<template>

<div id="app">

<div id="myDiv">

</div>

<router-view></router-view>

</div>

</template>

<script>

export default {

name: "App",

data(){

return{

list:[]

}

},

};

</script>

<style>

#app {

position: relative;

}

.lizi {

z-index: -1;

position: absolute;

}

</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 VueParticles from 'vue-particles'

Vue.use(VueParticles)

import $ from 'jquery'

import Axios from 'axios'

Vue.prototype.$axios = Axios

Axios.defaults.baseURL = '/api'

Axios.defaults.headers.post['Content-Type'] = 'application/json'

Vue.use(ElementUI);

Vue.config.productionTip = false

/\* eslint-disable no-new \*/

new Vue({

el: '#app',

router,

render: h => h(App)

})

ar action = "get";

var sql = "select \* from info";

export function postData(id) {

// sql = "insert into info values('3','3','3','3')";

// sql = "update info set name='5'";

sql = "delete info where id="+id;

console.log(sql,'-------------')

$.ajax({

type: 'post',

url: 'http://192.168.18.26:8087/ASHX/PublicClass.ashx?action=get',

async: false,

data: 'context=' + escape(sql),

success: function (data) {

console.log(data,'-------')

},

error: function () {

top.$.messager.alert("提示", "请求失败！");

}

});

}

export function changeData(date){

sql = "insert into info values('3','3','3','3')"

console.log(sql,'-------------',date,)

$.ajax({

type: 'post',

url: 'http://192.168.18.26:8087/ASHX/PublicClass.ashx?action=get',

async: false,

data: 'context=' + escape(sql),

success: function (data) {

console.log(data,'-------')

},

error: function () {

top.$.messager.alert("提示", "请求失败！");

}

});

}

export function updata(id){

sql = "update info set name='55',time='777',sum='66' where city='3'";

console.log(sql,'-------------',)

$.ajax({

type: 'post',

url: 'http://192.168.18.26:8087/ASHX/PublicClass.ashx?action=get',

async: false,

data: 'context=' + escape(sql),

success: function (data) {

console.log(data,'-------')

},

error: function () {

top.$.messager.alert("提示", "请求失败！");

}

});

}

import Vue from 'vue'

import Router from 'vue-router'

import table from '@/components/table'

Vue.use(Router)

export default new Router({

routes: [

{

path: '/',

name: 'table',

component: table

}

]

})

<template>

<div class="continer" :style="{background:bag,color:fontColor}">

<!-- <div class="title-car">诚望科技集团股份有限公司销售业绩表</div>

<header>

<div class="title">

<div @contextmenu.prevent="show()">销售人</div>

<div>客户</div>

<div>时间</div>

<div>金额</div>

</div>

</header>-->

<div class="table" id="out-table">

<!-- <layout :list="list"></layout> -->

<table

id="tableExcel"

class="table-ex"

border="1"

cellspacing="0"

width="100%"

align="center"

>

<tr>

<td colspan="5" align="center" class="hedaer">诚望科技集团股份有限公司销售业绩表</td>

</tr>

<tr>

<td>销售工号</td>

<td>时间（19年/月份）</td>

<td>销售金额</td>

<td>销售提成</td>

<td>操作</td>

</tr>

<tr width="100%" v-for="(item, tdindex) in list" :id="item.id">

<td width="20" height="50px" frame="vsides">{{ item.name }}</td>

<td width="20">{{ item.city }}</td>

<td width="20">{{ item.time }}</td>

<td width="20" frame="vsides">{{ item.sum }}</td>

<td width="20">

<button class="btn-change" @click="update(item.id,item)">修改数据</button>

<button class="move" @click="open(item.id)">删除</button>

</td>

</tr>

</table>

</div>

<div class="block">

<el-pagination

background

:page-size="10"

:pager-count="7"

hide-on-single-page

@current-change="currentChange"

@prev-click="pre"

@next-click="next"

:total="total"

></el-pagination>

</div>

<div class="options\_color">

<span>自定义背景颜色</span>

<input type="color" v-model="bag" @click="clickColor" />

<span>自定义字体颜色</span>

<input type="color" v-model="fontColor" />

<button @click="method5('tableExcel')">导出数据</button>

<button @click="add">添加数据</button>

</div>

<exta v-show="show" name="excel" class="excel" @submit="put" :flag="flag" :id="id" :date="fromt"></exta>

</div>

</template>

<script>

import api from "@/uitls/ajax";

import layout from "@/components/layout";

import exta from "@/components/excekl";

import { postData,} from "@/uitls/jqa";

import getData from "@/uitls/jq";

export default {

name: "ta",

components: {

layout

},

components: { exta },

data() {

return {

total: 0,

bag: "#ffffff",

fontColor: "#000000",

list: [],

pageNumbar: 0,

numberOfBranches: 10,

arr: [],

idTmr: "",

tableToExcel: "",

show: false,

flag:'',

id:'',

fromt:{}

};

},

created() {

let bag = localStorage.getItem("bag");

if (bag) {

this.bag = bag;

}

getData();

},

watch: {

bag(newVal, oldVal) {

localStorage.setItem("bag", newVal);

}

},

methods: {

currentChange(cpage) {

this.list = this.arr[cpage - 1];

},

pre(cpage) {

this.list = this.arr[cpage - 1];

},

next(cpage) {

this.list = this.arr[cpage - 1];

},

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;

},

clickColor() {

console.log(111);

},

exportTable() {

console.log(22);

},

getExplorer() {

var explorer = window.navigator.userAgent;

//ie

if (explorer.indexOf("MSIE") >= 0) {

return "ie";

}

//firefox

else if (explorer.indexOf("Firefox") >= 0) {

return "Firefox";

}

else if (explorer.indexOf("Chrome") >= 0) {

return "Chrome";

}

else if (explorer.indexOf("Opera") >= 0) {

return "Opera";

}

else if (explorer.indexOf("Safari") >= 0) {

return "Safari";

}

},

method5(tableid) {

if (this.getExplorer() == "ie") {

var curTbl = document.getElementById("tableid");

var oXL = new ActiveXObject("Excel.Application");

var oWB = oXL.Workbooks.Add();

var xlsheet = oWB.Worksheets(1);

var sel = document.body.createTextRange();

sel.moveToElementText(curTbl);

sel.select();

sel.execCommand("Copy");

xlsheet.Paste();

oXL.Visible = true;

try {

var fname = oXL.Application.GetSaveAsFilename(

"Excel.xls",

"Excel Spreadsheets (\*.xls), \*.xls"

);

} catch (e) {

print("Nested catch caught " + e);

} finally {

oWB.SaveAs(fname);

oWB.Close((savechanges = false));

oXL.Quit();

oXL = null;

this.idTmr = window.setInterval("Cleanup();", 1);

}

} else {

this.tableToExcel(tableid);

}

},

Cleanup() {

window.clearInterval(this.idTmr);

CollectGarbage();

},

put(data) {

this.show = false

},

open(id) {

this.$confirm("此操作将永久删除该数据, 是否继续?", "提示", {

confirmButtonText: "确定",

cancelButtonText: "取消",

type: "warning"

})

.then(() => {

postData(id)

this.$message({

type: "success",

message: "删除成功!"

});

})

.catch(() => {

this.$message({

type: "info",

message: "已取消删除"

});

});

},

add() {

this.show = true;

this.flag = 'add'

this.fromt = {}

},

update(id,data) {

this.fromt = data

this.show = true

this.flag = 'up'

this.id = id

},

},

mounted() {

api.requestList().then(res => {

this.num = res;

this.total = this.num.length;

this.arr = this.chunk(this.num, this.numberOfBranches);

this.list = this.arr[0];

});

this.tableToExcel = (function() {

var uri = "data:application/vnd.ms-excel;base64,",

template =

'<html><head><meta charset="UTF-8"></head><body><table>{table}</table></body></html>',

base64 = function(s) {

return window.btoa(unescape(encodeURIComponent(s)));

},

format = function(s, c) {

return s.replace(/{(\w+)}/g, function(m, p) {

return c[p];

});

};

return function(table, name) {

if (!table.nodeType) table = document.getElementById(table);

var ctx = { worksheet: name || "Worksheet", table: table.innerHTML };

window.location.href = uri + base64(format(template, ctx));

};

})();

}

};

</script>

<style lang="scss" scoped>

html,

body {

height: 100%;

}

body {

margin: 0;

}

.continer {

box-sizing: border-box;

width: 1200px;

background: red;

// border-radius: 10px;

margin: 0 auto;

padding: 0;

}

header {

height: 50px;

background: red;

display: flex;

div {

display: flex;

flex: 1;

text-align: center;

line-height: 50px;

justify-content: center;

color: white;

&.title div:hover {

background: #888888;

}

}

}

.table {

width: 100%;

border: 1px black solid;

box-sizing: border-box;

}

.content {

height: 50px;

display: flex;

text-align: center;

line-height: 50px;

justify-content: center;

overflow: hidden;

border-bottom: 1px black solid;

&.content:last-child {

border: none;

}

&.content:hover {

background: #eee;

color: #ffffff;

}

.li {

flex: 1;

box-sizing: border-box;

border-right: 1px black solid;

&.li:last-child {

border: none;

}

}

}

.options\_color {

position: fixed;

right: 0px;

top: 30px;

width: 155px;

p{

margin: 0

};

input{

width: 20px;

height: 20px;

}

}

.block {

float: left;

position: relative;

left: 50%;

top: 20px;

transform: translateX(-50%);

}

.table-ex {

td{

width: 240px;

}

tr {

height: 50px;

td {

text-align: center;

}

}

tr:hover {

background: #ffffff;

td {

color: red;

}

}

}

.hedaer {

background: red;

letter-spacing: 1px;

font-size: 20px;

color: #ffffff;

}

.hedaer:hover {

background: red;

color: #ffffff;

letter-spacing: 1px;

}

</style>

<template>

<div>

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

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

<div class="li">{{ item.city }}</div>

<div class="li">{{ item.time }}</div>

<div class="li">{{ item.sum }}</div>

</div>

</div>

</template>

<script>

export default {

name: 'layout',

props:{

list:{

type:Array

}

},

data() {

return {

// list:[{name:'999',city:'77',time:88,sum:111},{name:'999',city:'77',time:88,sum:111},]

}

},

}

</script>

<style lang="scss" scoped>

.content {

height: 50px;

display: flex;

text-align: center;

line-height: 50px;

justify-content: center;

overflow: hidden;

border-bottom: 1px black solid;

&.content:last-child {

border: none;

}

&.content:hover {

background: #eee;

color: #ffffff;

}

.li {

flex: 1;

box-sizing: border-box;

border-right: 1px black solid;

&.li:last-child {

border: none;

}

}

}

</style>

<template>

<div>

<div class="car">

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

<el-form-item label="销售工号">

<el-input v-model="form.name" placeholder="请输入员工工号"></el-input>

</el-form-item>

<el-form-item label="时间">

<el-input v-model="form.city" placeholder="请输入月份"></el-input>

</el-form-item>

<el-form-item label="金额">

<el-input v-model="form.time" placeholder="请输入销售金额"></el-input>

</el-form-item>

<el-form-item label="提成">

<el-input v-model="form.sum" placeholder="请输入销售提成"></el-input>

</el-form-item>

<el-button type="primary" @click="onSubmit" style="margin-left:30%">立即创建</el-button>

<el-button class="off" @click="off">取消</el-button>

</el-form>

</div>

<img src="../assets/2743ee2bb35d0dfbed92d6a5252ae23.png" class="bi" alt />

</div>

</template>

<script>

import { changeData, updata } from "@/uitls/jqa";

export default {

data() {

return {

form: {

name: "",

city: "",

time: "",

sum: "",

newId:''

}

};

},

props: {

flag: {

type: String

},

id: {

type: String

},

date:{

type: Object

}

},

created() {

console.log(this.flag);

},

watch: {

flag(newVal, oldVal) {

console.log(newVal);

},

id(newVal, oldVal) {

this.newId = newVal

},

date(newVal, oldVal) {

this.form = newVal

},

},

methods: {

onSubmit() {

this.$emit("submit", "off");

if (this.flag == "add") {

var sql = "select \* from info";

console.log(this.form);

var sql =

"insert into info values(" +

this.form.name +

"," +

this.form.city +

"," +

this.form.time +

"," +

this.form.sum +

")";

$.ajax({

type: "post",

url: "http://192.168.18.26:8087/ASHX/PublicClass.ashx?action=get",

async: false,

data: "context=" + escape(sql),

success: function(data) {

window.location.reload();

},

error: function() {

top.$.messager.alert("提示", "请求失败！");

}

});

} else {

sql = "update info set name='55',time='79',sum='' where city='3'";

sql = "update info set name="+this.form.name+",city="+this.form.city+",sum="+this.form.sum+",time="+this.form.time+" where id='"+this.newId+"'"

console.log(sql, "-------------");

$.ajax({

type: "post",

url: "http://192.168.18.26:8087/ASHX/PublicClass.ashx?action=get",

async: false,

data: "context=" + escape(sql),

success: function(data) {

},

error: function() {

top.$.messager.alert("提示", "请求失败！");

}

});

}

},

off() {

this.$emit("submit", "off");

this.$alert("是否放弃本次操作", "用户提示", {

confirmButtonText: "确定",

callback: action => {

this.$message({

type: "info",

message: `操作已取消`

});

}

});

}

}

};

</script>

<style lang="scss" scoped>

.car {

width: 500px;

height: 350px;

background: #f3f3f3;

padding: 20px 20px 0 0;

position: fixed;

top: 50%;

left: 50%;

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

z-index: 2222;

}

.bi {

position: fixed;

top: 0;

right: 0;

bottom: 0;

left: 0;

}

</style>

'use strict'

const path = require('path')

const utils = require('./utils')

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

var webpack=require("webpack")

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: {

because Vue

source contains it (although only uses it if it's native).

setImmediate: false,

// prevent webpack from injecting mocks to Node native modules

// that does not make sense for the client

dgram: 'empty',

fs: 'empty',

net: 'empty',

tls: 'empty',

child\_process: 'empty'

},

plugins: [

new webpack.optimize.CommonsChunkPlugin('common.js'),

new webpack.ProvidePlugin({

jQuery: "jquery",

$: "jquery"

})

]

}

{

"name": "table",

"version": "1.0.0",

"description": "A Vue.js project",

"author": "tj",

"private": true,

"scripts": {

"dev": "webpack-dev-server --inline --progress --config build/webpack.dev.conf.js",

"start": "npm run dev",

"unit": "cross-env BABEL\_ENV=test karma start test/unit/karma.conf.js --single-run",

"e2e": "node test/e2e/runner.js",

"test": "npm run unit && npm run e2e",

"build": "node build/build.js"

},

"dependencies": {

"axios": "^0.19.0",

"jquery": "^2.2.1",

"element-ui": "^2.10.1",

"file-saver": "^2.0.2",

"vue": "^2.5.2",

"vue-router": "^3.0.1",

"xlsx": "^0.15.1"

},

"devDependencies": {

"autoprefixer": "^7.1.2",

"babel-core": "^6.22.1",

"babel-helper-vue-jsx-merge-props": "^2.0.3",

"babel-loader": "^7.1.1",

"babel-plugin-istanbul": "^4.1.1",

"babel-plugin-syntax-jsx": "^6.18.0",

"babel-plugin-transform-runtime": "^6.22.0",

"babel-plugin-transform-vue-jsx": "^3.5.0",

"babel-preset-env": "^1.3.2",

"babel-preset-stage-2": "^6.22.0",

"babel-register": "^6.22.0",

"chai": "^4.1.2",

"chalk": "^2.0.1",

"chromedriver": "^2.27.2",

"copy-webpack-plugin": "^4.0.1",

"cross-env": "^5.0.1",

"cross-spawn": "^5.0.1",

"css-loader": "^0.28.0",

"extract-text-webpack-plugin": "^3.0.0",

"file-loader": "^1.1.4",

"friendly-errors-webpack-plugin": "^1.6.1",

"html-webpack-plugin": "^2.30.1",

"inject-loader": "^3.0.0",

"karma": "^1.4.1",

"karma-coverage": "^1.1.1",

"karma-mocha": "^1.3.0",

"karma-phantomjs-launcher": "^1.0.2",

"karma-phantomjs-shim": "^1.4.0",

"karma-sinon-chai": "^1.3.1",

"karma-sourcemap-loader": "^0.3.7",

"karma-spec-reporter": "0.0.31",

"karma-webpack": "^2.0.2",

"mocha": "^3.2.0",

"mockjs": "^1.0.1-beta3",

"nightwatch": "^0.9.12",

"node-notifier": "^5.1.2",

"node-sass": "^4.12.0",

"optimize-css-assets-webpack-plugin": "^3.2.0",

"ora": "^1.2.0",

"phantomjs-prebuilt": "^2.1.14",

"portfinder": "^1.0.13",

"postcss-import": "^11.0.0",

"postcss-loader": "^2.0.8",

"postcss-url": "^7.2.1",

"rimraf": "^2.6.0",

"sass-loader": "^7.1.0",

"selenium-server": "^3.0.1",

"semver": "^5.3.0",

"shelljs": "^0.7.6",

"sinon": "^4.0.0",

"sinon-chai": "^2.8.0",

"uglifyjs-webpack-plugin": "^1.1.1",

"url-loader": "^0.5.8",

"vue-loader": "^13.3.0",

"vue-particles": "^1.0.9",

"vue-style-loader": "^3.0.1",

"vue-template-compiler": "^2.5.2",

"webpack": "^3.6.0",

"webpack-bundle-analyzer": "^2.9.0",

"webpack-dev-server": "^2.9.1",

"webpack-merge": "^4.1.0"

},

"engines": {

"node": ">= 6.0.0",

"npm": ">= 3.0.0"

},

"browserslist": [

"> 1%",

"last 2 versions",

"not ie <= 8"

]

}