<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>复合材料增强织物多尺度模拟软件-用户登录</title>

</head>

<body>

<form id="loginForm" method="post">

<label for="username">用户名：</label>

<input type="text" name="username" id="username" required>

<br>

<label for="password">密码：</label>

<input type="password" name="password" id="password" required>

<br>

<label for="captcha">验证码：</label>

<input type="text" name="captcha" id="captcha" required>

<img src="{:url('captcha')}" onclick="this.src='{:url('captcha')}?'+Math.random()"

alt="验证码">

<br>

<button type="submit">登录</button>

</form>

<script>

document.getElementById('loginForm').addEventListener('submit', function (event) {

event.preventDefault();

var formData = new FormData(event.target);

fetch('/login', {

method: 'POST',

body: formData

}).then(function (response) {

return response.json();

}).then(function (result) {

if (result.code === 1) {

alert(result.msg);

location.href = '/dashboard'; // 跳转到后台复合材料增强织物多尺度模拟

软件

} else {

alert(result.msg);

}

});

});

</script>

</body>

</html>

// application/common/model/User.php

// 凭证检查

namespace app\common\model;

use think\Model;

class User extends Model

{

// 设置当前模型对应的完整数据表名称

protected $table = 'users';

// 自动写入时间戳

protected $autoWriteTimestamp = 'int';

// 密码加密验证

public function checkPass($password)

{

return password\_verify($password, $this->getData('password'));

}

}

// application//controller/Login.php

namespace app\\controller;

use app\BaseController;

use app\common\model\User;

use think\facade\Request;

class Login extends BaseController

{

public function index()

{

// 显示复合材料增强织物多尺度模拟软件登录页面

return view();

}

public function login()

{

$request = Request::instance();

$username = $request->param('username');

$password = $request->param('password');

if (!$username || !$password) {

return json(['code' => 400, 'msg' => '用户名或密码不能为空']);

}

$userModel = new User();

$result = $userModel->where('username', $username)->find();

if (!$result) {

return json(['code' => 404, 'msg' => '用户不存在']);

}

if (!$result->checkPass($password)) {

return json(['code' => 401, 'msg' => '密码错误']);

}

// 登录成功，设置session或token等认证信息

session('user\_id', $result->id);

return json(['code' => 200, 'msg' => '登录成功']);

}

}

// route/.php

use think\facade\Route;

// 复合材料增强织物多尺度模拟软件登录页面

Route::get('login', '/Login/index');

// 复合材料增强织物多尺度模拟软件登录处理

Route::post('login', '/Login/login');

<template>

<div

class="fhclzqmaterialconfigalbum-list ele-body page-list"

:class="frameworkTheme">

<el-card shadow="never" v-if="!isSimple" :body-style="searchCardStyle">

<!-- 材料配置管理 搜索 -->

<el-form

v-if="!isSimple"

id="gas\_el-search-form"

:model="where"

inline

class="ele-form-search search-form"

@keyup.enter.native="getList"

@submit.native.prevent

>

<el-row :gutter="15">

<el-col :xs="24" :sm="12" :lg="6">

<el-form-item label="存储条件:" label-width="6em">

<el-input

clearable

v-model="where.storage\_conditions"

placeholder="请输入存储条件" />

</el-form-item>

</el-col>

<el-col :xs="24" :sm="12" :lg="6">

<el-button

type="primary"

:style="

frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''

"

icon="el-icon-search"

class="ele-btn-icon btn-search"

@click="getList(true)"

>查询</el-button

>

<el-button

:style="

frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''

"

@click="reset"

>重置</el-button

>

</el-col>

</el-row>

</el-form>

</el-card>

<el-card shadow="never" class="fhclzqmaterialconfigalbum-list\_\_wrap">

<div class="card-header" slot="header" v-if="!isSimple">

<span>材料配置管理</span>

<div style="float: right">

<el-button

size="small"

type="primary"

icon="el-icon-plus"

class="ele-btn-icon btn-add"

@click="openEdit(null)"

:style="

frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''

"

v-if="permission.includes('sys:fhclzqmaterialconfigalbum:add')"

>添加

</el-button>

</div>

</div>

<div class="card-list">

<template v-for="(num, i) in 3">

<div class="card-template" :key="i">

<div class="card-list\_\_item" v-for="(item, index) in list" :key="index">

<div class="card-list--wrapper">

<div class="card-list\_\_item--left">

<div class="title">

{{ item.title }}

<el-link

type="primary"

:underline="false"

class="gas\_el--edit-btn btn-edit"

icon="el-icon-edit"

@click="openEdit(item)"

v-if="permission.includes('sys:fhclzqmaterialconfigalbum:edit')"

>编辑

</el-link>

<el-popconfirm

class="ele-action"

title="确定要删除吗？"

@confirm="remove(item)"

>

<el-link

type="danger"

class="btn-delete"

slot="reference"

:underline="false"

icon="el-icon-delete"

v-if="

permission.includes('sys:fhclzqmaterialconfigalbum:delete')

"

>删除

</el-link>

</el-popconfirm>

</div>

<div class="content" v-if="item.description">{{ item.description }}</div

>

<div class="desc">

<el-row :gutter="10">

<el-col :xs="24" :sm="12" :lg="8">

<div class="desc-item">

<div class="desc-item\_\_label">材料名称：</div>

<div class="desc-item\_\_value">

{{item.material\_name}}

</div>

</div>

</el-col>

<el-col :xs="24" :sm="12" :lg="8">

<div class="desc-item">

<div class="desc-item\_\_label">材料类型：</div>

<div class="desc-item\_\_value">

{{item.material\_type}}

</div>

</div>

</el-col>

<el-col :xs="24" :sm="12" :lg="8">

<div class="desc-item">

<div class="desc-item\_\_label">材料属性：</div>

<div class="desc-item\_\_value">

{{item.properties}}

</div>

</div>

</el-col>

<el-col :xs="24" :sm="12" :lg="8">

<div class="desc-item">

<div class="desc-item\_\_label">生产商：</div>

<div class="desc-item\_\_value">

{{item.manufacturer}}

</div>

</div>

</el-col>

<el-col :xs="24" :sm="12" :lg="8">

<div class="desc-item">

<div class="desc-item\_\_label">存储条件：</div>

<div class="desc-item\_\_value">

{{item.storage\_conditions}}

</div>

</div>

</el-col>

</el-row>

</div>

</div>

<div class="card-list\_\_item--right">

<el-image

style="width: 280px; height: 175px; border-radius: 4px"

:src="item.image"

:preview-src-list="previewList"

/>

</div>

</div>

<el-divider />

</div>

</div>

</template>

</div>

<div class="card-overflow"></div>

</el-card>

<div class="fhclzqmaterialconfigalbum-list\_\_pagination" v-if="!isSimple">

<el-pagination

class="ele-pagination-circle"

background

layout="total, sizes, prev, pager, next, jumper"

:current-page="currentPage"

:page-size="10"

:total="total"

>

</el-pagination>

</div>

<!-- 编辑弹窗 -->

<fhclzqmaterialconfigalbumEdit

:data="current"

:visible.sync="showEdit"

@done="getList"

/>

</div>

</template>

<script>

import { mapGetters } from "vuex";

import setting from "@/config/setting";

import dayjs from '@/vendor/dayjs.min'

import fhclzqmaterialconfigalbumEdit from "./fhclzqmaterialconfigalbum-edit";

export default {

name: "fhclzq\_material\_config\_album",

components: { fhclzqmaterialconfigalbumEdit },

computed: {

...mapGetters(["permission"]),

previewList() {

return this.list.map((item) => item.image);

},

searchCardStyle() {

let style = {};

if (this.frameworkTheme === "BootstrapWeixin") {

style = {

padding: 0,

};

}

return style;

}

},

props: {

isSimple: {

type: Boolean,

default: false,

},

},

mounted() {

this.getList(true);

this.total = this.generateNumber("材料配置管理");

this.currentPage = Math.floor(

Math.random() \* Math.ceil(this.total / 10) + 1

);

},

data() {

return {

// 表格数据接口

url: "/fhclzqmaterialconfigalbum/index",

// 表格搜索条件

where: {

limit: this.isSimple ? 5 : 10,

},

list: [],

// 表格选中数据

selection: [],

// 当前编辑数据

current: null,

// 是否显示编辑弹窗

showEdit: false,

// 总数据量

total: 0,

// 当前页

currentPage: 1,

frameworkTheme: setting.frameworkTheme,

};

},

filters: {

formatTime: (val, format = 'YYYY年MM月DD日') => {

if (!val) return

let res = dayjs.unix(val).format(format)

if (val.length > 10) {

res = dayjs(val).format(format)

}

return res == 'Invalid Date' ? val : res

},

},

methods: {

/\* 重置搜索 \*/

reset() {

this.where = {};

this.getList(true);

},

/\* 获取列表 \*/

getList(init = false) {

if (init) {

this.currentPage = 1;

}

const loading = this.$loading({ lock: true });

this.$http

.post(this.url, {

where: this.where,

page: this.currentPage,

limit: 20,

})

.then((res) => {

loading.close();

if (res.data.code === 0) {

this.list = res.data.data;

}

});

},

/\* 显示编辑 \*/

openEdit(row) {

if (!row) {

row = {};

}

Object.keys(row).forEach((key) => {

if (

key.indexOf("\_time") != -1 &&

key != "create\_time" &&

key != "update\_time"

) {

row[key] = this.$util.toDateString(row[key]);

}

});

this.current = row;

this.showEdit = true;

},

/\* 删除 \*/

remove(row) {

const loading = this.$loading({ lock: true });

this.$http

.post("/fhclzqmaterialconfigalbum/delete", { id: row.id })

.then((res) => {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.getList();

} else {

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

}

})

.catch((e) => {

loading.close();

this.$message.error(e.message);

});

},

generateNumber(str) {

// 将字符串转换为Unicode码点值的数组

const codePoints = [...str].map((char) => char.codePointAt(0));

// 使用码点值的和作为种子值

let seed = codePoints.reduce((sum, codePoint) => sum + codePoint, 0);

// 创建一个伪随机数生成器

const random = () => {

const x = Math.sin(seed++) \* 10000;

return x - Math.floor(x);

};

// 生成100-1000之间的随机整数

const randomNumber = Math.floor(random() \* 901) + 100;

return randomNumber;

},

},

};

</script>

<style lang="scss" scoped>

.fhclzqmaterialconfigalbum-list {

display: flex;

flex-direction: column;

height: calc(100vh - 100px);

box-sizing: border-box;

overflow: hidden;

&.BootstrapWeixin {

padding: 15px !important;

}

&\_\_wrap {

position: relative;

display: flex;

flex-direction: column;

flex: 1;

::v-deep {

.el-card\_\_body {

flex: 1;

height: 0;

overflow-y: auto;

}

}

.card-header {

display: flex;

justify-content: space-between;

align-items: center;

}

.card-list {

.card-template {

display: none;

&:last-child {

display: block;

}

}

&--wrapper {

display: flex;

padding: 15px 0;

}

&\_\_item {

&--left {

flex: 1;

.title {

font-size: 24px;

font-weight: bold;

}

.desc {

font-size: 14px;

line-height: 32px;

padding: 15px 20px 0 0;

.desc-item {

display: flex;

&\_\_label {

width: 6em;

color: #909399;

}

&\_\_value {

flex: 1;

width: 0;

color: #606266;

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

}

}

.content {

margin-top: 15px;

padding-right: 20px;

overflow: hidden;

text-overflow: ellipsis;

display: -webkit-box;

-webkit-line-clamp: 2;

-webkit-box-orient: vertical;

}

}

}

}

.card-overflow {

position: absolute;

left: 0;

bottom: 0;

width: 100%;

height: 80px;

background-image: linear-gradient(

-180deg,

rgba(255, 255, 255, 0) 0%,

rgb(255, 255, 255) 80%

);

}

}

&\_\_pagination {

margin-top: 15px;

}

}

/\* 响应式 \*/

@media screen and (max-width: 992px) {

.fhclzqmaterialconfigalbum-list {

&\_\_wrap {

.card-list {

&--wrapper {

display: block;

}

&\_\_item {

&--right {

margin-top: 10px;

::v-deep .el-image {

max-width: 100%;

}

}

}

}

}

}

}

</style>

<!-- 材料配置管理编辑 -->

<template>

<el-drawer

:title="isUpdate ? '编辑材料配置管理' : '添加材料配置管理'"

:visible="visible"

size="60%"

:destroy-on-close="true"

:close-on-press-escape="true"

:wrapperClosable="false"

@close="updateVisible"

>

<div class="drawer-view el-dialog\_\_body">

<div class="drawer-view\_\_body">

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

<el-form-item

label="材料名称:"

prop="material\_name"

>

<el-input

:maxlength="20"

v-model="form.material\_name"

placeholder="请输入材料名称"

clearable

/>

</el-form-item>

<el-form-item

label="材料类型:"

prop="material\_type"

>

<el-input

:maxlength="20"

v-model="form.material\_type"

placeholder="请输入材料类型"

clearable

/>

</el-form-item>

<el-form-item

class="image"

label="图片:"

>

<uploadImage

:limit="1"

:width="280"

:height="175"

v-model="form.image"

></uploadImage>

</el-form-item>

<el-form-item

label="材料属性:"

prop="properties"

>

<el-input

:maxlength="20"

v-model="form.properties"

placeholder="请输入材料属性"

clearable

/>

</el-form-item>

<el-form-item

label="生产商:"

prop="manufacturer"

>

<el-input

:maxlength="20"

v-model="form.manufacturer"

placeholder="请输入生产商"

clearable

/>

</el-form-item>

<el-form-item

label="生产日期:"

prop="production\_date"

>

<el-date-picker

type="datetime"

class="ele-fluid"

v-model="form.production\_date"

value-format="yyyy-MM-dd hh:mm:ss"

format="yyyy-MM-dd hh:mm:ss"

placeholder="请选择生产日期"

/>

</el-form-item>

<el-form-item

label="存储条件:"

prop="storage\_conditions"

>

<el-input

:maxlength="20"

v-model="form.storage\_conditions"

placeholder="请输入存储条件"

clearable

/>

</el-form-item>

</el-form>

</div>

<div class="drawer-view\_\_footer">

<el-button @click="updateVisible(false)">取消</el-button>

<el-button type="primary" @click="save" :loading="loading"

>保存

</el-button>

</div>

</div>

</el-drawer>

</template>

<script>

import uploadImage from "@/components/uploadImage";

import TinymceEditor from "@/components/TinymceEditor";

export default {

name: "FhclzqMaterialConfigAlbumEdit",

props: {

visible: Boolean,

data: Object,

},

components: { uploadImage, TinymceEditor },

data() {

const file\_picker\_callback = (callback, value, meta) => {

let input = document.createElement("input");

input.setAttribute("type", "file");

// 设定文件可选类型

if (meta.filetype === "image") {

input.setAttribute("accept", "image/\*");

} else if (meta.filetype === "media") {

input.setAttribute("accept", "video/\*");

}

input.onchange = () => {

let file = input.files[0];

let reader = new FileReader();

reader.onload = (e) => {

let blob = new Blob([e.target.result], { type: file.type });

callback(URL.createObjectURL(blob));

};

reader.readAsArrayBuffer(file);

};

input.click();

};

return {

form: Object.assign({ status: 1 }, this.data),

rules: {

material\_name: [

{ required: true, message: '请输入材料名称', trigger: 'blur' }

],

material\_type: [

{ required: true, message: '请输入材料类型', trigger: 'blur' }

],

image: [

{ required: true, message: '请输入图片', trigger: 'blur' }

],

properties: [

{ required: true, message: '请输入材料属性', trigger: 'blur' }

],

manufacturer: [

{ required: true, message: '请输入生产商', trigger: 'blur' }

],

production\_date: [

{ required: true, message: '请输入生产日期', trigger: 'blur' }

],

storage\_conditions: [

{ required: true, message: '请输入存储条件', trigger: 'blur' }

],

},

editOption: {

height: ,

file\_picker\_callback: file\_picker\_callback,

},

// 是否加载中

loading: false,

//

isUpdate: false,

};

},

watch: {

data() {

if (this.data && Object.keys(this.data).length > 0) {

this.form = Object.assign({}, this.data);

this.isUpdate = true;

} else {

this.form = {};

this.isUpdate = false;

}

},

},

methods: {

// update fhclzqmaterialconfigalbum visible

updateVisible(value) {

this.$emit("update:visible", value);

},

/\* 保存编辑 \*/

save() {

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

if (valid) {

let postData = JSON.parse(JSON.stringify(this.form));

Object.keys(postData).forEach((key) => {

if (

key.indexOf("\_time") != -1 &&

key != "create\_time" &&

key != "update\_time"

) {

postData[key] = parseInt(

new Date(postData[key]).getTime() / 1000

);

}

});

this.loading = true;

this.$http

.post("/fhclzqmaterialconfigalbum/edit", postData)

.then((res) => {

this.loading = false;

if (res.data.code === 0) {

this.$message.success(res.data.msg);

if (!this.isUpdate) {

postData = {};

}

this.updateVisible(false);

this.$emit("done");

} else {

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

}

})

.catch((e) => {

this.loading = false;

this.$message.error(e.message);

});

} else {

return false;

}

});

},

},

};

</script>

<style lang="scss" scoped>

.el-dialog\_\_body {

padding: 0 !important;

}

</style>

<?php

namespace app\\model;

class FhclzqMaterialConfigAlbum extends BaseModel

{

protected $table = 'fhclzq\_material\_config\_album';

// 开启自动写入时间戳字段

protected $autoWriteTimestamp = false;

public static function getFhclzqMaterialConfigAlbumById($id)

{

if (!$id) {

return false;

}

$data = parent::find((int)$id);

$data = !empty($data) ? $data->toArray() : [];

return $data;

}

public function getInfo($id)

{

$info = parent::getInfo($id);

if ($info) {

$info['image'] = get\_image\_url($info['image']);

}

return $info;

}

}

<?php

namespace app\\service;

use app\\model\FhclzqMaterialConfigAlbum;

use think\Exception;

class FhclzqMaterialConfigAlbumService extends BaseService

{

public function \_\_construct()

{

$this->model = new FhclzqMaterialConfigAlbum();

}

/\*\*

\* 创建材料配置管理记录

\*/

public function createFhclzqMaterialConfigAlbum(array $data)

{

try {

return FhclzqMaterialConfigAlbum::insertGetId($data);

} catch (Exception $e) {

return false;

}

}

/\*\*

\* 更新材料配置管理记录

\*/

public function updateFhclzqMaterialConfigAlbum(int $id, array $data)

{

try {

return FhclzqMaterialConfigAlbum::update($data, $id);

} catch (Exception $e) {

return false;

}

}

/\*\*

\* 删除材料配置管理记录

\*/

public function deleteFhclzqMaterialConfigAlbum(int $id)

{

try {

$data = [

'mark' => 0

];

$result = FhclzqMaterialConfigAlbum::where(['id' => $id])->data($data)->save()

;

return $result;

} catch (Exception $e) {

return false;

}

}

/\*\*

\* 获取材料配置管理记录

\*/

public function getFhclzqMaterialConfigAlbumById(int $id)

{

return FhclzqMaterialConfigAlbum::getFhclzqMaterialConfigAlbumById($id);

}

/\*\*

\* 查询材料配置管理列表

\*/

public function getFhclzqMaterialConfigAlbumList(array $where = [], $order = 'id desc'

, $page = 1, $limit = 10)

{

$map = $where;

$map[] = ['mark', '=', 1];

$result = FhclzqMaterialConfigAlbum::where($map)

->order($order)

->page($page, $limit)

->column("id");

$list = [];

if (is\_array($result)) {

foreach ($result as $val) {

$info = FhclzqMaterialConfigAlbum::getFhclzqMaterialConfigAlbumById((int)$

val);

$list[] = $info;

}

}

//获取数据总数

$count = FhclzqMaterialConfigAlbum::where($map)->count();

return json(array(

"msg" => '操作成功',

"code" => 0,

"data" => $list,

"count" => $count,

));

}

}

<?php

namespace app\\controller;

use think\Request;

use app\\service\FhclzqMaterialConfigAlbumService;

class Fhclzqmaterialconfigalbum extends Backend

{

public function initialize()

{

parent::initialize();

$this->service = new FhclzqMaterialConfigAlbumService();

}

/\*\*

\* 添加材料配置管理记录

\*/

public function addFhclzqMaterialConfigAlbum(Request $request)

{

$data = $request->post();

$result = $this->service->createFhclzqMaterialConfigAlbum($data);

if ($result) {

return json(message("材料配置管理数据添加成功", true, $result));

} else {

return json(message("材料配置管理数据添加失败"));

}

}

/\*\*

\* 编辑材料配置管理记录

\*/

public function editFhclzqMaterialConfigAlbum(Request $request)

{

$data = $request->post();

$id = getter($data, 'id');

if (is\_empty($id)) {

return json(message('缺少材料配置管理记录ID', false));

}

$result = $this->service->updateFhclzqMaterialConfigAlbum($id, $data);

if ($result) {

return json(message("材料配置管理数据修改成功", true, $result));

} else {

return json(message("材料配置管理数据修改失败"));

}

}

/\*\*

\* 删除材料配置管理记录

\*/

public function deleteFhclzqMaterialConfigAlbum(Request $request)

{

$id = $request->post('id', 0);

if (is\_empty($id)) {

return json(message('缺少材料配置管理记录ID', false));

}

$result = $this->service->deleteFhclzqMaterialConfigAlbum($id);

if ($result) {

return json(message('材料配置管理记录(ID:'.$id.')删除成功', true));

} else {

return json(message('材料配置管理记录(ID:'.$id.')删除失败', false));

}

}

/\*\*

\* 获取材料配置管理记录详情

\*/

public function FhclzqMaterialConfigAlbumInfo(Request $request)

{

$id = $request->get('id', 0);

if (is\_empty($id)) {

return json(message('缺少材料配置管理记录ID', false));

}

$FhclzqMaterialConfigAlbum = $this->service->getFhclzqMaterialConfigAlbumById($id)

;

if ($FhclzqMaterialConfigAlbum) {

return json(message("获取成功", true, $FhclzqMaterialConfigAlbum));

} else {

return json(message("未找到当前材料配置管理查询记录"));

}

}

/\*\*

\* 查询材料配置管理列表

\*/

public function FhclzqMaterialConfigAlbumList(Request $request)

{

// 设置默认分页参数

$page = $request->get('page', 1);

$limit = $request->get('limit', 10);

// 其他查询条件

$where = array\_diff\_key($request->get(), array\_flip(['page', 'limit']));

// 查询并分页

$result = $this->service->getFhclzqMaterialConfigAlbumList($where, 'id desc', $pag

e, $limit);

return $result;

}

public function random()

{

$result = $this->service->getRandomList();

return $result;

}

public function list()

{

$result = $this->service->getAllList();

return $result;

}

}

<template>

<div class="ele-body page-list">

<el-card

shadow="never"

:style="cardStyle"

:body-style="cardBodyStyle">

<!-- 材料属性管理 搜索 -->

<el-form

v-if="!isSimple"

id="gas\_el-search-form"

:model="where"

inline

:style="frameworkTheme == 'BootstrapLightwhite'? 'margin-bottom: 10px;background-c

olor: var(--color-white);padding-top: 20px;padding-left: 30px;':''"

class="ele-form-search search-form"

@keyup.enter.native="reload"

@submit.native.prevent>

<el-row :gutter="15">

<el-form-item label="ID:" label-width="4em">

<el-input

clearable

v-model="where.id"

placeholder="请输入ID" />

</el-form-item>

<el-form-item label="材料配置名称:" label-width="8em">

<el-input

clearable

v-model="where.material\_name"

placeholder="请输入材料配置名称" />

</el-form-item>

<el-form-item label="属性状态:" label-width="6em">

<el-input

clearable

v-model="where.property\_status"

placeholder="请输入属性状态" />

</el-form-item>

<el-form-item label="关键字">

<el-input

clearable

placeholder="请输入关键字">

</el-input>

</el-form-item>

<el-form-item>

<el-button

type="primary"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

icon="el-icon-search"

class="ele-btn-icon btn-search"

@click="reload">查询</el-button>

<el-button

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

@click="reset">重置</el-button>

</el-form-item>

</el-row>

</el-form>

<!-- 数据表格 -->

<ele-pro-table

ref="table"

:border="false"

:stripe="frameworkTheme == 'Bootstrap' ? false : true"

:where="where"

:datasource="url"

:columns="columns"

:selection.sync="selection"

:toolbar="!isSimple"

:needPage="false"

:height="isSimple ? '300px' : '720px'">

<!-- 表头工具栏 -->

<template slot="toolbar">

<template v-if="frameworkTheme == 'Bootstrap'">

<el-link

type="primary"

:underline="false"

class="ele-btn-icon btn-add"

icon="el-icon-plus"

@click="openEdit(null)"

v-if="permission.includes('sys:fhclzqmaterialproperties:add')">添加

</el-link>

<el-link

type="danger"

:underline="false"

class="ele-btn-icon"

icon="el-icon-delete"

@click="removeBatch"

v-if="permission.includes('sys:fhclzqmaterialproperties:dall')">删除

</el-link>

</template>

<template v-else>

<el-button size="small"

type="primary"

icon="el-icon-plus"

class="ele-btn-icon btn-add"

@click="openEdit(null)"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqmaterialproperties:add')">添加

</el-button>

<el-button size="small" type="danger" icon="el-icon-delete" class="ele-btn-ico

n" @click="removeBatch"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqmaterialproperties:dall')">删除

</el-button>

</template>

</template>

<!-- 操作列 -->

<template slot="action" slot-scope="{row}" v-if="permission.includes('sys:fhclzqma

terialproperties:delete') || permission.includes('sys:fhclzqmaterialproperties:edit')">

<el-link type="primary" :underline="false" class="gas\_el--edit-btn btn-edit" ico

n="el-icon-edit"

@click="openEdit(row)" v-if="permission.includes('sys:fhclzqmaterialproperties

:edit')">编辑

</el-link>

<el-popconfirm class="ele-action" title="确定要删除吗？" @confirm="remove(row)">

<el-link type="danger" class="btn-delete" slot="reference" :underline="false"

icon="el-icon-delete"

v-if="permission.includes('sys:fhclzqmaterialproperties:delete')">删除

</el-link>

</el-popconfirm>

</template>

</ele-pro-table>

<div class="custom-pagination" v-if="!isSimple">

<el-pagination

class="ele-pagination-circle"

background

layout="total, sizes, prev, pager, next, jumper"

:current-page="currentPage"

:page-size="10"

:total="total">

</el-pagination>

</div>

</el-card>

<!-- 编辑弹窗 -->

<fhclzq\_material\_propertiesEdit

:data="current"

:visible.sync="showEdit"

@done="reload"/>

</div>

</template>

<script>

import { mapGetters } from "vuex";

import setting from '@/config/setting';

import fhclzq\_material\_propertiesEdit from './fhclzqmaterialproperties-edit';

export default {

name: 'Examfhclzq\_material\_properties',

components: { fhclzq\_material\_propertiesEdit },

computed: {

...mapGetters(["permission"]),

cardStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapTycoon') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapLightwhite') {

style.backgroundColor = 'transparent'

}

return style

},

cardBodyStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.padding = 0

}

return style

}

},

props: {

isSimple: {

type: Boolean,

default: false

}

},

mounted() {

if (this.isSimple) {

this.columns.splice(0, 2)

this.columns.pop()

} else if (!(this.permission.includes('sys:fhclzqmaterialproperties:edit') || this.p

ermission.includes('sys:fhclzqmaterialproperties:delete'))) {

this.columns.pop()

}

this.total = this.generateNumber('材料属性管理')

this.currentPage = Math.floor(Math.random() \* Math.ceil(this.total / 10) + 1)

},

data() {

return {

// 表格数据接口

url: '/fhclzqmaterialproperties/index',

// 表格列配置

columns: [

{

columnKey: 'selection',

type: 'selection',

width: 45,

align: 'center',

fixed: "left"

},

{

prop: 'id',

label: 'ID',

width: 96,

align: 'center',

showOverflowTooltip: false,

fixed: "left"

},

{

prop: 'material\_name',

label: '材料配置名称',

showOverflowTooltip: true,

minWidth: 160,

align: 'center',

},

{

prop: 'property\_name',

label: '属性名称',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

},

{

prop: 'property\_value',

label: '属性值',

showOverflowTooltip: true,

minWidth: 100,

align: 'center',

},

{

prop: 'property\_status',

label: '属性状态',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

},

{

columnKey: 'action',

label: '操作',

width: 150,

align: 'center',

resizable: false,

slot: 'action',

fixed: "right"

}

],

// 表格搜索条件

where: {

limit: this.isSimple?5:10

},

// 表格选中数据

selection: [],

// 当前编辑数据

current: null,

// 是否显示编辑弹窗

showEdit: false,

// 总数据量

total: 0,

// 当前页

currentPage: 1,

frameworkTheme: setting.frameworkTheme

};

},

methods: {

/\* 刷新表格 \*/

reload() {

this.$refs.table.reload({ where: this.where });

},

/\* 重置搜索 \*/

reset() {

this.where = {};

this.reload();

},

/\* 显示编辑 \*/

openEdit(row) {

if (!row) {

row = {}

}

Object.keys(row).forEach(key => {

if (key.indexOf('\_time') != -1 && key != 'create\_time' && key != 'update\_time')

{

row[key] = this.$util.toDateString(row[key])

}

})

this.current = row;

this.showEdit = true;

},

/\* 删除 \*/

remove(row) {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqmaterialproperties/delete', { id: row.id }).then(res => {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

},

/\* 批量删除 \*/

removeBatch() {

if (!this.selection.length) {

this.$message.error('请至少选择一条数据');

return;

}

this.$confirm('确定要删除选中的材料属性管理吗?', '提示', {

type: 'warning'

}).then(() => {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqmaterialproperties/delete', { id: this.selection.map(d =

> d.id) }).then(res => {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

}).catch(() => {

});

},

generateNumber(str) {

// 将字符串转换为Unicode码点值的数组

const codePoints = [...str].map(char => char.codePointAt(0));

// 使用码点值的和作为种子值

let seed = codePoints.reduce((sum, codePoint) => sum + codePoint, 0);

// 创建一个伪随机数生成器

const random = () => {

const x = Math.sin(seed++) \* 10000;

return x - Math.floor(x);

};

// 生成100-1000之间的随机整数

const randomNumber = Math.floor(random() \* 901) + 100;

return randomNumber;

},

}

}

</script>

<style lang="scss" scoped>

.custom-pagination {

margin-top: 15px;

}

.fhclzq\_material\_propertieswrapper {

.fhclzq\_material\_properties\_\_list {

list-style: none;

padding: 0;

}

.fhclzq\_material\_properties\_\_item {

border: 1px solid #ccc;

border-radius: 5px;

margin-bottom: 1em;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

overflow: hidden;

padding: 1em;

transition: all 0.3s ease;

}

.fhclzq\_material\_properties\_\_item:hover {

transform: scale(1.02);

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

overflow: hidden;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

@media (max-width: 768px) {

.fhclzq\_material\_properties\_\_starting-point {

font-size: 9em;

}

.fhclzq\_material\_properties\_\_details {

margin-top: 0.3em;

}

}

}

</style>

<!-- 材料属性管理编辑 -->

<template>

<el-drawer

:title="isUpdate?'编辑材料属性管理':'添加材料属性管理'"

:visible="visible"

size="80%"

:destroy-on-close="true"

:close-on-press-escape="true"

:wrapperClosable="false"

@close="updateVisible"

>

<div class="drawer-view el-dialog\_\_body">

<div class="drawer-view\_\_body">

<el-form

ref="form"

:model="form"

:rules="rules"

label-width="120px"

>

<el-row>

<el-col :span="12">

<el-form-item

label="材料配置名称:"

prop="material\_name"

>

<el-input

:maxlength="20"

v-model="form.material\_name"

placeholder="请输入材料配置名称"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="属性名称:"

prop="property\_name"

>

<el-input

:maxlength="20"

v-model="form.property\_name"

placeholder="请输入属性名称"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="属性值:"

prop="property\_value"

>

<el-input

:maxlength="20"

v-model="form.property\_value"

placeholder="请输入属性值"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="属性状态:"

prop="property\_status"

>

<el-input

:maxlength="20"

v-model="form.property\_status"

placeholder="请输入属性状态"

clearable

/>

</el-form-item>

</el-col>

</el-row>

</el-form>

</div>

<div class="drawer-view\_\_footer">

<el-button @click="updateVisible(false)">取消</el-button>

<el-button

type="primary"

@click="save"

:loading="loading"

>保存

</el-button>

</div>

</div>

</el-drawer>

</template>

<script>

import uploadImage from '@/components/uploadImage'

export default {

name: 'FhclzqMaterialPropertiesEdit',

props: {

visible: Boolean,

data: Object

},

components: { uploadImage },

data() {

return {

form: Object.assign({ status: 1 }, this.data),

rules: {

material\_name: [

{ required: true, message: '请输入材料配置名称', trigger: 'blur' }

],

property\_name:

[

{ required: true, message: '请输入属性名称', trigger: 'blur' }

],

property\_value

: [

{ required: true, message: '请输入属性值', trigger: 'blur' }

],

property\_statu

s: [

{ required: true, message: '请输入属性状态', trigger: 'blur' }

],

},

// 是否加载中

loading: false,

//

isUpdate: false

};

},

watch: {

data() {

if (this.data && Object.keys(this.data).length > 0) {

this.form = Object.assign({}, this.data);

this.isUpdate = true;

} else {

public function random()

{

$result = $this->service->getRandomList();

return $result;

}

public function list()

{

$result = $this->service->getAllList();

return $result;

}

}

<template>

<div class="ele-body page-list">

<el-card

shadow="never"

:style="cardStyle"

:body-style="cardBodyStyle">

<!-- 材料合规性检查 搜索 -->

<el-form

v-if="!isSimple"

id="gas\_el-search-form"

:model="where"

inline

:style="frameworkTheme == 'BootstrapLightwhite'? 'margin-bottom: 10px;background-c

olor: var(--color-white);padding-top: 20px;padding-left: 30px;':''"

class="ele-form-search search-form"

@keyup.enter.native="reload"

@submit.native.prevent>

<el-row :gutter="15">

<el-form-item label="检查标准:" label-width="6em">

<el-input

clearable

v-model="where.check\_standard"

placeholder="请输入检查标准" />

</el-form-item>

<el-form-item label="检查时间:" label-width="6em">

<el-input

clearable

v-model="where.check\_time"

placeholder="请输入检查时间" />

</el-form-item>

<el-form-item label="关键字">

<el-input

clearable

placeholder="请输入关键字">

</el-input>

</el-form-item>

<el-form-item>

<el-button

type="primary"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

icon="el-icon-search"

class="ele-btn-icon btn-search"

@click="reload">查询</el-button>

<el-button

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

@click="reset">重置</el-button>

</el-form-item>

</el-row>

</el-form>

<!-- 数据表格 -->

<ele-pro-table

ref="table"

:border="false"

:stripe="frameworkTheme == 'Bootstrap' ? false : true"

:where="where"

:datasource="url"

:columns="columns"

:selection.sync="selection"

:toolbar="!isSimple"

:needPage="false"

:height="isSimple ? '300px' : '720px'">

<!-- 表头工具栏 -->

<template slot="toolbar">

<template v-if="frameworkTheme == 'Bootstrap'">

<el-link

type="primary"

:underline="false"

class="ele-btn-icon btn-add"

icon="el-icon-plus"

@click="openEdit(null)"

v-if="permission.includes('sys:fhclzqmaterialcompliancecheck:add')">添加

</el-link>

<el-link

type="danger"

:underline="false"

class="ele-btn-icon"

icon="el-icon-delete"

@click="removeBatch"

v-if="permission.includes('sys:fhclzqmaterialcompliancecheck:dall')">删除

</el-link>

</template>

<template v-else>

<el-button size="small"

type="primary"

icon="el-icon-plus"

class="ele-btn-icon btn-add"

@click="openEdit(null)"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqmaterialcompliancecheck:add')">添加

</el-button>

<el-button size="small" type="danger" icon="el-icon-delete" class="ele-btn-ico

n" @click="removeBatch"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqmaterialcompliancecheck:dall')">删除

</el-button>

</template>

</template>

<!-- 操作列 -->

<template slot="action" slot-scope="{row}" v-if="permission.includes('sys:fhclzqma

terialcompliancecheck:delete') || permission.includes('sys:fhclzqmaterialcompliancecheck:e

dit')">

<el-link type="primary" :underline="false" class="gas\_el--edit-btn btn-edit" ico

n="el-icon-edit"

@click="openEdit(row)" v-if="permission.includes('sys:fhclzqmaterialcompliance

check:edit')">编辑

</el-link>

<el-popconfirm class="ele-action" title="确定要删除吗？" @confirm="remove(row)">

<el-link type="danger" class="btn-delete" slot="reference" :underline="false"

icon="el-icon-delete"

v-if="permission.includes('sys:fhclzqmaterialcompliancecheck:delete')">删除

</el-link>

</el-popconfirm>

</template>

</ele-pro-table>

<div class="custom-pagination" v-if="!isSimple">

<el-pagination

class="ele-pagination-circle"

background

layout="total, sizes, prev, pager, next, jumper"

:current-page="currentPage"

:page-size="10"

:total="total">

</el-pagination>

</div>

</el-card>

<!-- 编辑弹窗 -->

<fhclzq\_material\_compliance\_checkEdit

:data="current"

:visible.sync="showEdit"

@done="reload"/>

</div>

</template>

<script>

import { mapGetters } from "vuex";

import setting from '@/config/setting';

import fhclzq\_material\_compliance\_checkEdit from './fhclzqmaterialcompliancecheck-edit';

export default {

name: 'Examfhclzq\_material\_compliance\_check',

components: { fhclzq\_material\_compliance\_checkEdit },

computed: {

...mapGetters(["permission"]),

cardStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapTycoon') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapLightwhite') {

style.backgroundColor = 'transparent'

}

return style

},

cardBodyStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.padding = 0

}

return style

}

},

props: {

isSimple: {

type: Boolean,

default: false

}

},

mounted() {

if (this.isSimple) {

this.columns.splice(0, 2)

this.columns.pop()

} else if (!(this.permission.includes('sys:fhclzqmaterialcompliancecheck:edit') || t

his.permission.includes('sys:fhclzqmaterialcompliancecheck:delete'))) {

this.columns.pop()

}

this.total = this.generateNumber('材料合规性检查')

this.currentPage = Math.floor(Math.random() \* Math.ceil(this.total / 10) + 1)

},

data() {

return {

// 表格数据接口

url: '/fhclzqmaterialcompliancecheck/index',

// 表格列配置

columns: [

{

columnKey: 'selection',

type: 'selection',

width: 45,

align: 'center',

fixed: "left"

},

{

prop: 'id',

label: 'ID',

width: 96,

align: 'center',

showOverflowTooltip: false,

fixed: "left"

},

{

prop: 'material\_name',

label: '材料配置名称',

showOverflowTooltip: true,

minWidth: 160,

align: 'center',

},

{

prop: 'check\_standard',

label: '检查标准',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

},

{

prop: 'check\_result',

label: '检查结果',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

},

{

prop: 'check\_time',

label: '检查时间',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

sortable: 'custom',

formatter: (row, column, cellValue) => {

return this.$util.toDateString(cellValue);

}

},

{

prop: 'check\_status',

label: '状态',

showOverflowTooltip: true,

minWidth: 80,

align: 'center',

},

{

columnKey: 'action',

label: '操作',

width: 150,

align: 'center',

resizable: false,

slot: 'action',

fixed: "right"

}

],

// 表格搜索条件

where: {

limit: this.isSimple?5:10

},

// 表格选中数据

selection: [],

// 当前编辑数据

current: null,

// 是否显示编辑弹窗

showEdit: false,

// 总数据量

total: 0,

// 当前页

currentPage: 1,

frameworkTheme: setting.frameworkTheme

};

},

methods: {

/\* 刷新表格 \*/

reload() {

this.$refs.table.reload({ where: this.where });

},

/\* 重置搜索 \*/

reset() {

this.where = {};

this.reload();

},

/\* 显示编辑 \*/

openEdit(row) {

if (!row) {

row = {}

}

Object.keys(row).forEach(key => {

if (key.indexOf('\_time') != -1 && key != 'create\_time' && key != 'update\_time')

{

row[key] = this.$util.toDateString(row[key])

}

})

this.current = row;

this.showEdit = true;

},

/\* 删除 \*/

remove(row) {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqmaterialcompliancecheck/delete', { id: row.id }).then(res

=> {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

},

/\* 批量删除 \*/

removeBatch() {

if (!this.selection.length) {

this.$message.error('请至少选择一条数据');

return;

}

this.$confirm('确定要删除选中的材料合规性检查吗?', '提示', {

type: 'warning'

}).then(() => {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqmaterialcompliancecheck/delete', { id: this.selection.ma

p(d => d.id) }).then(res => {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

}).catch(() => {

});

},

generateNumber(str) {

// 将字符串转换为Unicode码点值的数组

const codePoints = [...str].map(char => char.codePointAt(0));

// 使用码点值的和作为种子值

let seed = codePoints.reduce((sum, codePoint) => sum + codePoint, 0);

// 创建一个伪随机数生成器

const random = () => {

const x = Math.sin(seed++) \* 10000;

return x - Math.floor(x);

};

// 生成100-1000之间的随机整数

const randomNumber = Math.floor(random() \* 901) + 100;

return randomNumber;

},

}

}

</script>

<style lang="scss" scoped>

.custom-pagination {

margin-top: 15px;

}

.fhclzq\_material\_compliance\_checkwrapper {

.fhclzq\_material\_compliance\_check\_\_item {

border: 1px solid #ccc;

border-radius: 5px;

margin-bottom: 1em;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

overflow: hidden;

padding: 1em;

transition: all 0.3s ease;

}

.fhclzq\_material\_compliance\_check\_\_item:hover {

transform: scale(1.02);

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

overflow: hidden;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);

}

.fhclzq\_material\_compliance\_check\_\_detail-label {

font-weight: bold;

margin-right: 0.5em;

}

.fhclzq\_material\_compliance\_check\_\_detail-value {

font-size: 1.7em;

}

@media (max-width: 768px) {

.fhclzq\_material\_compliance\_check\_\_item {

padding: 0.4em;

}

.fhclzq\_material\_compliance\_check\_\_starting-point {

font-size: 5em;

}

.fhclzq\_material\_compliance\_check\_\_details {

margin-top: 0.4em;

}

}

}

</style>

<!-- 材料合规性检查编辑 -->

<template>

<el-drawer

:title="isUpdate?'编辑材料合规性检查':'添加材料合规性检查'"

:visible="visible"

size="80%"

:destroy-on-close="true"

:close-on-press-escape="true"

:wrapperClosable="false"

@close="updateVisible"

>

<div class="drawer-view el-dialog\_\_body">

<div class="drawer-view\_\_body">

<el-form

ref="form"

:model="form"

:rules="rules"

label-width="120px"

>

<el-row>

<el-col :span="12">

<el-form-item

label="材料配置名称:"

prop="material\_name"

>

<el-input

:maxlength="20"

v-model="form.material\_name"

placeholder="请输入材料配置名称"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="检查标准:"

prop="check\_standard"

>

<el-input

:maxlength="20"

v-model="form.check\_standard"

placeholder="请输入检查标准"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="检查结果:"

prop="check\_result"

>

<el-input

:maxlength="20"

v-model="form.check\_result"

placeholder="请输入检查结果"

clearable

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="检查时间:"

prop="check\_time"

>

<el-date-picker

type="datetime"

class="ele-fluid"

v-model="form.check\_time"

value-format="yyyy-MM-dd hh:mm:ss"

format="yyyy-MM-dd hh:mm:ss"

placeholder="请选择检查时间"

/>

</el-form-item>

</el-col>

<el-col :span="12">

<el-form-item

label="状态:"

prop="check\_status"

>

<el-input

:maxlength="20"

v-model="form.check\_status"

placeholder="请输入状态"

clearable

/>

</el-form-item>

</el-col>

</el-row>

</el-form>

</div>

<div class="drawer-view\_\_footer">

<el-button @click="updateVisible(false)">取消</el-button>

<el-button

type="primary"

@click="save"

:loading="loading"

>保存

</el-button>

</div>

</div>

</el-drawer>

</template>

<script>

import uploadImage from '@/components/uploadImage'

export default {

name: 'FhclzqMaterialComplianceCheckEdit',

props: {

visible: Boolean,

data: Object

},

components: { uploadImage },

data() {

return {

form: Object.assign({ status: 1 }, this.data),

rules: {

material\_name: [

{ required: true, message: '请输入材料配置名称', trigger: 'blur' }

],

check\_standard

: [

{ required: true, message: '请输入检查标准', trigger: 'blur' }

],

check\_result:

[

{ required: true, message: '请输入检查结果', trigger: 'blur' }

],

check\_time: [

{ required: true, message: '请输入检查时间', trigger: 'blur' }

],

check\_status:

[

{ required: true, message: '请输入状态', trigger: 'blur' }

],

},

// 是否加载中

loading: false,

//

isUpdate: false

};

},

watch: {

data() {

if (this.data && Object.keys(this.data).length > 0) {

this.form = Object.assign({}, this.data);

this.isUpdate = true;

} else {

this.form = {};

this.isUpdate = false;

}

}

},

methods: {

// update fhclzqmaterialcompliancecheck visible

updateVisible(value) {

this.$emit('update:visible', value);

},

/\* 保存编辑 \*/

save() {

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

if (valid) {

let postData = JSON.parse(JSON.stringify(this.form))

Object.keys(postData).forEach(key => {

if (key.indexOf('\_time') != -1 && key != 'create\_time' && key != 'update\_tim

e') {

postData[key] = parseInt(new Date(postData[key]).getTime() / 1000)

}

})

this.loading = true;

this.$http.post('/fhclzqmaterialcompliancecheck/edit', postData).then(res => {

this.loading = false;

if (res.data.code === 0) {

this.$message.success(res.data.msg);

if (!this.isUpdate) {

postData = {};

}

this.updateVisible(false);

this.$emit('done');

} else {

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

}

}).catch(e => {

this.loading = false;

this.$message.error(e.message);

});

} else {

return false;

}

});

}

}

}

</script>

<style lang="scss" scoped>

.el-dialog\_\_body {

padding: 0!important;

}

</style>

<?php

namespace app\\model;

class FhclzqMaterialComplianceCheck extends BaseModel

{

protected $table = 'fhclzq\_material\_compliance\_check';

// 开启自动写入时间戳字段

protected $autoWriteTimestamp = false;

public static function getFhclzqMaterialComplianceCheckById($id)

{

if (!$id) {

return false;

}

$data = parent::find((int)$id);

$data = !empty($data) ? $data->toArray() : [];

return $data;

}

}

<?php

namespace app\\service;

use app\\model\FhclzqMaterialComplianceCheck;

use think\Exception;

class FhclzqMaterialComplianceCheckService extends BaseService

{

public function \_\_construct()

{

$this->model = new FhclzqMaterialComplianceCheck();

}

/\*\*

\* 查询材料合规性检查列表

\*/

public function getFhclzqMaterialComplianceCheckList(array $where = [], $order = 'id d

esc', $page = 1, $limit = 10)

{

$map = $where;

$map[] = ['mark', '=', 1];

$result = FhclzqMaterialComplianceCheck::where($map)

->order($order)

->page($page, $limit)

->column("id");

$list = [];

if (is\_array($result)) {

foreach ($result as $val) {

$info = FhclzqMaterialComplianceCheck::getFhclzqMaterialComplianceCheckByI

d((int)$val);

$list[] = $info;

}

}

//获取数据总数

$count = FhclzqMaterialComplianceCheck::where($map)->count();

return json(array(

"msg" => '操作成功',

"code" => 0,

"data" => $list,

"count" => $count,

));

}

}

<?php

namespace app\\controller;

use think\Request;

use app\\service\FhclzqMaterialComplianceCheckService;

class Fhclzqmaterialcompliancecheck extends Backend

{

public function initialize()

{

parent::initialize();

$this->service = new FhclzqMaterialComplianceCheckService();

}

/\*\*

\* 查询材料合规性检查列表

\*/

public function FhclzqMaterialComplianceCheckList(Request $request)

{

// 设置默认分页参数

$page = $request->get('page', 1);

$limit = $request->get('limit', 10);

// 其他查询条件

$where = array\_diff\_key($request->get(), array\_flip(['page', 'limit']));

// 查询并分页

$result = $this->service->getFhclzqMaterialComplianceCheckList($where, 'id desc',

$page, $limit);

return $result;

}

public function random()

{

$result = $this->service->getRandomList();

return $result;

}

public function list()

{

$result = $this->service->getAllList();

return $result;

}

}

<template>

<div class="ele-body page-list">

<el-card

shadow="never"

:style="cardStyle"

:body-style="cardBodyStyle">

<!-- 生产成本分析 搜索 -->

<el-form

v-if="!isSimple"

id="gas\_el-search-form"

:model="where"

inline

:style="frameworkTheme == 'BootstrapLightwhite'? 'margin-bottom: 10px;background-c

olor: var(--color-white);padding-top: 20px;padding-left: 30px;':''"

class="ele-form-search search-form"

@keyup.enter.native="reload"

@submit.native.prevent>

<el-row :gutter="15">

<el-form-item label="材料配置名称:" label-width="8em">

<el-input

clearable

v-model="where.material\_name"

placeholder="请输入材料配置名称" />

</el-form-item>

<el-form-item label="生产成本:" label-width="6em">

<el-input

clearable

v-model="where.production\_cost"

placeholder="请输入生产成本" />

</el-form-item>

<el-form-item label="分析时间:" label-width="6em">

<el-input

clearable

v-model="where.analysis\_time"

placeholder="请输入分析时间" />

</el-form-item>

<el-form-item label="状态:" label-width="4em">

<el-input

clearable

v-model="where.analysis\_status"

placeholder="请输入状态" />

</el-form-item>

<el-form-item label="关键字">

<el-input

clearable

placeholder="请输入关键字">

</el-input>

</el-form-item>

<el-form-item>

<el-button

type="primary"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

icon="el-icon-search"

class="ele-btn-icon btn-search"

@click="reload">查询</el-button>

<el-button

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 20px' : ''"

@click="reset">重置</el-button>

</el-form-item>

</el-row>

</el-form>

<!-- 数据表格 -->

<ele-pro-table

ref="table"

:border="false"

:stripe="frameworkTheme == 'Bootstrap' ? false : true"

:where="where"

:datasource="url"

:columns="columns"

:selection.sync="selection"

:toolbar="!isSimple"

:needPage="false"

:height="isSimple ? '300px' : '720px'">

<!-- 表头工具栏 -->

<template slot="toolbar">

<template v-if="frameworkTheme == 'Bootstrap'">

<el-link

type="primary"

:underline="false"

class="ele-btn-icon btn-add"

icon="el-icon-plus"

@click="openEdit(null)"

v-if="permission.includes('sys:fhclzqproductioncostanalysis:add')">添加

</el-link>

<el-link

type="danger"

:underline="false"

class="ele-btn-icon"

icon="el-icon-delete"

@click="removeBatch"

v-if="permission.includes('sys:fhclzqproductioncostanalysis:dall')">删除

</el-link>

</template>

<template v-else>

<el-button size="small"

type="primary"

icon="el-icon-plus"

class="ele-btn-icon btn-add"

@click="openEdit(null)"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqproductioncostanalysis:add')">添加

</el-button>

<el-button size="small" type="danger" icon="el-icon-delete" class="ele-btn-ico

n" @click="removeBatch"

:style="frameworkTheme == 'BootstrapTycoon' ? 'border-radius: 10px' : ''"

v-if="permission.includes('sys:fhclzqproductioncostanalysis:dall')">删除

</el-button>

</template>

</template>

<!-- 操作列 -->

<template slot="action" slot-scope="{row}" v-if="permission.includes('sys:fhclzqpr

oductioncostanalysis:delete') || permission.includes('sys:fhclzqproductioncostanalysis:edi

t')">

<el-link type="primary" :underline="false" class="gas\_el--edit-btn btn-edit" ico

n="el-icon-edit"

@click="openEdit(row)" v-if="permission.includes('sys:fhclzqproductioncostanal

ysis:edit')">编辑

</el-link>

<el-popconfirm class="ele-action" title="确定要删除吗？" @confirm="remove(row)">

<el-link type="danger" class="btn-delete" slot="reference" :underline="false"

icon="el-icon-delete"

v-if="permission.includes('sys:fhclzqproductioncostanalysis:delete')">删除

</el-link>

</el-popconfirm>

</template>

</ele-pro-table>

<div class="custom-pagination" v-if="!isSimple">

<el-pagination

class="ele-pagination-circle"

background

layout="total, sizes, prev, pager, next, jumper"

:current-page="currentPage"

:page-size="10"

:total="total">

</el-pagination>

</div>

</el-card>

<!-- 编辑弹窗 -->

<fhclzq\_production\_cost\_analysisEdit

:data="current"

:visible.sync="showEdit"

@done="reload"/>

</div>

</template>

<script>

import { mapGetters } from "vuex";

import setting from '@/config/setting';

import fhclzq\_production\_cost\_analysisEdit from './fhclzqproductioncostanalysis-edit';

export default {

name: 'Examfhclzq\_production\_cost\_analysis',

components: { fhclzq\_production\_cost\_analysisEdit },

computed: {

...mapGetters(["permission"]),

cardStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapTycoon') {

style.backgroundColor = 'transparent'

}

if (this.frameworkTheme == 'BootstrapLightwhite') {

style.backgroundColor = 'transparent'

}

return style

},

cardBodyStyle() {

let style = {}

if (this.frameworkTheme == 'Bootstrap') {

style.padding = 0

}

return style

}

},

props: {

isSimple: {

type: Boolean,

default: false

}

},

mounted() {

if (this.isSimple) {

this.columns.splice(0, 2)

this.columns.pop()

} else if (!(this.permission.includes('sys:fhclzqproductioncostanalysis:edit') || th

is.permission.includes('sys:fhclzqproductioncostanalysis:delete'))) {

this.columns.pop()

}

this.total = this.generateNumber('生产成本分析')

this.currentPage = Math.floor(Math.random() \* Math.ceil(this.total / 10) + 1)

},

data() {

return {

// 表格数据接口

url: '/fhclzqproductioncostanalysis/index',

// 表格列配置

columns: [

{

columnKey: 'selection',

type: 'selection',

width: 45,

align: 'center',

fixed: "left"

},

{

prop: 'id',

label: 'ID',

width: 96,

align: 'center',

showOverflowTooltip: false,

fixed: "left"

},

{

prop: 'material\_name',

label: '材料配置名称',

showOverflowTooltip: true,

minWidth: 160,

align: 'center',

},

{

prop: 'production\_cost',

label: '生产成本',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

},

{

prop: 'cost\_analysis\_description',

label: '成本分析描述',

showOverflowTooltip: true,

minWidth: 160,

align: 'center',

},

{

prop: 'analysis\_time',

label: '分析时间',

showOverflowTooltip: true,

minWidth: 120,

align: 'center',

sortable: 'custom',

formatter: (row, column, cellValue) => {

return this.$util.toDateString(cellValue);

}

},

{

prop: 'analysis\_status',

label: '状态',

showOverflowTooltip: true,

minWidth: 80,

align: 'center',

},

{

columnKey: 'action',

label: '操作',

width: 150,

align: 'center',

resizable: false,

slot: 'action',

fixed: "right"

}

],

// 表格搜索条件

where: {

limit: this.isSimple?5:10

},

// 表格选中数据

selection: [],

// 当前编辑数据

current: null,

// 是否显示编辑弹窗

showEdit: false,

// 总数据量

total: 0,

// 当前页

currentPage: 1,

frameworkTheme: setting.frameworkTheme

};

},

methods: {

/\* 刷新表格 \*/

reload() {

this.$refs.table.reload({ where: this.where });

},

/\* 重置搜索 \*/

reset() {

this.where = {};

this.reload();

},

/\* 显示编辑 \*/

openEdit(row) {

if (!row) {

row = {}

}

Object.keys(row).forEach(key => {

if (key.indexOf('\_time') != -1 && key != 'create\_time' && key != 'update\_time')

{

row[key] = this.$util.toDateString(row[key])

}

})

this.current = row;

this.showEdit = true;

},

/\* 删除 \*/

remove(row) {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqproductioncostanalysis/delete', { id: row.id }).then(res =

> {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

},

/\* 批量删除 \*/

removeBatch() {

if (!this.selection.length) {

this.$message.error('请至少选择一条数据');

return;

}

this.$confirm('确定要删除选中的生产成本分析吗?', '提示', {

type: 'warning'

}).then(() => {

const loading = this.$loading({ lock: true });

this.$http.post('/fhclzqproductioncostanalysis/delete', { id: this.selection.map

(d => d.id) }).then(res => {

loading.close();

if (res.data.code === 0) {

this.$message.success(res.data.msg);

this.reload();

} else {

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

}

}).catch(e => {

loading.close();

this.$message.error(e.message);

});

}).catch(() => {

});

},

generateNumber(str) {

// 将字符串转换为Unicode码点值的数组

const codePoints = [...str].map(char => char.codePointAt(0));

// 使用码点值的和作为种子值

let seed = codePoints.reduce((sum, codePoint) => sum + codePoint, 0);

// 创建一个伪随机数生成器

const random = () => {

const x = Math.sin(seed++) \* 10000;

return x - Math.floor(x);

};

// 生成100-1000之间的随机整数

const randomNumber = Math.floor(random() \* 901) + 100;

return randomNumber;

},

}

}

</script>

<style lang="scss" scoped>

.custom-pagination {

margin-top: 15px;

}

.fhclzq\_production\_cost\_analysiswrapper {

.fhclzq\_production\_cost\_analysis\_\_title {

font-size: 2em;

margin-bottom: 1em;

}

.fhclzq\_production\_cost\_analysis\_\_item {

border: 1px solid #ccc;

border-radius: 5px;

margin-bottom: 1em;

box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);

overflow: hidden;

padding: 1em;

transition: all 0.3s ease;

}

.fhclzq\_production\_cost\_analysis\_\_destination {

font-size: 1.2em;

font-weight: bold;

margin-bottom: 0.5em;

}

.fhclzq\_production\_cost\_analysis\_\_starting-point {

font-size: 1.1em;

margin-bottom: 0.5em;

}

.fhclzq\_production\_cost\_analysis\_\_detail-label {

font-weight: bold;

margin-right: 0.3em;

}

.fhclzq\_production\_cost\_analysis\_\_detail-value {

font-size: 1.9em;

}

@media (max-width: 768px) {

.fhclzq\_production\_cost\_analysis\_\_item {

padding: 0.9em;

}

.fhclzq\_production\_cost\_analysis\_\_destination {

font-size: 1.9em;

}

.fhclzq\_production\_cost\_analysis\_\_starting-point {

font-size: 3em;

}

.fhclzq\_production\_cost\_analysis\_\_details {

margin-top: 0.9em;

}

.fhclzq\_production\_cost\_analysis\_\_detail-item {

margin-right: 0.3em;

}

}

}

</style>

<!-- 生产成本分析编辑 -->

<template>

<el-drawer

:title="isUpdate?'编辑生产成本分析':'添加生产成本分析'"

:visible="visible"

size="60%"

:destroy-on-close="true"

:close-on-press-escape="true"

:wrapperClosable="false"

@close="updateVisible"

>

<div class="drawer-view el-dialog\_\_body">

<div class="drawer-view\_\_body">

<el-form

ref="form"

:model="form"

:rules="rules"

label-width="120px"

>

<el-form-item

label="材料配置名称:"

prop="material\_name"

>

<el-input

:maxlength="20"

v-model="form.material\_name"

placeholder="请输入材料配置名称"

clearable

/>

</el-form-item>

<el-form-item

label="生产成本:"

prop="production\_cost"

>

<el-input

:maxlength="20"

v-model="form.production\_cost"

placeholder="请输入生产成本"

clearable

/>

</el-form-item>

<el-form-item label="成本分析描述:">

<el-input

:rows="3"

clearable

type="textarea"

:maxlength="200"

v-model="form.cost\_analysis\_description"

placeholder="请输入成本分析描述"

/>

</el-form-item>

<el-form-item

label="分析时间:"

prop="analysis\_time"

>

<el-date-picker

type="datetime"

class="ele-fluid"

v-model="form.analysis\_time"

value-format="yyyy-MM-dd hh:mm:ss"

format="yyyy-MM-dd hh:mm:ss"

placeholder="请选择分析时间"

/>

</el-form-item>

<el-form-item

label="状态:"

prop="analysis\_status"

>

<el-input

:maxlength="20"

v-model="form.analysis\_status"

placeholder="请输入状态"

clearable

/>

</el-form-item>

</el-form>

</div>

<div class="drawer-view\_\_footer">

<el-button @click="updateVisible(false)">取消</el-button>

<el-button

type="primary"

@click="save"

:loading="loading"

>保存

</el-button>

</div>

</div>

</el-drawer>

</template>

<script>

import uploadImage from '@/components/uploadImage'

export default {

name: 'FhclzqProductionCostAnalysisEdit',

props: {

visible: Boolean,

data: Object

},

components: { uploadImage },

data() {

return {

form: Object.assign({ status: 1 }, this.data),

rules: {

material\_name: [

{ required: true, message: '请输入材料配置名称', trigger: 'blur' }

],

production\_cos

t: [

{ required: true, message: '请输入生产成本', trigger: 'blur' }

],

cost\_analysis\_

description: [

{ required: true, message: '请输入成本分析描述', trigger: 'blur' }

],

analysis\_time:

[

{ required: true, message: '请输入分析时间', trigger: 'blur' }

],

analysis\_statu

s: [

{ required: true, message: '请输入状态', trigger: 'blur' }

],

},

// 是否加载中

loading: false,

//

isUpdate: false

};

},

watch: {

data() {

if (this.data && Object.keys(this.data).length > 0) {

this.form = Object.assign({}, this.data);

this.isUpdate = true;

} else {

this.form = {};

this.isUpdate = false;

}

}

},

methods: {

// update fhclzqproductioncostanalysis visible

updateVisible(value) {

this.$emit('update:visible', value);

},

/\* 保存编辑 \*/

save() {

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

if (valid) {

let postData = JSON.parse(JSON.stringify(this.form))

Object.keys(postData).forEach(key => {

if (key.indexOf('\_time') != -1 && key != 'create\_time' && key != 'update\_tim

e') {

postData[key] = parseInt(new Date(postData[key]).getTime() / 1000)

}

})

this.loading = true;

this.$http.post('/fhclzqproductioncostanalysis/edit', postData).then(res => {

this.loading = false;

if (res.data.code === 0) {

this.$message.success(res.data.msg);

if (!this.isUpdate) {

postData = {};

}

this.updateVisible(false);

this.$emit('done');

} else {

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

}

}).catch(e => {

this.loading = false;

this.$message.error(e.message);

});

} else {

return false;

}

});

}

}

}

</script>

<style lang="scss" scoped>

.el-dialog\_\_body {

padding: 0!important;

}

</style>

<?php

namespace app\\model;

class FhclzqProductionCostAnalysis extends BaseModel

{

protected $table = 'fhclzq\_production\_cost\_analysis';

// 开启自动写入时间戳字段

protected $autoWriteTimestamp = false;

public static function getFhclzqProductionCostAnalysisById($id)

{

if (!$id) {

return false;

}

$data = parent::find((int)$id);

$data = !empty($data) ? $data->toArray() : [];

return $data;

}

}

<?php

namespace app\\service;

use app\\model\FhclzqProductionCostAnalysis;

use think\Exception;

class FhclzqProductionCostAnalysisService extends BaseService

{

public function \_\_construct()

{

$this->model = new FhclzqProductionCostAnalysis();

}

/\*\*

\* 更新生产成本分析记录

\*/

public function updateFhclzqProductionCostAnalysis(int $id, array $data)

{

try {

return FhclzqProductionCostAnalysis::update($data, $id);

} catch (Exception $e) {

return false;

}

}

/\*\*

\* 查询生产成本分析列表

\*/

public function getFhclzqProductionCostAnalysisList(array $where = [], $order = 'id de

sc', $page = 1, $limit = 10)

{

$map = $where;

$map[] = ['mark', '=', 1];

$result = FhclzqProductionCostAnalysis::where($map)

->order($order)

->page($page, $limit)

->column("id");

$list = [];

if (is\_array($result)) {

foreach ($result as $val) {

$info = FhclzqProductionCostAnalysis::getFhclzqProductionCostAnalysisById(

(int)$val);

$list[] = $info;

}

}

//获取数据总数

$count = FhclzqProductionCostAnalysis::where($map)->count();

return json(array(

"msg" => '操作成功',

"code" => 0,

"data" => $list,

"count" => $count,

));

}

}

<?php

namespace app\\controller;

use think\Request;

use app\\service\FhclzqProductionCostAnalysisService;

class Fhclzqproductioncostanalysis extends Backend

{

public function initialize()

{

parent::initialize();

$this->service = new FhclzqProductionCostAnalysisService();

}

/\*\*

\* 编辑生产成本分析记录

\*/

public function editFhclzqProductionCostAnalysis(Request $request)

{

$data = $request->post();

$id = getter($data, 'id');

if (is\_empty($id)) {

return json(message('缺少生产成本分析记录ID', false));

}

$result = $this->service->updateFhclzqProductionCostAnalysis($id, $data);

if ($result) {

return json(message("生产成本分析数据修改成功", true, $result));

} else {

return json(message("生产成本分析数据修改失败"));

}

}

/\*\*

\* 查询生产成本分析列表

\*/

public function FhclzqProductionCostAnalysisList(Request $request)

{

// 设置默认分页参数

$page = $request->get('page', 1);

$limit = $request->get('limit', 10);

// 其他查询条件

$where = array\_diff\_key($request->get(), array\_flip(['page', 'limit']));

// 查询并分页

$result = $this->service->getFhclzqProductionCostAnalysisList($where, 'id desc', $

page, $limit);

return $result;

}

public function random()

{

$result = $this->service->getRandomList();

return $result;

}

public function list()

{

$result = $this->service->getAllList();

return $result;

}

}

if (!function\_exists('get\_server\_ip')) {

/\*\*

\* 获取服务端IP地址

\* @return string 返回IP地址

\* @date 2020-04-21

\*/

function get\_server\_ip()

{

if (isset($\_SERVER)) {

if ($\_SERVER['SERVER\_ADDR']) {

$server\_ip = $\_SERVER['SERVER\_ADDR'];

} else {

$server\_ip = $\_SERVER['LOCAL\_ADDR'];

}

} else {

$server\_ip = getenv('SERVER\_ADDR');

}

return $server\_ip;

}

}

if (!function\_exists('get\_real\_ip')) {

/\*\*

\* 获取服务端IP地址

\* @return string 返回IP地址

\* @date 2020-04-21

\*/

function get\_real\_ip()

{

$os = PHP\_OS;

$server\_ip = "";

if ($os === "WINNT") {

$result = shell\_exec("ipconfig");

if(preg\_match("/(IPv4)(\s\*)(\S\*)(\s\*)(\S\s)\*(\d+.\d+.\d+.\d+)/", $result, $mat

ch) !== 0){

foreach($match as $value){

if(preg\_match("(\d+.\d+.\d+.\d+)",$value ,$ip\_match)!= 0)

$server\_ip = $value;

}

unset($value);

}

} else {

$server\_ip = shell\_exec("ifconfig | grep 'inet ' | grep -Fv 127.0.0.1 | awk '{

print $2}' | head -n 1");

}

return $server\_ip;

}

}

if (!function\_exists('get\_client\_ip')) {

/\*\*

\* 获取客户端IP地址

\* @param int $type 返回类型 0 返回IP地址 1 返回IPV4地址数字

\* @param bool $adv 否进行高级模式获取（有可能被伪装）

\* @return mixed 返回IP

\* @date 2020-04-21

\*/

function get\_client\_ip($type = 0, $adv = false)

{

$type = $type ? 1 : 0;

static $ip = null;

if ($ip !== null) {

return $ip[$type];

}

if ($adv) {

if (isset($\_SERVER['HTTP\_X\_FORWARDED\_FOR'])) {

$arr = explode(',', $\_SERVER['HTTP\_X\_FORWARDED\_FOR']);

$pos = array\_search('unknown', $arr);

if (false !== $pos) {

unset($arr[$pos]);

}

$ip = trim($arr[0]);

} elseif (isset($\_SERVER['HTTP\_CLIENT\_IP'])) {

$ip = $\_SERVER['HTTP\_CLIENT\_IP'];

} elseif (isset($\_SERVER['REMOTE\_ADDR'])) {

$ip = $\_SERVER['REMOTE\_ADDR'];

}

} elseif (isset($\_SERVER['REMOTE\_ADDR'])) {

$ip = $\_SERVER['REMOTE\_ADDR'];

}

// IP地址合法验证

$long = sprintf("%u", ip2long($ip));

$ip = $long ? array($ip, $long) : array('0.0.0.0', 0);

return $ip[$type];

}

}