) {

e.printStackTrace();

}

}

return AjaxResult.error();

}

}

package com.jiexun.cms.controller;

import java.util.List;

import java.util.Set;

import javax.servlet.http.HttpServletResponse;

import com.jiexun.cms.domain.CmsMessageLike;

import com.jiexun.common.utils.SecurityUtils;

import com.jiexun.framework.web.service.SysPermissionService;

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.cms.domain.CmsMessage;

import com.jiexun.cms.service.ICmsMessageService;

import com.jiexun.common.utils.poi.ExcelUtil;

import com.jiexun.common.core.page.TableDataInfo;

/\*\*

\* 留言管理Controller

\* @author ning

\* @date 2022-01-15

\*/

@RestController

@RequestMapping("/cms/message")

public class CmsMessageController extends BaseController {

@Autowired

private ICmsMessageService cmsMessageService;

@Autowired

private SysPermissionService permissionService;

/\*\*

\* 首页查询留言列表

\*/

@GetMapping("/cms/list")

public TableDataInfo cmsList(CmsMessage cmsMessage) {

startPage();

List<CmsMessage> list = cmsMessageService.selectMessageList(cmsMessage);

return getDataTable(list);

}

/\*\*

\* 首页新增留言

\*/

@PostMapping("/cms/addMessage")

public AjaxResult addMessage(@RequestBody CmsMessage cmsMessage) {

Long parentId = cmsMessage.getParentId();

if (parentId != null) {

CmsMessage message = cmsMessageService.selectCmsMessageById(parentId);

if (message.getMainId() != null) {

cmsMessage.setMainId(message.getMainId());

} else {

cmsMessage.setMainId(parentId);

}

}

return toAjax(cmsMessageService.insertCmsMessage(cmsMessage));

}

/\*\*

\* 首页新增点赞

\*/

@PostMapping("/cms/addCmsMessageLike")

public AjaxResult addCmsMessageLike(@RequestBody CmsMessageLike cmsMessageLike) {

return toAjax(cmsMessageService.addCmsMessageLike(cmsMessageLike));

}

/\*\*

\* 首页删除点赞

\*/

@Log(title = "删除留言点赞", businessType = BusinessType.DELETE)

@PostMapping("/cms/delCmsMessageLike")

public AjaxResult delCmsMessageLike(@RequestBody CmsMessageLike cmsMessageLike) {

return toAjax(cmsMessageService.delCmsMessageLike(cmsMessageLike));

}

/\*\*

\* 查询留言管理列表

\*/

@PreAuthorize("@ss.hasPermi('cms:message:list')")

@GetMapping("/list")

public TableDataInfo list(CmsMessage cmsMessage) {

startPage();

// 角色集合

Set<String> roles = permissionService.getRolePermission(getLoginUser().getUser());

if (!SecurityUtils.isAdmin(getUserId()) && !roles.contains("admin") && !roles.contains("cms")) {

cmsMessage.setCreateBy(getUsername());

}

cmsMessage.setDelFlag("0");

List<CmsMessage> list = cmsMessageService.selectCmsMessageList(cmsMessage);

return getDataTable(list);

}

/\*\*

\* 导出留言管理列表

\*/

@PreAuthorize("@ss.hasPermi('cms:message:export')")

@Log(title = "留言管理", businessType = BusinessType.EXPORT)

@PostMapping("/export")

public void export(HttpServletResponse response, CmsMessage cmsMessage) {

List<CmsMessage> list = cmsMessageService.selectCmsMessageList(cmsMessage);

ExcelUtil<CmsMessage> util = new ExcelUtil<CmsMessage>(CmsMessage.class);

util.exportExcel(response, list, "留言管理数据");

}

/\*\*

\* 获取留言管理详细信息

\*/

@PreAuthorize("@ss.hasPermi('cms:message:query')")

@GetMapping(value = "/{id}")

public AjaxResult getInfo(@PathVariable("id") Long id) {

return AjaxResult.success(cmsMessageService.selectCmsMessageById(id));

}

/\*\*

\* 新增留言管理

\*/

@PreAuthorize("@ss.hasPermi('cms:message:add')")

@Log(title = "留言管理", businessType = BusinessType.INSERT)

@PostMapping

public AjaxResult add(@RequestBody CmsMessage cmsMessage) {

return toAjax(cmsMessageService.insertCmsMessage(cmsMessage));

}

/\*\*

\* 修改留言管理

\*/

@PreAuthorize("@ss.hasPermi('cms:message:edit')")

@Log(title = "留言管理", businessType = BusinessType.UPDATE)

@PutMapping

public AjaxResult edit(@RequestBody CmsMessage cmsMessage) {

return toAjax(cmsMessageService.updateCmsMessage(cmsMessage));

}

/\*\*

\* 删除留言管理

\*/

@PreAuthorize("@ss.hasPermi('cms:message:remove')")

@Log(title = "留言管理", businessType = BusinessType.DELETE)

@DeleteMapping("/{ids}")

public AjaxResult remove(@PathVariable Long[] ids) {

return toAjax(cmsMessageService.deleteCmsMessageByIds(ids));

}

}