石化网站内容管理系统源代码

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'

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/FileUpload"

// 图片上传组件

import ImageUpload from "@/components/ImageUpload"

// 图片预览组件

import ImagePreview from "@/components/ImagePreview"

// 字典标签组件

import DictTag from '@/components/DictTag'

// 头部标签组件

import VueMeta from 'vue-meta'

// 字典数据组件

import DictData from '@/components/DictData'

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">石化网站内容管理系统</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.route|| this.redirect || "/" }).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>

<div class="app-box">

<!-- <h3><img src="@/assets/home/04dtbzyzcfg.jpg" alt="" style="background: #438ae2;"/><b>{{ name }}</b></h3>-->

<div class="breadcrumb">

<img src="@/assets/home/location\_03.png" style="height: 18px;margin-right: 5px;"/>

<el-breadcrumb separator="/">

<el-breadcrumb-item :to="{ path: '/carbonStandard/main/index' }">首页</el-breadcrumb-item>

<el-breadcrumb-item>{{ name }}</el-breadcrumb-item>

</el-breadcrumb>

</div>

<div style="width: 80%;">

<div style="width: 29%;float: left;">

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

<div class="cardTitle">{{ name }}</div>

<hr style="width: 108%;margin: 10px 0 10px -4%;"/>

<el-row class="li-pointer" style="margin-top: 10px;">

<div v-for="(item, index) in channelList" :class="[item.highLight ? 'laterBox' : 'frontBox']" @click="clickChannel(item.id)">

<span style="margin-left: 40px;">{{ item.name }}</span>

</div>

</el-row>

</el-card>

</div>

<div style="width: 70%;float: left;margin-left: 1%;">

<div class="checkChannelName">

<img src="@/assets/home/list\_01.png" style="height: 30px; float: left; margin-top: 11px;"/>

<div class="txt">{{ channelName }}</div>

</div>

<hr/>

<ul class="box-content li-pointer" style="width: 100%;">

<div v-if="tableData.length==0" style="text-align: center">暂无数据</div>

<li v-else v-for="(item,index) in tableData" :key="index" class="flex\_between font16" @click="gotoContent(item)">

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

<div>{{ parseTime(item.releaseTime, '{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"

/>

</div>

</div>

</div>

</template>

<script>

import {getList, getSubNavList, getListTemplate} from "@/api/carbonStandard/index";

export default {

metaInfo() {

return {

title: this.$route.query.name

}

},

data() {

return {

// 遮罩层

loading: true,

allHighLight: true,

// 列表数据

tableData: [],

channelList: [],

// 总条数

total: 0,

pageSize: ['10'],

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 10,

companyName: null,

companyAbb: null,

companyParent: null,

companyZone: null,

companyState: null,

companyCity: null,

companyArea: null,

appId: this.$route.query.appId,

channelId: this.$route.query.channelId

},

name: this.$route.query.name,

parentChannelId: this.$route.query.channelId,

channelName: '全部',

}

},

created() {

this.init();

},

methods: {

init() {

this.allHighLight = true;

this.channelList = [{id: this.$route.query.channelId, name: '全部', highLight: true}];

this.getList()

this.getSubChannelList();

},

getList() {

this.loading = true;

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

if (response.code != 200) {

this.$message.error(response.msg);

} else {

this.tableData = response.rows;

this.total = response.total;

this.loading = false;

}

})

},

getSubChannelList() {

getSubNavList(this.parentChannelId).then(response => {

if (response.code == 200) {

let rows = response.rows;

if (!!rows) {

for (let c of rows) {

this.channelList.push(c);

}

}

} else {

this.$message.error(response.msg);

}

})

},

clickChannel(channelId) {

this.allHighLight = false;

for (let channel of this.channelList) {

if (channel.id == channelId) {

channel.highLight = true;

this.channelName = channel.name;

} else {

channel.highLight = false;

}

}

this.queryParams.channelId = channelId ? channelId : this.$route.query.channelId;

this.getList();

},

gotoContent(item) {

getListTemplate(item.uuid).then(res => {

if (res.code != 200) {

this.$message.error(res.msg);

} else {

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

path: "/carbonStandard/main/" + res.data.tempRoute,

query: {id: item.uuid}

});

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

}

})

}

},

computed: {

userdata() {

let start = (this.pageNum - 1) \* this.pageSize;

let end = this.pageNum \* this.pageSize;

return this.number.slice(start, end)

}

},

watch: {

"$route": function (to, from) {

this.queryParams.channelId = to.query.channelId

this.parentChannelId = to.query.channelId;

this.queryParams.appId = to.query.appId

this.name = to.query.name

this.init();

},

}

}

</script>

<style lang="scss">

.app-box {

width: 80%;

height: 100%;

display: flex;

align-items: center;

flex-direction: column;

margin: 0 auto;

h3 {

width: 80%;

height: 40px;

background-color: #eeeeee;

color: #438ae2;

line-height: 40px;

img {

margin-left: 2%;

}

}

.box-content {

width: 80%;

height: 63vh;

}

.box-content li {

line-height: 4.4vh;

border-bottom: 1px solid #eeeeee;

padding-left: 1vw;

//font-size: 1vh;

}

.box-content li:hover {

background: #eeeeee;

}

.box-content li div:nth-child(1) {

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.box-content li div:nth-child(2) {

width: 5vw;

text-align: center;

}

}

</style>

<style scoped>

.el-table tr {

height: 4.7vh;

}

.pagination-container {

width: 80%;

margin: 0 auto;

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

height: 6vh;

}

/deep/ .el-pagination {

transform: translate(-43%, 0%);

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

/\*margin-top:20px;\*/

}

/deep/ .el-breadcrumb {

font-size: 16px;

}

</style>

<style type="text/css">

.frontBox,.laterBox {

background: url(~@/assets/home/mainJiantou.png) center;

background-size: 101%;

height: 50px;

line-height: 50px;

margin-top: 10px;

border-radius: 5px;

border: none;

}

/\* 点击前样式 \*/

.frontBox {

color: #000000;

opacity: 0.6;

width: 95%;

}

/\* 点击后样式 \*/

.laterBox {

color: #FFFFFF;

opacity: 1;

width: 97%;

}

.app-box .box-content {

width: 80%;

height: auto;

min-height: 600px;

}

.box-card {

min-height: 700px;

}

.cardTitle {

border-bottom: 1px solid #d0d0d0;

padding-bottom: 5px;

width: 100%;

text-align: center;

font-size: 35px;

height: 70px;

background: url(~@/assets/home/blueCircle.png) no-repeat center;

background-size: 205% 115px;

color: white;

line-height: 70px;

border-radius: 9px;

}

.checkChannelName {

height: 50px;

padding-left: 10px;

border-radius: 5px;

background: #f5f5f4;

width: 100%;

}

.txt {

font-weight: bold;

letter-spacing: 2px;

min-width: 115px;

height: 50px;

line-height: 50px;

padding-left: 5px;

font-size: 22px;

color: #255a95;

text-align: left;

}

hr {

width: 98%;

border: 1px solid #a1e4ff45;

margin: 10px;

}

.breadcrumb {

margin: 20px 0 10px 0;

width: 80%;

height: 30px;

display: flex;

padding-left: 30px;

border-bottom: 1px solid #f3f3f3;

}

</style>

<template>

<div class="flex-center">

<div id="box">

<div class="box-content">

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

<div class="header-box">

<div class="title">{{contents.title}}</div>

<div class="message">

<span class="date">发布时间:{{ contents.releaseTime }}</span>

<span class="readCount">阅读次数:{{ contents.readCount }}</span>

<span class="" v-show="!!contents.subTitle">副标题:{{ contents.subTitle }}</span>

<span class="source" v-if="!!contents.source">来源:{{ contents.source }}</span>

<span class="" v-if="!!contents.author">作者:{{ contents.author }}</span>

</div>

</div>

</div>

<div class="ql-editor" v-html="contents.content"></div>

</div>

<div class="comments">

<div class="attachment" v-if="!!contents.otherFileList">

<h3>相关附件</h3>

<ul>

<li v-for="(item,index) in contents.otherFileList" :key="index">

<div>{{index+1}}、{{item.originalName}}({{item.fileSize}}kb)</div>

<div class="li-pointer" @click="download(item)"><span>下载</span></div>

</li>

</ul>

</div>

<h3>发表评论</h3>

<div class="comments-box flex\_between">

<div class="user-avatar">

<img v-if="avatar==''" :src="avatars" class="user-avatar">

<img v-else :src="avatar" class="user-avatar">

</div>

<div class="comments-content">

<div>

<el-input type="textarea" v-model="commentContents" :rows="3" placeholder="请输入内容" maxlength="200" show-word-limit resize='none'></el-input>

</div>

<div class="content-bottom">

<el-button :disabled="commentContents==''" class="sub" type="primary" @click="save">发表</el-button>

</div>

</div>

</div>

<h3>评论列表</h3>

<div class="comments-list">

<ul>

<li v-if="commentList.length==0" style="justify-content: center">暂无数据</li>

<li v-else v-for="(item,index) in commentList">

<div class="item-left">

<img :src="item.avatar?imgUrl+item.avatar:avatars" alt="">

</div>

<div class="item-right">

<div class="item-name"><h5>{{item.createByName}}</h5></div>

<div class="item-comment">{{item.content}}</div>

<div class="item-tools">

<div class="item-date li-pointer">{{item.createTime}}<span class="delComment" v-if="item.createBy==name" @click="delComments(item)">删除</span></div>

<div class="replyBtn li-pointer">

<div @click="replys(item)">

<img src="../../../../../assets/home/carbonStandard/xcp-reply.png" alt="">

<span v-show="!item.replyFlag">回复</span>

<!-- <span v-show="item.replyFlag">取消回复</span>-->

</div>

<div @click="like(item)" class="like">

<img src="../../../../../assets/home/carbonStandard/like.png" alt="" v-show="!item.like">

<img src="../../../../../assets/home/carbonStandard/liked.png" alt="" v-show="item.like">

<span v-show="item.likeNum!=0">{{item.likeNum}}</span>

</div>

</div>

</div>

<!-- <div class="reply-box" v-show="item.replyFlag">-->

<!-- <div>-->

<!-- <el-input type="textarea" v-model="commentContent" :rows="3" :placeholder="placeholder" maxlength="200" show-word-limit resize='none'></el-input>-->

<!-- </div>-->

<!-- <div class="content-bottom">-->

<!-- <el-button :disabled="commentContent==''" class="sub" type="primary" @click="saves">发表</el-button>-->

<!-- </div>-->

<!-- </div>-->

<ul class="uls" v-if="item.children">

<li v-for="(items,index) in item.children" v-if="index<item.maxChildren">

<div>

<div class="item-left">

<img :src="items.avatar?imgUrl+items.avatar:avatars" alt="">

</div>

<div class="item-right">

<div class="item-name"><h5>{{items.createByName}}</h5></div>

<div class="item-comment"><span>@{{items.pCreateByName}}</span>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;{{items.content}}</div>

<div class="item-tools">

<div class="item-date li-pointer">{{items.createTime}}<span class="delComment" v-if="items.createBy==name" @click="delComments(items)">删除</span></div>

<div class="replyBtn li-pointer">

<div @click="replys(items)">

<img src="../../../../../assets/home/carbonStandard/xcp-reply.png" alt="">

<span class="reply">回复</span>

</div>

<!-- <span v-show="items.replyFlag">取消回复</span>-->

<div @click="like(items)" class="like">

<img src="../../../../../assets/home/carbonStandard/like.png" alt="" v-show="!items.like">

<img src="../../../../../assets/home/carbonStandard/liked.png" alt="" v-show="items.like">

<span v-show="items.likeNum!=0">{{items.likeNum}}</span>

</div>

</div>

</div>

<!-- <div class="reply-box" v-show="items.replyFlag">-->

<!-- <div>-->

<!-- <el-input type="textarea" v-model="commentContent" :rows="3" :placeholder="placeholder" maxlength="200" show-word-limit resize='none'></el-input>-->

<!-- </div>-->

<!-- <div class="content-bottom">-->

<!-- <el-button :disabled="commentContent==''" class="sub" type="primary" @click="saves">发表</el-button>-->

<!-- </div>-->

<!-- </div>-->

</div>

</div>

<!-- <ul v-if="items.children">-->

<!-- <comment :asideChildren="items.children"></comment>-->

<!-- </ul>-->

</li>

<!-- <comments :asideChildren="item.children"></comments>-->

<div @click="moreChildren(item)" class="moreChildren" v-if="item.maxChildren<item.children.length">查看更多回复</div>

</ul>

</div>

</li>

</ul>

<div class="moreComments" @click="moreComment" v-if="!moreCommentFlag" v-show="commentList.length!=0">查看更多评论</div>

<div class="moreComments" v-else>我是有底线的</div>

</div>

</div>

</div>

<el-dialog

title="评论框"

:visible.sync="dialogVisible"

width="30%"

:before-close="handleClose">

<el-input type="textarea" v-model="commentContent" :rows="3" :placeholder="placeholder" maxlength="200" show-word-limit resize='none'></el-input>

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

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

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

</span>

</el-dialog>

</div>

</template>

<script>

import {addComment, delComment, getCommentList, getContents, likeComment, likedComment} from "@/api/carbonStandard/index";

import "quill/dist/quill.core.css"

import { mapGetters } from 'vuex'

import comments from "@/views/home/ment/carbonStandard/components/comment";

export default {

name: "contents",

data(){

return {

contents: {},

avatars:'https://himg.bdimg.com/sys/portrait/item/wise.1.f718d4b8.Uf7bevSO-PTM1FTV3RWKpg.jpg?time=4346',

text:'',

commentParams:{

appId:'',

parentId:'',//对某个人的评论

mainId:'',//内容id

content:'',//评论内容

type:'',//评论类型：1对人评论，2对项目评论，3对资源评论

blogId:'',//parentId

pageNum:1,

pageSize:5,

},

commentList:[],

commentTotal:'',

imgUrl: process.env.VUE\_APP\_BASE\_API,

replyFlag:false,

placeholder:'',

commentContent:'',

commentContents:'',

dialogVisible: false,

moreCommentFlag:false,

}

},

components:{

comments

},

computed:{

...mapGetters([

'avatar',

'name'

]),

},

created(){

this.getContent()

},

methods:{

download(item){

window.location.href=this.imgUrl + "/common/getMinIoFileById?id="+item.uuid

},

moreChildren(item){

item.maxChildren = (item.children.length+1)

},

moreComment(){

this.commentParams.pageNum = this.commentParams.pageNum+1

getCommentList(this.commentParams).then(res => {

res.rows.map(item => {

item.maxChildren=2

this.commentList.push(item)

})

if(res.total<=(this.commentParams.pageNum\*this.commentParams.pageSize)){

this.moreCommentFlag=true

}

this.commentTotal= res.total

})

},

//点赞

like(item){

if(item.like){

likedComment(item.id).then(res => {

if(res.code==200){

this.$message({

message: '取消点赞成功',

type: 'success'

});

this.getContentsList()

}

})

}else{

likeComment(item.id).then(res => {

if(res.code==200){

this.$message({

message: '点赞成功',

type: 'success'

});

this.getContentsList()

}

})

}

},

//删除评论

delComments(item){

delComment(item.id).then(res => {

if(res.code==200){

this.$message({

message: '删除成功',

type: 'success'

});

this.getContentsList()

}

})

},

handleClose(done) {

this.$confirm('确认关闭？')

.then(\_ => {

done();

})

.catch(\_ => {});

},

reply(items){

// if()

this.commentList.map(item => {

if(item.id!=items.id){

item.replyFlag = false

}

})

items.replyFlag=!items.replyFlag

this.placeholder = '回复'+items.createByName

this.commentParams.appId = this.contents.appId

this.commentParams.parentId = items.id

this.commentParams.mainId = this.contents.uuid

this.commentParams.type = 1

this.commentParams.blogId = this.contents.uuid

},

replys(items){

this.dialogVisible = true

this.placeholder = '回复'+items.createByName

this.commentParams.appId = this.contents.appId

this.commentParams.parentId = items.id

this.commentParams.mainId = this.contents.uuid

this.commentParams.type = 1

this.commentParams.blogId = this.contents.uuid

},

getContent(){

getContents(this.$route.query.id).then(res => {

this.contents=res.data

this.commentParams.appId = this.contents.appId

this.commentParams.mainId = this.contents.uuid

this.getContentsList()

})

},

saves(){

this.commentParams.content = this.commentContent

addComment(this.commentParams).then(res => {

if(res.code=='200'){

this.$message({

message: '发表成功',

type: 'success'

});

this.dialogVisible = false

this.commentParams={

appId:'',

parentId:'',//对某个人的评论

mainId:'',//内容id

content:'',//评论内容

type:'',//评论类型：1对人评论，2对项目评论，3对资源评论

blogId:''//parentId

}

this.getContentsList()

// this.$message('发表成功');

}

})

},

save(){

this.commentParams.content = this.commentContents

this.commentParams.type = 3

this.commentParams.parentId = this.contents.uuid

this.commentContents=''

addComment(this.commentParams).then(res => {

if(res.code=='200'){

this.$message({

message: '发表成功',

type: 'success'

});

this.commentParams={

appId:'',

parentId:'',//对某个人的评论

mainId:'',//内容id

content:'',//评论内容

type:'',//评论类型：1对人评论，2对项目评论，3对资源评论

blogId:''//parentId

}

this.getContentsList()

// this.$message('发表成功');

}

})

},

getContentsList(){

this.commentParams.appId = this.contents.appId

this.commentParams.mainId = this.contents.uuid

getCommentList(this.commentParams).then(res => {

res.rows.map(item => {

item.replyFlag = false

item.maxChildren=2

if(item.children!=null){

item.children.map(items => {

items.replyFlag = false

})

}

if(res.total<=(this.commentParams.pageNum\*this.commentParams.pageSize)){

this.moreCommentFlag=true

}

})

this.commentList = res.rows

this.commentTotal= res.total

})

}

}

}

</script>

<style scoped>

#box{

width: 60%;

}

.box-content{

width: 100%;

border-bottom: 1px solid #dcdcdc;

}

.header-box{

width: 100%;

}

.content-header{

border-bottom: 1px solid #dcdcdc;

}

.content-header .title{

font-weight: 600;

font-size: 4vh;

text-align: center;

/\*line-height: 11vh;\*/

padding: 5vh 5vw;

font-family: "宋体","微软雅黑" ;

}

.content-header .message{

font-size: 2vh;

padding-bottom: 2vh;

}

.content-header .message .date,.content-header .message .readCount{

margin-right: 1vw;

}

.content-header .message .date,.content-header .message .source{

margin-right: 1vw;

}

.comments h3{

font-weight: bold;

}

.comments-box{

align-items: stretch;

}

.user-avatar img{

width: 3vw;

border-radius: 50%;

margin-right: 2vw;

margin-left: 2vw;

margin-top: 2vh;

}

.comments-content{

width: 100%;

/\*height: 10vh;\*/

}

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

height: 10vh;

border-radius: 10px;

}

.content-bottom{

padding: 1vh 0;

}

.comments-list ul li{

display: flex;

/\*height: 12vh;\*/

padding-bottom: 2vh;

}

.comments-list ul li .item-left img{

width: 3vw;

/\*height: 6.2vh;\*/

border-radius: 50%;

margin-right: 2vw;

margin-left: 2vw;

}

.comments-list ul li .item-right{

width: 100%;

}

.comments-list ul li .item-right .item-name h5{

/\*font-weight: bold;\*/

font-size:2.4vh;

margin: 0;

}

.comments-list ul li .item-right .item-comment{

font-size: 1.6vh;

margin: 1.5vh 0;

}

.comments-list ul li .item-right .item-comment span{

color: #00afff;

}

.comments-list ul li .item-right .item-tools{

display: flex;

justify-content: space-between;

font-size: 1.4vh;

padding-bottom: 1vh;

}

.comments-list ul li .item-right .item-tools .item-date{

color: #666666;

}

.comments-list ul li .item-right .item-tools .replyBtn img{

width: 1.2vw;

vertical-align: middle;

}

.uls li{

width: 100%;

}

.uls li>div{

width: 100%;

display: flex;

/\*height: 12vh;\*/

padding-bottom: 2vh;

}

.uls li .item-left img{

width: 3vw;

/\*height: 6.2vh;\*/

border-radius: 50%;

margin-right: 2vw;

margin-left: 2vw;

}

.uls li .item-right{

width: 100%;

}

.uls li .item-right .item-name h5{

/\*font-weight: bold;\*/

font-size:2.4vh;

margin: 0;

}

.uls li .item-right .item-comment{

font-size: 1.6vh;

margin: 1.5vh 0;

}

.uls li .item-right .item-tools{

display: flex;

justify-content: space-between;

font-size: 1.4vh;

padding-bottom: 1vh;

}

.uls li .item-right .item-tools .item-date{

color: #666666;

}

.uls li .item-right .item-tools .replyBtn,.comments-list ul li .item-right .item-tools .replyBtn {

display: flex;

}

.uls li .item-right .item-tools .replyBtn .like,.comments-list ul li .item-right .item-tools .replyBtn .like{

margin-left: 1vw;

color: #f33;

}

.uls li .item-right .item-tools .replyBtn img{

width: 1.2vw;

vertical-align: middle;

}

.delComment {

margin-left: 1vw;

opacity: 0;

}

.item-tools:hover .delComment {

opacity: 1;

}

.moreComments {

height: 6vh;

background: #f5f5f6;

border-radius: 10px;

text-align: center;

line-height: 6vh;

margin-bottom: 3vh;

cursor: pointer;

}

.moreChildren {

width: 10vw;

height: 5vh;

border-radius: 10px;

text-align: center;

line-height: 5vh;

cursor: pointer;

background: #f5f5f6;

font-size: 1.4vh;

color: #00afff;

}

.attachment h3{

font-weight: bold;

}

.attachment{

padding-bottom: 1vh;

}

.attachment ul {

padding-left: 12%;

}

.attachment ul li{

width: 50%;

display: flex;

line-height: 50px;

/\*justify-content: space-between;\*/

/\*border-bottom: 1px solid #dcdcdc;\*/

}

.attachment ul li div:nth-child(2){

margin-left: 2vw;

color: #00afff;

}

/deep/.ql-editor p{

font-size: 18px;

line-height: 1.5;

}

.w-e-text-container [data-slate-editor] p{

}

</style>

<style>

/\*.uls>div>li{\*/

/\* border-bottom: 1px solid #666666;\*/

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

/\*}\*/

</style>

<template>

<div class="app-big-box flex-center">

<!-- <NavBar></NavBar> -->

<div class="app-above">

<div class="app-above-left">

<div class="block">

<!-- 轮播图 -->

<el-carousel trigger="click">

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

<img :src="imgUrl+item.titleImg" alt="" @click="gotoContent(item)">

<!-- <h3 class="small">{{ item.url }}</h3>-->

</el-carousel-item>

</el-carousel>

</div>

</div>

<div class="app-above-right">

<el-tabs v-model="activeName" @tab-click="handleClick">

<el-tab-pane label="国内动态" name="7">

<ul class="searchtit li-pointer">

<div v-if="domesticDynamicList.length==0" class="noData">暂无数据</div>

<li v-else v-for="item in domesticDynamicList" @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div class="data">{{ item.releaseTime | dateFilter }}</div>

</li>

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟双碳人才培训体系研讨会举办</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>区域碳达峰碳中和规划体系1.0正式发布</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟第十八次工作组全会完美收官</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>数字赋能碳达峰碳中和论碳</span><span class="data">6-13</span></li>-->

</ul>

</el-tab-pane>

<el-tab-pane label="国际动态" name="8">

<ul class="searchtit li-pointer">

<div v-if="domesticDynamicList.length==0" class="noData">暂无数据</div>

<li v-for="item in domesticDynamicList" v-else @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div class="data">{{ item.releaseTime | dateFilter }}</div></li>

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟双碳人才培训体系研讨会举办</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>区域碳达峰碳中和规划体系1.0正式发布</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟第十八次工作组全会完美收官</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>数字赋能碳达峰碳中和论碳</span><span class="data">6-13</span></li>-->

</ul>

</el-tab-pane>

<el-tab-pane label="政策法规" name="4">

<ul class="searchtit li-pointer">

<div v-if="domesticDynamicList.length==0" class="noData">暂无数据</div>

<li v-for="item in domesticDynamicList" v-else @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div class="data">{{ item.releaseTime | dateFilter }}</div></li>

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟双碳人才培训体系研讨会举办</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>区域碳达峰碳中和规划体系1.0正式发布</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟第十八次工作组全会完美收官</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>数字赋能碳达峰碳中和论碳</span><span class="data">6-13</span></li>-->

</ul>

</el-tab-pane>

</el-tabs>

</div>

</div>

<div class="app-input">

<div class="app-hrader">

<el-input class="input-pad" prefix-icon="el-icon-search" placeholder="请输入关键字" v-model="retrieveContent" />

<!-- <i class="el-input\_\_icon el-icon-search "></i>-->

<el-button class="jssearch jssearch1" type="primary" @click="retrieve">检索</el-button>

<el-button class="jssearch jssearch2" type="primary" @click="gjretrieve">高级检索</el-button>

</div>

</div>

<div class="app-buttom">

<div class="app-buttom-above">

<div class="app-buttom-above-left">

<el-tabs v-model="activeName1" @tab-click="handleClicks">

<el-tab-pane label="国内标准" name="5">

<ul class="searchtit li-pointer">

<div v-if="bz.length==0" class="noData">暂无数据</div>

<li v-else v-for="item in bz" @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div class="data">{{ item.releaseTime | dateFilter }}</div>

</li>

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟双碳人才培训体系研讨会举办</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>区域碳达峰碳中和规划体系1.0正式发布</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟第十八次工作组全会完美收官</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>数字赋能碳达峰碳中和论碳</span><span class="data">6-13</span></li>-->

</ul>

</el-tab-pane>

<el-tab-pane label="国际标准" name="6">

<ul class="searchtit li-pointer">

<div v-if="bz.length==0" class="noData">暂无数据</div>

<li v-for="item in bz" v-else @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div class="data">{{ item.releaseTime | dateFilter }}</div></li>

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>2022ICT深度观察报告会|数字赋能碳达峰碳中</span><span class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟双碳人才培训体系研讨会举办</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>区域碳达峰碳中和规划体系1.0正式发布</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>工业互联网产业联盟第十八次工作组全会完美收官</span><span-->

<!-- class="data">6-13</span></li>-->

<!-- <li><span><span class="circle"></span>数字赋能碳达峰碳中和论碳</span><span class="data">6-13</span></li>-->

</ul>

</el-tab-pane>

</el-tabs>

<!-- <div class="app-under-box">-->

<!-- <div class="under-top">国内标准 <p class="line"></p>-->

<!-- </div>-->

<!-- <ul class="searchtit li-pointer">-->

<!-- <div v-if="bz.length==0" class="noData">暂无数据</div>-->

<!-- <li v-for="item in bz" v-else @click="gotoContent(item)"><div><div class="circle">·</div><div>{{ item.title }}</div></div><div-->

<!-- class="data">{{ item.releaseTime | dateFilter }}</div></li>-->

<!--&lt;!&ndash; <li><span><span class="circle"></span>积极开展碳标签评价工作对于低碳发展以及双碳目标实现的重要意义</span><span&ndash;&gt;-->

<!--&lt;!&ndash; class="data">6-13</span></li>&ndash;&gt;-->

<!--&lt;!&ndash; <li><span><span class="circle"></span>知会能源成为行业发展的新趋势</span><span class="data">6-13</span>&ndash;&gt;-->

<!--&lt;!&ndash; </li>&ndash;&gt;-->

<!--&lt;!&ndash; <li><span><span class="circle"></span>绿色智能制造创赢计划助力中小企业</span><span class="data">6-13</span>&ndash;&gt;-->

<!--&lt;!&ndash; </li>&ndash;&gt;-->

<!--&lt;!&ndash; <li><span><span class="circle"></span>清洁能源这么好，为何想要转型这么难</span><span class="data">6-13</span>&ndash;&gt;-->

<!--&lt;!&ndash; </li>&ndash;&gt;-->

<!--&lt;!&ndash; <li><span><span class="circle"></span>工业节能减排系统能源管理解决方案</span><span class="data">6-13</span>&ndash;&gt;-->

<!--&lt;!&ndash; </li>&ndash;&gt;-->

<!-- </ul>-->

<!-- </div>-->

</div>

<div class="app-buttom-above-right">

<div class="under-top">互动交流 <p class="line"></p>

</div>

<div class="graphic">

<dl>

<dt><img src="../../../../../assets/home/carbonStandard/opinion.png" alt=""></dt>

<dd>意见征集</dd>

</dl>

<dl>

<dt><img src="../../../../../assets/home/carbonStandard/mail.png" alt=""></dt>

<dd>平台信箱</dd>

</dl>

<dl>

<dt><img src="../../../../../assets/home/carbonStandard/hotline.png" alt=""></dt>

<dd>服务热线</dd>

</dl>

</div>

</div>

</div>

<div class="app-buttom-under">

<div class="appraise" @click="goto('home')"><span>公共服务平台</span>

</div>

<div class="appraise" @click="goto('technology')"><img src="../../../../../assets/home/carbonStandard/evaluation.png" alt=""><span>低碳评价</span>

</div>

<div class="products" @click="goto('carbonProducts')">

<img src="../../../../../assets/home/carbonStandard/product.png" alt=""><span>低碳产品</span>

</div>

<div class="base" @click="goto('skip')">

<img src="../../../../../assets/home/carbonStandard/database.png" alt=""><span>低碳数据库</span></div>

<div class="serve" @click="goto('refinery')">

<img src="../../../../../assets/home/carbonStandard/service.png" alt=""><span>低碳服务</span></div>

</div>

</div>

<!-- <Footer></Footer> -->

</div>

</template>

<script>

// import NavBar from './NavBar.vue'

// import Footer from './Footer.vue'

import {getList, getListTemplate,getListRotationChart} from "@/api/carbonStandard/index";

export default {

data() {

return {

activeName: "7",

activeName1:'5',

swiper:[],

queryParams: {

pageNum: 1,

pageSize: 6,

appName: null,

delState: 0,

appId:4,

channelId:7

},

//国内动态

domesticDynamicList:[],

bz:[],

swiper\_queryParams:{

pageNum: 1,

pageSize: 6,

appName: null,

appId:4,

channelIds:[7,8,4]

},

retrieveContent:'',

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

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

};

},

created() {

},

filters:{

dateFilter(original){

var dt = new Date(original);

var m = (dt.getMonth()+1).toString().padStart(2,0) //月

var d = (dt.getDate()).toString().padStart(2,0) //日

//返回日期格式

return `${m}-${d}`

}

},

mounted() {

this.getlist(7);

this.getBzList(5);

this.getListRotationCharts()

},

methods: {

//高级检索

gjretrieve(){

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

path: "/carbonStandard/main/retrieve"

});

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

},

//检索

retrieve(){

if(this.retrieveContent!=''){

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

path: "/carbonStandard/main/retrieve",

query:{content:this.retrieveContent}

});

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

}

},

//获取轮播图列表

getListRotationCharts(){

getListRotationChart(this.swiper\_queryParams).then(res => {

this.swiper= res.rows

})

},

getlist(cId){

this.queryParams.channelId = cId;

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

this.domesticDynamicList=res.rows

})

},

getBzList(cId){

this.queryParams.channelId = cId;

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

this.bz=res.rows

})

},

handleClick(tab, event) {

this.getlist(tab.name)

},

handleClicks(tab, event) {

this.getBzList(tab.name)

},

gotoContent(item) {

getListTemplate(item.uuid).then(res => {

if (res.code!=200){

this.$message.error(res.msg);

}else {

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

path: "/carbonStandard/main/content",

query:{id:item.uuid}

});

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

// this.$router.push({

// path: "/carbonStandard/main/content",

// query:{id:item.uuid}

// });

}

})

},

goto(item){

if(item=='technology'){

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

path: "/technology/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");

}

},

},

// components: { NavBar, Footer }

};

</script>

<style lang="scss">

.app-big-box {

width: 100%;

display: flex;

flex-direction: column;

.app-above {

width: 62.5%;

height: 36.3vh;

margin-top: 1%;

display: flex;

justify-content: space-between;

.app-above-left {

width: 57.4%;

height: 100%;

//margin-right: 2vw;

.block {

width: 100%;

height: 100%;

img{

width: 100%;

}

//.el-carousel\_\_item h3 {

// color: #475669;

// font-size: 14px;

// opacity: 0.75;

// line-height: 150px;

// margin: 0;

//}

.el-carousel\_\_item:nth-child(2n) {

background-color: #99a9bf;

}

.el-carousel\_\_item:nth-child(2n+1) {

background-color: #d3dce6;

}

}

}

.app-above-right {

width: 40%;

height: 100%;

.searchtit {

margin-top: 5px;

width: 100%;

li {

width: 100%;

line-height: 5vh;

display: flex;

justify-content: space-between;

color: #5d5d5d;

&>span{

display: inline-block;

}

&>div:nth-child(1){

//width: 90%;

display: flex;

justify-content: space-between;

align-items: center;

overflow: hidden;

}

&>div>div:nth-child(2){

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

}

&>div:nth-child(2){

//overflow:hidden;

//text-overflow:ellipsis;

white-space:nowrap;

}

.circle {

//width: 5px;

//height: 5px;

//background-color: #5d5d5d;

//border-radius: 50%;

display: inline-block;

margin-right: 5px;

font-size: 4vh;

}

.data {

//width: 40px;

color: #989898;

font-size: 10px;

}

}

}

}

}

.react-item {

width: 36%;

.notice {

width: 100%;

float: right;

position: relative;

}

span.more {

position: absolute;

right: 0;

top: 0;

color: #999999;

padding: 8px;

}

.notice {

border: none;

box-shadow: none;

}

}

.app-input {

width: 62.5%;

height: 70px;

margin-top: 1%;

background-color: #e0ecfc;

display: flex;

justify-content: center;

align-items: center;

.app-hrader {

display: flex;

justify-content: center;

align-items: center;

//.el-icon-search {

// position: absolute;

// top: 126px;

// left: 495px;

//}

.input-pad {

width: 400px;

margin-right: 0.8vw;

.el-input\_\_inner {

text-indent: 10px;

}

}

.jssearch {

border-bottom-right-radius: 5px;

border-top-right-radius: 5px;

}

}

}

.app-buttom {

width: 62.5%;

height: 400px;

margin-top: 2%;

.app-buttom-above {

width: 100%;

height: 70%;

display: flex;

.app-buttom-above-left {

width: 50%;

height: 100%;

padding-right: 4vw;

.searchtit {

margin-top: 5px;

width: 100%;

li {

width: 100%;

line-height: 5vh;

display: flex;

justify-content: space-between;

color: #5d5d5d;

&>span{

display: inline-block;

}

&>div:nth-child(1){

//width: 90%;

display: flex;

justify-content: space-between;

align-items: center;

overflow: hidden;

}

&>div>div:nth-child(2){

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

}

&>div:nth-child(2){

//overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

}

.circle {

//width: 5px;

//height: 5px;

//background-color: #5d5d5d;

//border-radius: 50%;

display: inline-block;

margin-right: 5px;

font-size: 4vh;

}

.data {

//width: 40px;

color: #989898;

font-size: 10px;

}

}

}

//.app-under-box {

// width: 100%;

//

// .under-top {

// width: 100%;

// font-size: 17px;

// border-bottom: 1px solid #ccc;

// font-weight: bold;

// line-height: 35px;

// position: relative;

//

// .line {

// width: 80px;

// height: 3px;

// background-color: #4797ee;

// position: absolute;

// padding: 0;

// margin: 0;

// top: 34px;

// }

// }

//

// .searchtit {

// margin-top: 5px;

// li {

// line-height: 38px;

// display: flex;

// justify-content: space-between;

// color: #5d5d5d;

// &>div:nth-child(1){

// width: 90%;

// display: flex;

// align-items: center;

// overflow: hidden;

// }

// &>div>div:nth-child(2){

// overflow:hidden;

// text-overflow:ellipsis;

// white-space:nowrap;

//

// }

// &>div:nth-child(2){

// //overflow:hidden;

// //text-overflow:ellipsis;

// white-space:nowrap;

// }

// .circle {

// //width: 5px;

// //height: 5px;

// //background-color: #5d5d5d;

// //border-radius: 50%;

// display: inline-block;

// margin-right: 5px;

// font-size: 4vh;

// }

//

// .data {

// //width: 3vw;

// color: #989898;

// font-size: 10px;

// }

// }

// }

//}

}

.app-buttom-above-right {

width: 50%;

height: 100%;

.under-top {

width: 100%;

font-size: 17px;

border-bottom: 1px solid #ccc;

font-weight: bold;

line-height: 35px;

position: relative;

.line {

width: 80px;

height: 3px;

background-color: #4797ee;

position: absolute;

padding: 0;

margin: 0;

top: 34px;

}

}

.graphic {

height: 100%;

display: flex;

justify-content: space-between;

align-items: center;

dl {

//margin-left: 50px;

//margin-top: 90px;

line-height: 30px;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

}

dt>img {

margin-left: 35px;

}

dd {

text-align: center;

}

}

}

}

.app-buttom-under {

width: 100%;

height: 30%;

margin-top: 2%;

display: flex;

justify-content: center;

.appraise {

width: 22%;

height: 6vh;

margin-left: 20px;

margin-top: 10px;

display: flex;

justify-content: center;

align-items: center;

color: #fff;

background-image: linear-gradient(#5a92f0,#2057b1);

// background-color: #4d83de;

//box-shadow: 0 0 10px #4d83de;

img {

margin-right: 15px;

}

}

.products {

width: 22%;

height: 6vh;

margin-left: 20px;

margin-top: 10px;

display: flex;

justify-content: center;

align-items: center;

color: #fff;

background-image: linear-gradient(#4ee7eb,#17b5b9);

// background-color: #3fdde4;

img {

margin-right: 15px;

}

}

.base {

width: 22%;

height: 6vh;

margin-left: 20px;

margin-top: 10px;

display: flex;

justify-content: center;

align-items: center;

color: #fff;

background-image: linear-gradient(#55b6f9,#0973bd);

//background-color: #49adf3;

img {

margin-right: 15px;

}

}

.serve {

width: 22%;

height: 6vh;

margin-left: 20px;

margin-top: 10px;

display: flex;

justify-content: center;

align-items: center;

color: #fff;

background-image: linear-gradient(#3dcae9,#0a89a5);

//background-color: #32c4e3;

img {

margin-right: 15px;

}

}

}

}

}

.noData{

text-align: center;

}

</style>

<style scoped>

/deep/ .el-carousel\_\_container{

height: 100%;

}

/deep/ .el-carousel--horizontal{

height: 100%;

}

/deep/ .el-button.jssearch1 {

background-color: #3aa2eb;

border: none;

}

/deep/ .el-button.jssearch2 {

background-color: #3bc9e8;

border: none;

}

/deep/ .el-tabs\_\_item {

font-size: 16px;

}

</style>

<template>

<div class="app-box">

<h3><img src="../img/0400.jpg" alt="" style="background: #438ae2;" /><b>{{ name }}</b>

</h3>

<div class="breadcrumb">

<el-breadcrumb separator="/">

<el-breadcrumb-item :to="{ path: '/carbonStandard/main/index' }">首页</el-breadcrumb-item>

<el-breadcrumb-item><a href="/carbonStandard/main/list">{{ name }}</a></el-breadcrumb-item>

</el-breadcrumb>

</div>

<!-- <el-table v-loading="loading" :data="tableData" class="table" :show-header="false">-->

<!-- <el-table-column prop="title" min-width="90"/>-->

<!-- <el-table-column prop="releaseTime" min-width="10" :formatter="dateFilter" />-->

<!-- </el-table>-->

<ul class="box-content li-pointer">

<div v-if="tableData.length==0" style="text-align: center">暂无数据</div>

<li v-else v-for="(item,index) in tableData" :key="index" class="flex\_between font16" @click="gotoContent(item)">

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

<div>{{ parseTime(item.releaseTime, '{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"

/>

</div>

</template>

<script>

import {getList,getListTemplate} from "@/api/carbonStandard/index";

export default {

metaInfo() {

return {

title: this.$route.query.name

}

},

data() {

return {

// 遮罩层

loading: true,

// 列表数据

tableData: [],

// 总条数

total: 0,

pageSize:['14'],

// 查询参数

queryParams: {

pageNum: 1,

pageSize: 14,

companyName: null,

companyAbb: null,

companyParent: null,

companyZone: null,

companyState: null,

companyCity: null,

companyArea: null,

appId:this.$route.query.appId,

channelId:this.$route.query.channelId

},

name:this.$route.query.name

}

},

created() {

this.getList()

},

methods: {

getList() {

this.loading = true;

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

if (response.code!=200){

this.$message.error(response.msg);

}else {

this.tableData = response.rows;

this.total = response.total;

this.loading = false;

}

})

},

gotoContent(item) {

getListTemplate(item.uuid).then(res => {

if (res.code!=200){

this.$message.error(res.msg);

}else {

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

path: "/carbonStandard/main/"+res.data.tempRoute,

query:{id:item.uuid}

});

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

}

})

}

},

computed: {

userdata() {

let start = (this.pageNum - 1) \* this.pageSize;

let end = this.pageNum \* this.pageSize;

return this.number.slice(start, end)

}

},

watch:{

"$route":function(to,from){

this.queryParams.channelId=to.query.channelId

this.queryParams.appId = to.query.appId

this.name = to.query.name

this.getList()

//from 对象中包含当前地址

//to 对象中包含目标地址

//其实还有一个next参数的，这个参数是控制路由是否跳转的，如果没写，可以不用写next()来代表允许路由跳转，如果写了就必须写next(),否则路由是不会生效的。

},

}

}

</script>

<style lang="scss">

.app-box {

width: 80%;

height: 100%;

display: flex;

align-items: center;

flex-direction: column;

margin: 0 auto;

h3 {

width: 80%;

height: 40px;

background-color: #eeeeee;

color: #438ae2;

line-height: 40px;

img {

margin-left: 2%;

}

}

.box-content{

width: 80%;

height: 63vh;

}

.box-content li{

line-height: 4.4vh;

border-bottom: 1px solid #eeeeee;

padding-left: 1vw;

//font-size: 1vh;

}

.box-content li:hover{

background: #eeeeee;

}

.box-content li div:nth-child(1){

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

}

.box-content li div:nth-child(2){

width: 5vw;

text-align: center;

}

//.table {

// width: 80%;

// display: flex;

// flex-direction: column;

// justify-content: space-between;

// // text-align: center;

//

//}

//.el-table\_1\_column\_2 .cell {

// margin-left: 80%;

//}

.breadcrumb {

width: 80%;

height: 30px;

display: flex;

}

}

</style>

<style scoped>

.el-table tr{

height: 4.7vh;

}

.pagination-container{

width: 80%;

}

/deep/ .el-pagination{

transform:translate(-43%,0%);

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

/\*margin-top:20px;\*/

}

/\*/deep/ .el-table--enable-row-hover .el-table\_\_body tr:hover > td.el-table\_\_cell{\*/

/\* width: 90%;\*/

/\*}\*/

/\*.pagination-container{\*/

/\* width: 47%;\*/

/\*}\*/

</style>

<template>

<div class="app-box">

<div class="box-top">

<el-input class="input-pad" prefix-icon="el-icon-search" placeholder="请输入关键字" v-model="retrieveContent" />

<el-button class="jssearch jssearch1" type="primary" @click="retrieve">检索</el-button>

<el-button class="jssearch jssearch2" type="primary" @click="dialogVisible=true">高级检索</el-button>

</div>

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

<!-- <el-breadcrumb separator="/">-->

<!-- <el-breadcrumb-item :to="{ path: '/carbonStandard/main/index' }">首页</el-breadcrumb-item>-->

<!-- <el-breadcrumb-item><a href="/carbonStandard/main/list">{{ name }}</a></el-breadcrumb-item>-->

<!-- </el-breadcrumb>-->

<!-- </div>-->

<!-- <el-table v-loading="loading" :data="tableData" class="table" :show-header="false">-->

<!-- <el-table-column prop="title" min-width="90"/>-->

<!-- <el-table-column prop="releaseTime" min-width="10" :formatter="dateFilter" />-->

<!-- </el-table>-->

<ul class="box-content li-pointer">

<li v-for="(item,index) in tableData" :key="index" class=" font16" @click="gotoContent(item)">

<div v-html="item.title" class="item-title"></div>

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

<!-- <div>{{ item.releaseTime | dateFilter }}</div>-->

</li>

</ul>

<pagination

v-show="total>0"

:total="total"

:page-sizes='pageSize'

:page.sync="queryParams.pageNum"

:limit.sync="queryParams.pageSize"

@pagination="retrieve"

/>

<el-dialog

title="检索条件"

:visible.sync="dialogVisible"

width="50%"

:before-close="handleClose">

<div>

<el-row :gutter="16">

<el-form ref="form" :model="form" :rules="rules" label-width="100px">

<el-col :span="12">

<el-form-item label="所属栏目" prop="channelId">

<el-select v-model="form.channelId" placeholder="请选择所属栏目" style="width: 100%;">

<el-option

v-for="dict in columnList"

:key="dict.id"

:label="dict.name"

:value="parseInt(dict.id)"

></el-option>

</el-select>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="标题" prop="title">

<el-input v-model="form.title" placeholder="请输入标题"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="日期" prop="retrieveTime">

<!-- <el-date-picker clearable-->

<!-- v-model="form.releaseTime"-->

<!-- type="date"-->

<!-- value-format="yyyy-MM-dd"-->

<!-- placeholder="请选择发布日期" style="width: 100%;">-->

<!-- </el-date-picker>-->

<el-date-picker

v-model="retrieveTime"

type="daterange"

range-separator="至"

start-placeholder="开始日期"

end-placeholder="结束日期">

</el-date-picker>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="来源" prop="title">

<el-input v-model="form.source" placeholder="请输入来源"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="作者" prop="title">

<el-input v-model="form.createBy" placeholder="请输入作者"/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item label="内容" prop="title">

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

</el-form-item>

</el-col>

<!-- <el-col :span="12">-->

<!-- <el-form-item label="标题" prop="channelId">-->

<!-- <el-select v-model="form.contentTempId" placeholder="请选择模版" style="width: 100%;">-->

<!-- <el-option-->

<!-- v-for="dict in contentTempList"-->

<!-- :key="dict.id"-->

<!-- :label="dict.tempName"-->

<!-- :value="parseInt(dict.id)"-->

<!-- ></el-option>-->

<!-- </el-select>-->

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

<!-- </el-col>-->

<!-- <el-col :span="24">-->

<!-- <el-form-item label="标题" prop="title">-->

<!-- <el-input v-model="form.title" placeholder="请输入标题"/>-->

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

<!-- </el-col>-->

<!-- <el-col :span="24">-->

<!-- <el-form-item label="副标题" prop="subTitle">-->

<!-- <el-input v-model="form.subTitle" placeholder="请输入副标题"/>-->

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

<!-- </el-col>-->

<!-- <el-col :span="12">-->

<!-- <el-form-item label="标题图片">-->

<!-- <image-upload-min-i-o v-model="form.titleImg" :limit="1"/>-->

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

<!-- </el-col>-->

<!-- <el-col :span="6">-->

<!-- <el-form-item label="热点" prop="isHot">-->

<!-- <el-radio-group v-model="form.isHot">-->

<!-- <el-radio-->

<!-- v-for="dict in dict.type.cms\_content\_hot"-->

<!-- :key="dict.value"-->

<!-- :label="dict.value"-->

<!-- >{{dict.label}}-->

<!-- </el-radio>-->

<!-- </el-radio-group>-->

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

<!-- </el-col>-->

<!-- <el-col :span="6">-->

<!-- <el-form-item label="轮播图" prop="rotationChart">-->

<!-- <el-radio-group v-model="form.rotationChart">-->

<!-- <el-radio-->

<!-- v-for="dict in dict.type.cms\_content\_rotation\_chart"-->

<!-- :key="dict.value"-->

<!-- :label="dict.value"-->

<!-- >{{dict.label}}-->

<!-- </el-radio>-->

<!-- </el-radio-group>-->

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

<!-- </el-col>-->

<!-- <el-col :span="12">-->

<!-- <el-form-item label="发布日期" prop="releaseTime">-->

<!-- <el-date-picker clearable-->

<!-- v-model="form.releaseTime"-->

<!-- type="date"-->

<!-- value-format="yyyy-MM-dd"-->

<!-- placeholder="请选择发布日期" style="width: 100%;">-->

<!-- </el-date-picker>-->

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

<!-- </el-col>-->

</el-form>

</el-row>

</div>

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

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

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

</span>

</el-dialog>

</div>

</template>

<script>

import {getList, getListTemplate, getNavList} from "@/api/carbonStandard/index";

export default {

name: "retrieve",

data(){

return{

retrieveContent:this.$route.query.content,

// 列表数据

tableData: [],

total: 0,

pageSize:['10'],

queryParams: {

pageNum: 1,

pageSize: 10,

companyName: null,

companyAbb: null,

companyParent: null,

companyZone: null,

companyState: null,

companyCity: null,

companyArea: null,

appId:4,

globalQuery:''

// channelId:this.$route.query.channelId

},

columnQueryParams:{

appId:4

},

dialogVisible:false,

form:{

appId:4,

pageNum: 1,

pageSize: 10,

},

// 表单校验

rules:{

},

columnList:[],

retrieveTime:''

}

},

mounted(){

this.retrieve()

this.getColumnList()

},

methods:{

retrieveSub(){

this.form.releaseTimeStart = this.retrieveTime[0]

this.form.releaseTimeEnd = this.retrieveTime[1]

getList(this.form).then(res => {

this.tableData = res.rows

this.total = res.total

this.dialogVisible = false

})

},

getColumnList(){

getNavList(this.columnQueryParams).then(res => {

this.columnList=res.rows

})

},

handleClose(done) {

this.$confirm('确认关闭？')

.then(\_ => {

done();

})

.catch(\_ => {});

},

retrieve(){

this.queryParams.globalQuery = this.retrieveContent

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

this.tableData = res.rows

this.total = res.total

})

},

gotoContent(item){

getListTemplate(item.uuid).then(res => {

if (res.code!=200){

this.$message.error(res.msg);

}else {

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

path: "/carbonStandard/main/content",

query:{id:item.uuid}

});

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

// this.$router.push({

// path: "/carbonStandard/main/content",

// query:{id:item.uuid}

// });

}

})

}

}

}

</script>

<style lang="scss">

.app-box {

width: 80%;

height: 100%;

display: flex;

align-items: center;

flex-direction: column;

margin: 0 auto;

.box-top{

width: 80%;

height: 10vh;

background-color: #e0ecfc;

margin-top: 1vh;

display: flex;

align-items: center;

justify-content: center;

.input-pad{

width: 25%;

margin-right: 1vw;

}

}

.box-content{

width: 80%;

}

.box-content li{

line-height: 4.4vh;

border-bottom: 1px solid #eeeeee;

padding-left: 1vw;

//font-size: 1vh;

}

.box-content li:hover{

background: #eeeeee;

}

.box-content li div:nth-child(1){

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

}

//.box-content li div:nth-child(2){

// width: 5vw;

// text-align: center;

//}

.breadcrumb {

width: 80%;

height: 30px;

display: flex;

}

}

</style>

<style scoped>

.el-table tr{

height: 4.7vh;

}

.pagination-container{

width: 80%;

height: 6.2vh;

}

/deep/ .el-pagination{

transform:translate(-100%,0%);

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

/\*margin-top:20px;\*/

}

/deep/ .el-button.jssearch1 {

background-color: #3aa2eb;

border: none;

}

/deep/ .el-button.jssearch2 {

background-color: #3bc9e8;

border: none;

}

.item-title{

display: flex;

align-items: center;

font-weight: bold;

}

.item-content{

width: 100%;

height: 6.5vh;

/\*display: flex;\*/

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

/\*font-weight: normal;\*/

overflow:hidden;

text-overflow:ellipsis;

white-space:nowrap;

font-size: 1.4vh;

color: #666666;

}

.item-content p strong{

font-weight: normal;

font-weight: normal!important;

}

</style>

<template>

<div class="bottom">

<div>版权所有@中国石化石油化工科学院<span>地址：中国北京市海淀区学院路18号</span></div>

<div>邮编：10083<span>ICP备案序号：京ICP备05037230</span></div>

</div>

</template>

<script>

</script>

<style lang="scss">

.bottom {

width: 100%;

height: 80px;

background-color: #e1e9ec;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

color: #335875;

font-size: 13px;

}

</style>

<template>

<div class="bottom">

<div>版权所有@中国石化石油化工科学院<span>地址：中国北京市海淀区学院路18号</span></div>

<div>邮编：10083<span>ICP备案序号：京ICP备05037230</span></div>

</div>

</template>

<script>

</script>

<style lang="scss">

.bottom {

width: 100%;

height: 80px;

background-color: #e1e9ec;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

color: #335875;

font-size: 13px;

}

</style>

<template>

<el-main>

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

</el-main>

</template>

<script>

export default {

name: 'cmsMain',

computed: {

key() {

return this.$route.path

}

}

}

</script>

<style scoped>

/\*.el-main {\*/

/\* padding: 20px 0;\*/

/\* min-height: 650px;\*/

/\*}\*/

/deep/ .el-main {

padding:10px 0;

min-height:868px;

overflow-y: hidden;

}

/\*.el-main {\*/

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

/\* \*/

/\* overflow-y: hidden;\*/

/\*}\*/

</style>

<template>

<div class="app-top">

<div class="app-top-abover">

<div class="app-title">

<img src="../../../../assets/home/carbonFootprint/logo.png" alt="">

<h3>低碳标准与政策法规</h3>

</div>

<div class="app-text">

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

<el-input class="search" placeholder="请输入要查询的内容" suffix-icon="el-icon-search" size="mini">

</el-input>

<el-button class="but-one" size="mini" type="primary" @click="tologin">注册</el-button>

<el-button class="but-two" size="mini" type="primary" @click="tologin">登录</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 class="search" placeholder="请输入要查询的内容" suffix-icon="el-icon-search" size="mini">

</el-input>

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

<div class="app-top-under">

<ul>

<li @click="home"><img src="../../../../assets/home/carbonStandard/home.png" alt="" />首页</li>

<li @click="policy(item)" v-for="(item,index) in navList" :key="index">

<img :src="require('../../../../assets/home/carbonStandard/'+item.id+'.png')" alt="" />

{{ item.name }}</li>

<!-- <li><img src="../../../../assets/home/carbonStandard/standard.png" alt="" />国内标准</li>-->

<!-- <li><img src="../../../../assets/home/carbonStandard/internationalStandard.png" alt="" />国际标准</li>-->

<!-- <li><img src="../../../../assets/home/carbonStandard/domestic.png" alt="" />国内动态</li>-->

<!-- <li><img src="../../../../assets/home/carbonStandard/international.png" alt="" />国际动态</li>-->

<!-- <li><img src="../../../../assets/home/carbonStandard/technology.png" alt="" />低碳技术</li>-->

<!-- <li><img src="../../../../assets/home/carbonStandard/market.png" alt="" />碳市场</li>-->

</ul>

</div>

</div>

</template>

<script>

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

import { mapGetters } from "vuex";

import router from '../../../../router'

import {getNavList} from "@/api/carbonStandard/index";

export default {

name: 'navbar',

data() {

return {

islogin: false,

queryParams:{

appId:4

},

navList:[]

}

},

computed: {

...mapGetters([

'avatar',

'name'

]),

},

created() {

this.login();

this.getNavList()

},

methods: {

getNavList(){

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

this.navList=res.rows

})

},

home(){

this.$router.push({

path: "/carbonStandard/main/index",

});

},

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

redirect: this.$route.fullPath

},

});

},

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 = '/carbonStandard';

})

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

},

policy(item) {

this.$router.push({

path: "/carbonStandard/main/"+item.listTempRoute,

query:{

appId:item.appId,

channelId:item.id,

name:item.name

}

});

}

}

}

</script>

<style lang="scss">

.avatar-container {

margin-right: 30px;

}

.avatar-Name {

margin-left: 10px;

cursor: pointer;

float: right;

font-size: 16px;

color: #ffffff;

white-space:nowrap;

}

.app-top {

width: 100%;

height: 12.23vh;

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

background-size: cover;

display: flex;

justify-content: space-between;

flex-direction: column;

align-items: center;

.app-top-abover {

width: 100%;

height: 70%;

display: flex;

justify-content: space-between;

.app-title {

display: flex;

align-items: center;

img {

width: 45px;

height: 45px;

margin: 0 8px 0 30px;

}

}

h3 {

color: #fff;

}

.app-text {

display: flex;

div:nth-child(1) {

height: 100%;

display: flex;

align-items: center;

justify-content: center;

.input {

width: 300px;

margin-right: 5px;

background-color: rgba(111, 150, 255, 0.8);

}

.but-one {

margin: 0 5px;

}

.but-two {

margin: 0 20px 0 5px;

}

}

}

}

.app-top-under {

width: 62.5%;

height: 30%;

ul {

display: flex;

justify-content: space-between;

align-items: center;

color: #fff;

height: 100%;

li {

//margin: 10px;

cursor:pointer;

//display: flex;

img {

height: 1.5vh;

vertical-align: middle;

margin-right: 5px;

}

}

}

}

}

</style>

<style scoped>

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

background: rgba(255, 255, 255, 0.3);

border: none;

color: #FFFFFF;

}

.el-button {

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

border: none;

}

</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id： 1 低碳技术评价平台 2 低碳产品检验检测与评价平台 3 石化行业碳足迹数据库 4 低碳标准与政策法规 5 炼化企业低碳技术服务

\* @param channelId 栏目id 4政策法规 5国内标准 6国际标准 7国内动态 8国际动态 9低碳技术 10碳市场

\* @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));

}

}