诚望仓储管理系统源代码

<template>

<div>

<div class="header">

诚望

<span v-for="(item, index) in $route.meta" :key="index">

<i class="el-icon-arrow-right">&nbsp;</i>

{{ item }}

</span>

</div>

</div>

</template>

<script>

export default {

name: "headerTop",

created() {

// console.log(this.$route, "//////");

}

};

</script>

<style lang="scss" scoped>

.header {

height: 50px;

color: #666666;

background: #ffffff;

line-height: 50px;

padding-left: 20px;

}

</style>

<template>

<div>

<div class="list-box"></div>

<div class="list">

<div class="a">

请填写编号:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="number" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="b">

请填写品牌:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="brand" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="c">

请填写型号:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="modle" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="d">

请填写状态:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="status" placeholder="0是以借出,1是在库" class="lista"></el-input>

</div>

<div class="e">

请填写出库时间:

<el-input v-model="time" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="e">

请填写使用人:

&nbsp;&nbsp;&nbsp;<el-input v-model="name" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="btns">

<el-button type="primary" @click="changeList(1)">确认</el-button>

<el-button type="warning" @click="changeList(2)">取消</el-button>

</div>

</div>

</div>

</template>

<script>

export default {

name: "list",

props: {

index: {

type: Number

},

showflag: {

type: Boolean

}

},

data() {

return {

number: "",

modle: "",

time: "",

brand: "",

status: "",

name:""

};

},

methods: {

changeList(data) {

let flag = data;

var list = {

'number': this.number,

'modle': this.modle,

'time': this.time,

'brand': this.brand,

'status' : this.status,

'name' : this.name

};

this.$emit("changeData", list, this.index, flag);

},

}

};

</script>

<style lang="scss" scoped>

.list-box {

width: 80%;

background: #ffffff;

height: 500px;

top: 100px;

position: absolute;

}

.list {

position: relative;

z-index: 1000;

width: 50%;

position: absolute;

top: 100px;

left: 50%;

transform: translateX(-50%);

height: 350px;

border: 1px solid #f6f6f6;

border-radius: 4px;

background: #ffffff;

padding: 30px 0px 0 30px;

.lista {

width: 50%;

margin-left: 20px;

margin-top: 15px;

}

}

.btns {

margin: 10px auto;

width: 40%;

;

margin-top: 30px;

}

</style>

<template>

<div>

<div class="list-box"></div>

<div class="list">

<div class="a">

请填写编号:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="number" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="b">

请填写品牌:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="brand" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="c">

请填写名称:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="modle" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="d">

请填写入库时间:

<el-input v-model="time" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="e" v-show="showflag">

请填写类型:

&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<el-input v-model="genre" placeholder="请输入内容" class="lista"></el-input>

</div>

<div class="btns">

<el-button type="primary" @click="changeList(1)">确认</el-button>

<el-button type="warning" @click="changeList(2)">取消</el-button>

</div>

</div>

</div>

</template>

<script>

export default {

name: "list",

props: {

index: {

type: Number

},

showflag: {

type: Boolean

}

},

data() {

return {

number: "",

modle: "",

time: "",

brand: "",

genre: ''

};

},

methods: {

changeList(data) {

let flag = data;

if (this.showflag) {

var list = {

'number': this.number,

'modle': this.modle,

'time': this.time,

'brand': this.brand,

'application': this.genre

};

} else {

var list = {

'number': this.number,

'modle': this.modle,

'time': this.time,

'brand': this.brand,

};

}

this.$emit("changeData", list, this.index, flag);

},

}

};

</script>

<style lang="scss" scoped>

.list-box {

width: 100%;

background: #ffffff;

height: 400px;

top: 100px;

position: absolute;

}

.list {

position: relative;

z-index: 1000;

width: 50%;

position: absolute;

top: 100px;

left: 50%;

transform: translateX(-50%);

height: 350px;

border: 1px solid #f6f6f6;

border-radius: 4px;

background: #ffffff;

padding: 30px 0px 0 30px;

.lista {

width: 50%;

margin-left: 20px;

margin-top: 15px;

}

}

.btns {

margin: 10px auto;

width: 40%;

;

margin-top: 30px;

}

</style>

<template>

<div>

<h1>你的账户或者密码不正确</h1>

</div>

</template>

<script>

export default {

name:'err'

}

</script>

<style lang="scss" scoped>

div{

}

</style>

import Vue from 'vue'

import App from './App'

import router from './router'

import ElementUI from 'element-ui';

import 'element-ui/lib/theme-chalk/index.css';

import store from '@/store'

require('./mock/index.js')

Vue.use(ElementUI);

import '@/style/reset.css'

Vue.config.productionTip = false

new Vue({

el: '#app',

router,

store,

render: h => h(App)

})

import storage from './storage'

const TokenKey = 'AUTH\_TOKEN'

const UserKey = 'USER'

export function getToken() {

return (

storage.get(TokenKey) || {

accessToken: '',

expiresIn: 0,

tokenType: 'bearer'

}

)

}

export function setToken(token) {

return storage.set(TokenKey, token)

}

export function removeToken() {

return storage.remove(TokenKey)

}

export function getUser() {

return storage.get(UserKey)

}

export function setUser(user) {

return storage.set(UserKey, user)

}

export function removeUser() {

return storage.remove(UserKey)

}

import axios from 'axios';

import QS from 'qs';

if (process.env.NODE\_ENV == 'development') {

axios.defaults.baseURL = '/api';

} else if (process.env.NODE\_ENV == 'debug') {

axios.defaults.baseURL = '';

} else if (process.env.NODE\_ENV == 'production') {

axios.defaults.baseURL = '/api';

}

axios.defaults.timeout = 10000;

axios.defaults.headers.post['Content-Type'] = 'application/x-www-form-urlencoded;charset=UTF-8';

axios.interceptors.request.use(

config => {

const token = store.state.token;

token && (config.headers.Authorization = token);

console.log('我是请求拦截器')

return config;

},

error => {

return Promise.error(error);

})

axios.interceptors.response.use(

response => {

console.log('成功')

if (response.status === 200) {

return Promise.resolve(response);

} else {

return Promise.reject(response);

}

},

error => {

if (error.response.status) {

switch (error.response.status) {

case 401:

router.replace({

path: '/login',

query: { redirect: router.currentRoute.fullPath }

});

break;

case 403:

Toast({

message: '登录过期，请重新登录',

duration: 1000,

forbidClick: true

});

localStorage.removeItem('token');

store.commit('loginSuccess', null);

setTimeout(() => {

router.replace({

path: '/login',

query: {

redirect: router.currentRoute.fullPath

}

});

}, 1000);

break;

// 404请求不存在

case 404:

Toast({

message: '网络请求不存在',

duration: 1500,

forbidClick: true

});

break;

default:

Toast({

message: error.response.data.message,

duration: 1500,

forbidClick: true

});

}

return Promise.reject(error.response);

}

}

);

export function get(url, params) {

return new Promise((resolve, reject) => {

axios.get(url, {

params: params

})

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

export function post(url, params) {

return new Promise((resolve, reject) => {

axios.post(url, QS.stringify(params))

.then(res => {

resolve(res.data);

})

.catch(err => {

reject(err.data)

})

});

}

function chunk(arr, size) {

var objArr = new Array();

var index = 0;

var objArrLen = arr.length / size;

for (var i = 0; i < objArrLen; i++) {

var arrTemp = new Array();

for (var j = 0; j < size; j++) {

arrTemp[j] = arr[index++];

if (index == arr.length) {

break;

}

}

objArr[i] = arrTemp;

}

return objArr;

}

export default chunk

import Vue from 'vue'

import Storage from 'vue-ls'

Vue.use(Storage, {

namespace: 'MicroSurvey\_\_', // key prefix

name: 'storage', // name variable Vue.[ls] or this.[$ls],

storage: 'local' // storage name session, local, memory

})

export default Vue.storage

.footer{

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

}

html {

color: #000;

background: #fff;

}

body,

div,

dl,

dt,

dd,

ul,

ol,

li,

h1,

h2,

h3,

h4,

h5,

h6,

pre,

form,

fieldset,

input,

textarea,

p,

blockquote,

th,

td {

margin: 0;

padding: 0;

font-family: Verdana, Arial, Helvetica, sans-serif;

}

input:-internal-autofill-previewed,

input:-internal-autofill-selected {

transition: background-color 5000s ease-in-out 0s !important;

background-color: #fff !important;

}

table {

border-collapse: collapse;

border-spacing: 0;

}

fieldset,

img {

border: 0;

}

address,

caption,

cite,

code,

dfn,

em,

strong,

th,

var {

font-style: normal;

font-weight: normal;

}

ol,

ul {

list-style: none;

}

caption,

th {

text-align: left;

}

h1,

h2,

h3,

h4,

h5,

h6 {

/\* font-size: 100%; \*/

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

}

q:before,

q:after {

content: '';

}

abbr,

acronym {

border: 0;

}

import Vue from 'vue'

import VueX from 'vuex'

import kindStore from './user'

Vue.use(VueX)

const store = new VueX.Store({

modules: {

kindStore,

}

})

export default store

import {

getCharge

} from '@/api/http'

import { getToken, setToken} from '@/utile/auth'

import storage from '../../utile/storage'

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

const store = {

state: { // 存放当前页面的状态

token: '1'

},

getters: { // 状态的衍生值，其实就是state中的计算属性

},

actions: { // 处理组件中的异步操作

loginRequest(context,p) {

console.log(storage,'')

console.log(context,p)

getCharge(p).then((res) => {

context.commit('changeToken',res.token)

})

.catch(err => console.log(err))

}

},

mutations: { // 唯一改变当前页面的状态的地方

changeToken(state, data){

state.token = data

getToken({'aa': state.token})

setToken(state.token)

}

}

}

export default store

const login = () => import('@/view/login')

const err = () => import('@/components/err')

const home = () => import('@/view/home/home')

const navHome = () => import('@/view/homeNavList/home')

const phone = () => import('@/view/homeNavList/phone')

const camera = () => import('@/view/homeNavList/camera')

const seat = () => import('@/view/homeNavList/seat')

const electricWire = () => import('@/view/homeNavList/electricWire')

const workStationery = () => import('@/view/homeNavList/workStationery')

const ExWarehouse = () => import('@/view/homeNavList/ExWarehouse')

const routes = [{

path: '/',

name: 'login',

component: login,

}, {

path: '/err',

// name: 'err',

component: err,

meta: {

}

}, {

path: '/home',

name: 'home',

component: home,

meta: {

requireAuth: true

},

children: [{

path: '/home',

name: "navHome",

component: navHome,

meta: ['A级品', '电脑'],

}, {

path: '/phone',

name: "phone",

component: phone,

meta: ['B级品', '电脑配件']

}, {

path: '/camera',

name: "camera",

component: camera,

meta: ['c级品', '摄像头 风扇']

}, {

path: '/seat',

name: "seat",

component: seat,

meta: ['易损品', '座椅']

}, {

path: '/electricWire',

name: 'electricWire',

component: electricWire,

meta: ['网线', '拖线板']

}, {

path: '/workStationery',

name: 'workStationery',

component: workStationery,

meta: ['文具', '办公用品']

}, {

path: '/ExWarehouse',

name: 'ExWarehouse',

component: ExWarehouse,

meta: ['状态', '物品状态']

}]

}]

export default routes

import Vue from 'vue'

import Router from 'vue-router'

import routes from './routes'

Ve.use(Router)

const router = new Router({

routes

})

router.beforeEach((to, from, next) => {

var aa = 1

if (to.meta.requireAuth) {

if (aa) {

next()

} else {

next('/')

}

} else {

next()

}

})

export default router

<template>

<div class="box">

<h2>仓储</h2>

<div class="continer">

<input type="text" placeholder="请输入你的手机" v-model="phone" :state="phoneChange" />

<form action>

<input

type="password"

placeholder="请输入你的密码"

v-model="pass"

:state="passeChange"

autocomplete="tel"

/>

</form>

<button @click="confirmLogin" v-show="2==flag">登陆</button>

<div class="btn" v-show="1==flag" @click="loginHint">登陆</div>

<div class="footer-remind">

<p>温馨提示：</p>

<p>注册过的用户可直接登陆</p>

<p>没有账户的用户可找管理员申请</p>

</div>

</div>

</div>

</template>

<script>

import { mapState, mapActions, mapGetters } from "vuex";

import listComputer from "@/utile/dian";

import parts from "@/utile/accessories";

import fan from "@/utile/fan";

import type from "@/utile/type";

import stationery from "@/utile/stationery";

import seat from "@/utile/seat";

import power from "@/utile/power";

export default {

name: "login",

data() {

return {

pass: "123456",

phone: "18595716975",

btnShow: 1

};

},

computed: {

phoneChange() {

if (this.phoneInput == "") {

return;

} else {

if (/^1[3|4|5|6|7|8|9][0-9]\d{8}$/.test(this.phone)) {

return true;

} else {

return false;

}

}

},

passeChange() {

if (this.pass == "") {

return;

} else {

if (this.pass.length > 5 && this.pass.length < 12) {

// console.log(222);

return true;

} else {

return false;

}

}

},

flag() {

if (this.phoneChange && this.passeChange) {

return 2;

} else {

return 1;

}

}

},

mounted() {

getCharge({'a':2}).then((res) => { console.log(res)

} )

console.log(this.$store.state.kindStore)

this.loginRequest({'a':1})

console.log(this.$router.params)

},

methods: {

...mapActions(["loginRequest"]),

confirmLogin() {

this.loginRequest({ a: 1 }).then(res => {

this.$router.push({ name: "home" });

});

if (!localStorage.getItem("dian")) {

localStorage.setItem("dian", listComputer);

}

if (!localStorage.getItem("parts")) {

localStorage.setItem("parts", parts);

}

if (!localStorage.getItem("fan")) {

localStorage.setItem("fan", fan);

}

if (!localStorage.getItem("seat")) {

localStorage.setItem("seat", seat);

}

if (!localStorage.getItem("stationery")) {

localStorage.setItem("stationery", stationery);

}

if (!localStorage.getItem("power")) {

localStorage.setItem("power", power);

}

if (!localStorage.getItem("type")) {

localStorage.setItem("type", type);

}

},

loginHint() {

if (!/^1[3|4|5|6|7|8|9][0-9]\d{8}$/.test(this.phone)) {

alert("请输入正确的手机号");

} else if (!(this.pass.length > 5 && this.pass.length < 12)) {

alert("请填写正确的6~12位的密码");

}

}

}

};

</script>

<style lang="scss" scoped>

.box {

height: 100vh;

width: 100vw;

background: #324057;

h2 {

text-align: center;

color: #fff;

position: absolute;

top: 25%;

left: 50%;

transform: translateX(-50%);

}

}

.continer {

height: 260px;

width: 370px;

margin: 0 auto;

position: absolute;

top: 50%;

right: 50%;

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

padding-top: 10px;

background: #ffffff;

border-radius: 4px;

input {

height: 35px;

width: 80%;

background: #e8f0fe;

margin-top: 20px;

border: 1px solid #becad8;

border-radius: 4px;

padding-left: 8px;

margin-left: 10%;

box-sizing: border-box;

}

button {

height: 40px;

width: 70%;

margin-left: 15%;

margin-top: 40px;

border: none;

background: #419efa;

border-radius: 4px;

font-size: 18px;

}

.btn {

height: 40px;

width: 70%;

margin-left: 15%;

margin-top: 40px;

border: none;

background: #9dcdfc;

border-radius: 4px;

position: absolute;

text-align: center;

line-height: 40px;

bottom: 70px;

z-index: 66;

}

.footer-remind {

color: red;

font-size: 12px;

text-align: center;

margin-top: 10px;

position: absolute;

bottom: 10px;

left: 50%;

transform: translateX(-50%);

}

}

</style>

<template>

<div>

<head-top></head-top>

<visitor-pie :pieData="pieData"></visitor-pie>

</div>

</template>

<script>

import headTop from '../components/headTop'

import visitorPie from '@/components/visitorPie'

import {getUserCity} from '@/api/getData'

export default {

data(){

return {

pieData: {},

}

},

components: {

headTop,

visitorPie,

},

mounted(){

this.initData();

},

methods: {

async initData(){

try{

const res = await getUserCity();

if (res.status == 1) {

this.pieData = res.user\_city;

}else{

throw new Error(res)

}

}catch(err){

console.log('获取用户分布信息失败',err);

}

},

}

}

</script>

<style lang="less">

@import '../style/mixin';

</style>

<template>

<div class="continer">

<div class="box">

<h2>&nbsp;&nbsp;诚望仓储管理系统</h2>

<el-row class="tac">

<el-col :span="24">

<el-menu

:default-active="defaultActive"

theme="dark"

class="el-menu-vertical-demo"

background-color="#313F56"

text-color="#ffffff"

style="border:none"

router

>

<el-submenu index="1">

<template slot="title">

<i class="el-icon-s-home"></i>

<span>电子系列</span>

</template>

<el-menu-item index="home">A级产品</el-menu-item>

<el-menu-item index="phone">B级产品</el-menu-item>

<el-menu-item index="camera">C级产品</el-menu-item>

</el-submenu>

<el-submenu index="2">

<template slot="title">

<i class="el-icon-s-fold"></i>

<span>硬件系类</span>

</template>

<el-menu-item index="seat">桌椅系列</el-menu-item>

<el-menu-item index="electricWire">网线，拖线板</el-menu-item>

<el-menu-item index="workStationery">办公用品</el-menu-item>

</el-submenu>

<el-submenu index="3">

<template slot="title">

<i class="el-icon-s-order"></i>

<span>仓储物品状态</span>

</template>

<el-menu-item index="ExWarehouse">仓储物品状态</el-menu-item>

</el-submenu>

</el-menu>

</el-col>

</el-row>

</div>

<div class="right">

<header-top></header-top>

<router-view></router-view>

</div>

</div>

</template>

<script>

import HelloWorld from "@/components/HelloWorld";

import headerTop from "@/components/headerTop";

export default {

name: "home",

data() {

return {

hh: "66",

a:1

};

},

computed: {

defaultActive: function() {

return this.$route.path.replace("/", "");

}

},

components: {

HelloWorld,

headerTop

},

watch: {

$route(newVal, oldVal) {

// console.log(newVal,'[[[[[[')

}

},

created() {

// this.$emit("fn");

// console.log(this.$route.path,'=======')

this.$router.push({ path: "/home" });

},

methods: {

fn(msg) {

}

}

};

</script>

<style lang="scss" scoped>

.box {

margin: 0;

width: 15%;

float: left;

min-height: 100vh;

background: #313f56;

}

h2 {

color: #ffffff;

}

.right {

float: right;

height: 100vh;

overflow: auto;

/background: #8A2BE2;

width: 85%;

padding-left: 0px;

box-sizing: border-box;

}

</style>

<template>

<div>

<head-top></head-top>

<div class="edit\_container">

<quill-editor v-model="content"

ref="myQuillEditor"

class="editer"

:options="editorOption"

@ready="onEditorReady($event)">

</quill-editor>

</div>

<div class="submit\_btn">

<el-button type="primary" @click="submit">提交</el-button>

</div>

</div>

</template>

<script>

import headTop from '../components/headTop'

import { quillEditor } from 'vue-quill-editor'

export default {

data(){

return {

content: '<h3>文本编辑</h3>',

editorOption: {

}

}

},

components: {

headTop,

quillEditor,

},

computed: {

editor() {

return this.$refs.myQuillEditor.quill

}

},

methods: {

onEditorReady(editor) {

console.log('editor ready!', editor)

},

submit(){

console.log(this.content);

this.$message.success('提交成功！');

}

},

}

<script>

import headTop from '../components/headTop'

import {getOrderList, getOrderCount, getResturantDetail, getUserInfo, getAddressById} from '@/api/getData'

export default {

data(){

return {

tableData: [],

currentRow: null,

offset: 0,

limit: 20,

count: 0,

currentPage: 1,

restaurant\_id: null,

expendRow: [],

}

},

components: {

headTop,

},

created(){

this.restaurant\_id = this.$route.query.restaurant\_id;

this.initData();

},

mounted(){

},

methods: {

async initData(){

try{

const countData = await getOrderCount({restaurant\_id: this.restaurant\_id});

if (countData.status == 1) {

this.count = countData.count;

}else{

throw new Error('获取数据失败');

}

this.getOrders();

}catch(err){

console.log('获取数据失败', err);

}

},

handleSizeChange(val) {

console.log(`每页 ${val} 条`);

},

handleCurrentChange(val) {

this.currentPage = val;

this.offset = (val - 1)\*this.limit;

this.getOrders()

},

async getOrders(){

const Orders = await getOrderList({offset: this.offset, limit: this.limit, restaurant\_id: this.restaurant\_id});

this.tableData = [];

Orders.forEach((item, index) => {

const tableData = {};

tableData.id = item.id;

tableData.total\_amount = item.total\_amount;

tableData.status = item.status\_bar.title;

tableData.user\_id = item.user\_id;

tableData.restaurant\_id = item.restaurant\_id;

tableData.address\_id = item.address\_id;

tableData.index = index;

this.tableData.push(tableData);

})

},

async expand(row, status){

if (status) {

const restaurant = await getResturantDetail(row.restaurant\_id);

const userInfo = await getUserInfo(row.user\_id);

const addressInfo = await getAddressById(row.address\_id);

this.tableData.splice(row.index, 1, {...row, ...{restaurant\_name: restaurant.name, restaurant\_address: restaurant.address, address: addressInfo.address, user\_name: userInfo.username}});

this.$nextTick(() => {

this.expendRow.push(row.index);

})

}else{

const index = this.expendRow.indexOf(row.index);

this.expendRow.splice(index, 1)

}

},

},

}

</script>

<style lang="less">

@import '../style/mixin';

.edit\_container{

padding: 40px;

margin-bottom: 40px;

}

.editer{

height: 350px;

}

.submit\_btn{

text-align: center;

}

</style>

.bis(@url) {

background-image: url(@url);

background-repeat: no-repeat;

background-size: 100% 100%;

}

.allcover{

position:absolute;

top:0;

right:0;

}

.ctt {

position: absolute;

top: 50%;

left: 50%;

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

}

.ctp(@width, @height) {

position: absolute;

top: 50%;

left: 50%;

margin-top: -@height/2;

margin-left: -@width/2;

}

.tb {

position: absolute;

top: 50%;

transform: translateY(-50%);

}

.lr {

position: absolute;

left: 50%;

transform: translateX(-50%);

}

.wh(@width, @height){

width: @width;

height: @height;

}

.ft(@size, @line-height) {

font-size: @size;

line-height:@line-height;

}

.sc(@size, @color){

font-size: @size;

color: @color;

}

body, div, span, header, footer, nav, section, aside, article, ul, dl, dt, dd, li, a, p, h1, h2, h3, h4,h5, h6, i, b, textarea, button, input, select, figure, figcaption {

padding: 0;

margin: 0;

list-style: none;

font-style: normal;

text-decoration: none;

border: none;

font-family: "Microsoft Yahei",sans-serif;

-webkit-tap-highlight-color:transparent;

-webkit-font-smoothing: antialiased;

&:focus {

outline: none;

}

}

{

width: 0px;

height: 0px;

background-color: #F5F5F5;

}

::-webkit-scrollbar-track

{

-webkit-box-shadow: inset 0 0 1px rgba(0,0,0,0);

border-radius: 10px;

background-color: #F5F5F5;

}

::-webkit-scrollbar-thumb

{

border-radius: 10px;

-webkit-box-shadow: inset 0 0 6px rgba(0,0,0,.3);

background-color: #555;

}

input[type="button"], input[type="submit"], input[type="search"], input[type="reset"] {

-webkit-appearance: none;

}

textarea { -webkit-appearance: none;}

html,body{

height: 100%;

width: 100%;

// background-color: #F5F5F5;

}

.fillcontain{

height: 100%;

width: 100%;

}

.clear:after{

content: '';

display: block;

clear: both;

}

.clear{

zoom:1;

}

.back\_img{

background-repeat: no-repeat;

background-size: 100% 100%;

}

.margin{

margin: 0 auto;

}

.left{

float: left;

}

.right{

float: right;

}

.hide{

display: none;

}

.show{

display: block;

}

.ellipsis{

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.fj(@type: space-between){

display: flex;

justify-content: @type;

}

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存风扇</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>型号</p>

<p>类型</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: true,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("fan");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("fan", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);

this.data.splice(index, 1);

localStorage.setItem("fan", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, flag) {

if (flag === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("fan", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("fan", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("fan", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存线板</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>线板型号</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("power");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("power", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);

this.data.splice(index, 1);

localStorage.setItem("power", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

c

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("power", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("power", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("dpower", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存线板</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>物品型号</p>

<p>存储状态</p>

<p>出库时间</p>

<p>使用人</p>

<p>操作</p>

</div>

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

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

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

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

<div class="number">{{ item.status | status}}</div>

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/loan";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("type");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("type", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);780

this.data.splice(index, 1);

localStorage.setItem("type", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("type", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("type", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("type", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存电脑</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>电脑型号</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("dian");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("dian", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);

this.data.splice(index, 1);

localStorage.setItem("dian", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存电脑</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>电脑型号</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("dian");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("dian", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);

this.data.splice(index, 1);

localStorage.setItem("dian", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("dian", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存桌椅</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>桌椅型号</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("seat");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("seat", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);780

this.data.splice(index, 1);

localStorage.setItem("seat", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

<template>

<div class="demo">

<div class="continer">

<div class="head">

<div class="text">&nbsp;&nbsp;公司库存桌椅</div>

<el-button type="warning" class="btn" @click="addList">添加</el-button>

</div>

<div class="content">

<div class="title">

<p>品牌</p>

<p>编号</p>

<p>桌椅型号</p>

<p>入库时间</p>

<p>操作</p>

</div>

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

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

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

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

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

<div class="operation">

<el-button type="warning" icon="el-icon-edit" circle @click="changeLIst(index)"></el-button>

<el-button type="danger" icon="el-icon-delete" circle @click="removeList(index)"></el-button>

</div>

</div>

<change-list

:index="index"

:showflag="showflag"

@changeData="changListData"

v-show="showList"

></change-list>

<el-button type="text"></el-button>

</div>

<div class="footer">

<el-pagination

background

:page-size="13"

:pager-count="7"

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

</div>

</div>

</template>

<script>

import changeList from "@/components/changaList";

import chhunk from "@/utile/chunk";

import { constants } from "fs";

export default {

name: "computer",

components: {

changeList

},

data() {

return {

list: [],

showList: false,

index: 0,

showflag: false,

data: "",

total: "",

page: 0

};

},

created() {

var list = localStorage.getItem("seat");

this.list = JSON.parse(list);

this.total = this.list.length;

this.data = chhunk(this.list, 13)[0];

},

methods: {

removeList(index) {

var a = confirm("确认删除吗");

if (a) {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list.splice(num, 1);

this.data.splice(index, 1);

localStorage.setItem("seat", JSON.stringify(this.list));

} else {

this.list.splice(index, 1);780

this.data.splice(index, 1);

localStorage.setItem("seat", JSON.stringify(this.list));

}

}

},

currentChange(cpage) {

let n = cpage - 1;

this.data = chhunk(this.list, 13)[n];

this.page = cpage;

},

pre(cpage) {

this.page = cpage;

},

next(cpage) {

this.page = cpage;

},

changeLIst(index) {

this.showList = true;

this.index = index;

},

changListData(list, index, a) {

if (a === 1) {

if (list.modle && list.time && list.brand && list.number) {

if (index === 1875637) {

this.list.push(list);

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

} else {

if (this.page) {

let Page = this.page - 1;

let num = Page \* 13 + index;

this.list[num] = list;

this.data[index] = list;

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

} else {

this.list[index] = list;

this.data[index] = list;

localStorage.setItem("seat", JSON.stringify(this.list));

this.showList = false;

}

}

} else {

this.showList = false;

this.open();

}

} else {

this.showList = false;

}

},

addList() {

this.showList = true;

this.index = 1875637;

},

open() {

this.$alert("请准确的填入信息", {

confirmButtonText: "确定"

});

}

}

};

</script>

<style lang="scss" scoped>

.demo {

height: 900px;

overflow: auto;

}

.continer {

.head {

height: 50px;

padding-left: 15px;

line-height: 50px;

.text {

float: left;

}

.btn {

float: right;

margin-right: 50px;

}

}

}

.content {

.title {

height: 50px;

display: flex;

width: 100%;

flex: 1;

p {

flex: 1;

text-align: center;

line-height: 50px;

&:hover {

background: #f6f6f6;

}

}

}

.list {

height: 50px;

display: flex;

width: 100%;

flex: 1;

div {

flex: 1;

text-align: center;

line-height: 50px;

}

}

}

.footer {

height: 30px;

margin: 0 auto;

margin-bottom: 30x;

display: flex;

justify-content: center;

}

</style>

'use strict'

require('./check-versions')()

process.env.NODE\_ENV = 'production'

const ora = require('ora')

const rm = require('rimraf')

const path = require('path')

const chalk = require('chalk')

const webpack = require('webpack')

const config = require('../config')

const webpackConfig = require('./webpack.prod.conf')

const spinner = ora('building for production...')

spinner.start()

rm(path.join(config.build.assetsRoot, config.build.assetsSubDirectory), err => {

if (err) throw err

webpack(webpackConfig, (err, stats) => {

spinner.stop()

if (err) throw err

process.stdout.write(stats.toString({

colors: true,

modules: false,

children: false, // If you are using ts-loader, setting this to true will make TypeScript errors show up during build.

chunks: false,

chunkModules: false

}) + '\n\n')

if (stats.hasErrors()) {

console.log(chalk.red(' Build failed with errors.\n'))

process.exit(1)

}

console.log(chalk.cyan(' Build complete.\n'))

console.log(chalk.yellow(

' Tip: built files are meant to be served over an HTTP server.\n' +

' Opening index.html over file:// won\'t work.\n'

))

})

})

'use strict'

const path = require('path')

const utils = require('./utils')

const config = require('../config')

const vueLoaderConfig = require('./vue-loader.conf')

function resolve (dir) {

return path.join(\_\_dirname, '..', dir)

}

module.exports = {

context: path.resolve(\_\_dirname, '../'),

entry: {

app: './src/main.js'

},

output: {

path: config.build.assetsRoot,

filename: '[name].js',

publicPath: process.env.NODE\_ENV === 'production'

? config.build.assetsPublicPath

: config.dev.assetsPublicPath

},

resolve: {

extensions: ['.js', '.vue', '.json'],

alias: {

'vue$': 'vue/dist/vue.esm.js',

'@': resolve('src'),

}

},

module: {

rules: [

{

test: /\.vue$/,

loader: 'vue-loader',

options: vueLoaderConfig

},

{

test: /\.js$/,

loader: 'babel-loader',

include: [resolve('src'), resolve('test'), resolve('node\_modules/webpack-dev-server/client')]

},

{

test: /\.(png|jpe?g|gif|svg)(\?.\*)?$/,

loader: 'url-loader',

options: {

limit: 10000,

name: utils.assetsPath('img/[name].[hash:7].[ext]')

}

},

{

test: /\.(mp4|webm|ogg|mp3|wav|flac|aac)(\?.\*)?$/,

loader: 'url-loader',

options: {

limit: 10000,

name: utils.assetsPath('media/[name].[hash:7].[ext]')

}

},

{

test: /\.(woff2?|eot|ttf|otf)(\?.\*)?$/,

loader: 'url-loader',

options: {

limit: 10000,

name: utils.assetsPath('fonts/[name].[hash:7].[ext]')

}

}

]

},

node: {

setImmediate: false,

dgram: 'empty',

fs: 'empty',

net: 'empty',

tls: 'empty',

child\_process: 'empty'

}

}

import deindent from 'de-indent'

import { parseHTML } from 'compiler/parser/html-parser'

import { makeMap } from 'shared/util'

const splitRE = /\r?\n/g

const replaceRE = /./g

const isSpecialTag = makeMap('script,style,template', true)

Parse a single-file component (\*.vue) file into an SFC Descriptor Object.

export function parseComponent (

content: string,

options?: Object = {}

): SFCDescriptor {

const sfc: SFCDescriptor = {

template: null,

script: null,

styles: [],

customBlocks: [],

errors: []

}

let depth = 0

let currentBlock: ?SFCBlock = null

let warn = msg => {

sfc.errors.push(msg)

}

if (process.env.NODE\_ENV !== 'production' && options.outputSourceRange) {

warn = (msg, range) => {

const data: WarningMessage = { msg }

if (range.start != null) {

data.start = range.start

}

if (range.end != null) {

data.end = range.end

}

sfc.errors.push(data)

}

}

function start (

tag: string,

attrs: Array<ASTAttr>,

unary: boolean,

start: number,

end: number

) {

if (depth === 0) {

currentBlock = {

type: tag,

content: '',

start: end,

attrs: attrs.reduce((cumulated, { name, value }) => {

cumulated[name] = value || true

return cumulated

}, {})

}

if (isSpecialTag(tag)) {

checkAttrs(currentBlock, attrs

} else { // custom blocks

sfc.customBlocks.push(currentBlock)

}

}

if (!unary) {

depth++

}

}

function checkAttrs (block: SFCBlock, attrs: Array<ASTAttr>) {

for (let i = 0; i < attrs.length; i++) {

const attr = attrs[i]

if (attr.name === 'lang') {

block.lang = attr.value

}

if (attr.name === 'scoped') {

block.scoped = true

}

if (attr.name === 'module') {

block.module = attr.value || true

}

if (attr.name === 'src') {

block.src = attr.value

}

}

}

function end (tag: string, start: number) {

if (depth === 1 && currentBlock) {

currentBlock.end = start

let text = content.slice(currentBlock.start, currentBlock.end)

if (options.deindent !== false) {

text = deindent(text)

}

pad content so that linters and pre-processors can output correct

line numbers in errors and warnings

if (currentBlock.type !== 'template' && options.pad) {

text = padContent(currentBlock, options.pad) + text

}

currentBlock.content = text

currentBlock = null

}

depth--

}

function padContent (block: SFCBlock, pad: true | "line" | "space") {

if (pad === 'space') {

return content.slice(0, block.start).replace(replaceRE, ' ')

} else {

const offset = content.slice(0, block.start).split(splitRE).length

const padChar = block.type === 'script' && !block.lang

? '//\n'

: '\n'

return Array(offset).join(padChar)

}

}

parseHTML(content, {

warn,

start,

end,

outputSourceRange: options.outputSourceRange

})

return sfc

}

export const setStore = (name, content) => {

if (!name) return;

if (typeof content !== 'string') {

content = JSON.stringify(content);

}

window.localStorage.setItem(name, content);

}

export const getStore = name => {

if (!name) return;

return window.localStorage.getItem(name);

}

export const removeStore = name => {

if (!name) return;

window.localStorage.removeItem(name);

}

export const getStyle = (element, attr, NumberMode = 'int') => {

let target;

if (attr === 'scrollTop') {

target = element.scrollTop;

}else if(element.currentStyle){

target = element.currentStyle[attr];

}else{

target = document.defaultView.getComputedStyle(element,null)[attr];

}

return NumberMode == 'float'? parseFloat(target) : parseInt(target);

}

export const loadMore = (element, callback) => {

let windowHeight = window.screen.height;

let height;

let setTop;

let paddingBottom;

let marginBottom;

let requestFram;

let oldScrollTop;

document.body.addEventListener('scroll',() => {

loadMore();

}, false)

element.addEventListener('touchstart',() => {

height = element.offsetHeight;

setTop = element.offsetTop;

paddingBottom = getStyle(element,'paddingBottom');

marginBottom = getStyle(element,'marginBottom');

},{passive: true})

element.addEventListener('touchmove',() => {

loadMore();

},{passive: true})

element.addEventListener('touchend',() => {

oldScrollTop = document.body.scrollTop;

moveEnd();

},{passive: true})

const moveEnd = () => {

requestFram = requestAnimationFrame(() => {

if (document.body.scrollTop != oldScrollTop) {

oldScrollTop = document.body.scrollTop;

loadMore();

moveEnd();

}else{

cancelAnimationFrame(requestFram);

height = element.offsetHeight;

loadMore();

}

})

}

const loadMore = () => {

if (document.body.scrollTop + windowHeight >= height + setTop + paddingBottom + marginBottom) {

callback();

}

}

}

export const showBack = callback => {

let requestFram;

let oldScrollTop;

document.addEventListener('scroll',() => {

showBackFun();

}, false)

document.addEventListener('touchstart',() => {

showBackFun();

},{passive: true})

document.addEventListener('touchmove',() => {

showBackFun();

},{passive: true})

document.addEventListener('touchend',() => {

oldScrollTop = document.body.scrollTop;

moveEnd();

},{passive: true})

const moveEnd = () => {

requestFram = requestAnimationFrame(() => {

if (document.body.scrollTop != oldScrollTop) {

oldScrollTop = document.body.scrollTop;

moveEnd();

}else{

cancelAnimationFrame(requestFram);

}

showBackFun();

})

}

const showBackFun = () => {

if (document.body.scrollTop > 500) {

callback(true);

}else{

callback(false);

}

}

}

export const animate = (element, target, duration = 400, mode = 'ease-out', callback) => {

clearInterval(element.timer);

if (duration instanceof Function) {

callback = duration;

duration = 400;

}else if(duration instanceof String){

mode = duration;

duration = 400;

}

if (mode instanceof Function) {

callback = mode;

mode = 'ease-out';

}

const attrStyle = attr => {

if (attr === "opacity") {

return Math.round(getStyle(element, attr, 'float') \* 100);

} else {

return getStyle(element, attr);

}

}

const rootSize = parseFloat(document.documentElement.style.fontSize);

const unit = {};

const initState = {};

Object.keys(target).forEach(attr => {

if (/[^\d^\.]+/gi.test(target[attr])) {

unit[attr] = target[attr].match(/[^\d^\.]+/gi)[0] || 'px';

}else{

unit[attr] = 'px';

}

initState[attr] = attrStyle(attr);

});

Object.keys(target).forEach(attr => {

if (unit[attr] == 'rem') {

target[attr] = Math.ceil(parseInt(target[attr])\*rootSize);

}else{

target[attr] = parseInt(target[attr]);

}

});

let flag = true; //假设所有运动到达终点

const remberSpeed = {};//记录上一个速度值,在ease-in模式下需要用到

element.timer = setInterval(() => {

Object.keys(target).forEach(attr => {

let iSpeed = 0; //步长

let status = false; //是否仍需运动

let iCurrent = attrStyle(attr) || 0; //当前元素属性址

let speedBase = 0; //目标点需要减去的基础值，三种运动状态的值都不同

let intervalTime; //将目标值分为多少步执行，数值越大，步长越小，运动

switch(mode){

case 'ease-out':

speedBase = iCurrent;

intervalTime = duration\*5/400;

break;

case 'linear':

speedBase = initState[attr];

intervalTime = duration\*20/400;

break;

case 'ease-in':

let oldspeed = remberSpeed[attr] || 0;

iSpeed = oldspeed + (target[attr] - initState[attr])/duration;

remberSpeed[attr] = iSpeed

break;

default:

speedBase = iCurrent;

intervalTime = duration\*5/400;

}

if (mode !== 'ease-in') {

iSpeed = (target[attr] - speedBase) / intervalTime;

iSpeed = iSpeed > 0 ? Math.ceil(iSpeed) : Math.floor(iSpeed);

}

switch(mode){

case 'ease-out':

status = iCurrent != target[attr];

break;

case 'linear':

status = Math.abs(Math.abs(iCurrent) - Math.abs(target[attr])) > Math.abs(iSpeed);

break;

case 'ease-in':

status = Math.abs(Math.abs(iCurrent) - Math.abs(target[attr])) > Math.abs(iSpeed);

break;

default:

status = iCurrent != target[attr];

}

if (status) {

flag = false;

if (attr === "opacity") {

element.style.filter = "alpha(opacity:" + (iCurrent + iSpeed) + ")";

element.style.opacity = (iCurrent + iSpeed) / 100;

} else if (attr === 'scrollTop') {

element.scrollTop = iCurrent + iSpeed;

}else{

element.style[attr] = iCurrent + iSpeed + 'px';

}

} else {

flag = true;

}

if (flag) {

clearInterval(element.timer);

if (callback) {

callback();

}

}

})

}, 20);

}