

F:\git\java\mar3\filemonitor\target\src\src-0.doc

0:F:\git\nodejs\vue\vuestep\vueproject\src\api\axiosFun.js

```
import axios from 'axios';
```

```
//
```

```
const loginreq = (method, url, params) => {  
  return axios({  
    method: method,  
    url: url,  
    headers: {  
      'Content-Type': 'application/x-www-form-urlencoded',  
    },  
    data: params,  
    traditional: true,  
    transformRequest: [  
      function(data) {  
        let ret = ''  
        for (let it in data) {  
          ret +=  
            encodeURIComponent(it) +  
            '=' +  
            encodeURIComponent(data[it]) +  
            '&  
        }  
        return ret  
      }  
    ]  
  }).then(res => res.data);  
};
```

```
//
```

```
const req = (method, url, params) => {  
  return axios({  
    method: method,  
    url: url,  
    headers: {  
      'Content-Type': 'application/x-www-form-urlencoded',  
      token: localStorage.getItem('logintoken')  
    },  
    data: params,  
    traditional: true,  
    transformRequest: [  
      function(data) {  
        let ret = ''  
        for (let it in data) {  
          ret +=  
            encodeURIComponent(it) +  
            '=' +  
            encodeURIComponent(data[it]) +  
            '&  
        }  
        return ret  
      }  
    ]  
  });  
};
```

```

function(data) {
  let ret = ""
  for (let it in data) {
    ret +=
      encodeURIComponent(it) +
      '=' +
      encodeURIComponent(data[it]) +
      '&'
  }
  return ret
}
]
}).then(res => res.data);
};

```

```

export {
  loginreq,
  req
}

```

1:F:\git\nodejs\vue\vuestep\vueproject\src\api\basisMG.js

```
import axios from 'axios';
```

```
import { req } from './axiosFun';
```

```

/**
 *
 **/

```

```
// -
```

```
export const GoodsList = (params) => { return req("post", "/api/Goods/list", params) };

```

```
// -
```

```
export const GoodsSave = (params) => { return req("post", "/api/Goods/save", params) };

```

```
// -
```

```
export const GoodsDelete = (params) => { return axios.delete("/api/Goods/delete?ids=" + params + "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

```

```

/**
 *
 **/

```

```
// -
```

```
export const MachineList = (params) => { return req("post", "/api/Machine/list", params) };

```

```
// -
```

```
export const MachineSave = (params) => { return req("post", "/api/Machine/save", params) };

```

```
// -
export const MachineDelete = (params) => { return axios.delete("/api/Machine/delete?ids=" +
params + "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const MachineAisleList = (params) => { return req("post", "/api/MachineAisle/list", params)
};
// -
export const MachineAisleDelete = (params) => { return
axios.delete("/api/MachineAisle/delete?ids=" + params + "&token=" +
localStorage.getItem('logintoken')).then(res => res.data) };
// -
export const MachineAisleRsave = (params) => { return req("post", "/api/MachineAisle/save",
params) };
```

2:F:\git\nodejs\vue\vuestep\vueproject\src\api\payMG.js

import axios from 'axios';

import { req } from './axiosFun';

```
/**
 *
 **/
// -
export const MachineConfigList = (params) => { return req("post", "/api/MachineConfig/list",
params) };
// -
export const MachineConfigSave = (params) => { return req("post", "/api/MachineConfig/save",
params) };
// -
export const MachineConfigDelete = (params) => { return
axios.delete("/api/MachineConfig/delete?ids=" + params + "&token=" +
localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const ConfigList = (params) => { return req("post", "/api/Config/list", params) };
// -
```

```

export const ConfigSave = (params) => { return req("post", "/api/Config/save", params) };
// -
export const ConfigDelete = (params) => { return axios.delete("/api/Config/delete?ids=" + params
+ "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

/**
 * _
 **/
// -
export const OrderList = (params) => { return req("post", "/api/Order/list", params) };
// -s
export const OrderDelete = (params) => { return axios.delete("/api/Order/delete?ids=" + params +
"&token=" + localStorage.getItem('logintoken')).then(res => res.data) };
// -
export const OrderRefund = (params) => { return req("post", "/api/Order/refund", params) };

```

3:F:\git\nodejs\vue\vuestep\vueproject\src\api\userMG.js

```

import axios from 'axios';
import { loginreq, req } from './axiosFun';

//
export const login = (params) => { return loginreq("post", "/api/login", params) };
//
export const menu = (params) => { return axios.get("/api/menu?&token=" +
localStorage.getItem('logintoken')).then(res => res.data) };
//
export const loginout = () => { return axios.delete("/api/login?&token=" +
localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const userList = (params) => { return req("post", "/api/User/list", params) };
// -
export const userSave = (params) => { return req("post", "/api/User/save", params) };
// -
export const userDelete = (params) => { return axios.delete("/api/User/delete?ids=" + params +
"&token=" + localStorage.getItem('logintoken')).then(res => res.data) };
// -
export const userPwd = (params) => { return req("post", "/api/User/pwd", params) };
// -

```

```

export const userLock = (params) => { return axios.get("/api/User/lock?userId=" + params.userId +
"&lock=" + params.lock + "&token=" + localStorage.getItem('logintoken')) };
// -
export const UserDeptTree = (params) => { return axios.get("/api/UserDept/tree/" + params +
"?token=" + localStorage.getItem('logintoken')) };
// -
export const UserDeptSave = (params) => { return req("post", "/api/UserDept/save", params) };
// -
export const UserDeptdeptTree = (params) => { return axios.get("/api/UserDept/deptTree/" +
params + "?token=" + localStorage.getItem('logintoken')) };
// -
export const UserChangeDept = (params) => { return req("post", "/api/User/change/dept", params)
};
// -
export const userExpireToken = (params) => { return req("get", "/api/User/expireToken/" +
params, {}) };
// -
export const userFlashCache = (params) => { return req("get", "/api/User/flashCache/" + params,
{}); };

/**
 *
 **/
// -Module/list
export const ModuleList = () => { return req("post", "/api/Module/list") };
// -
export const ModuleGet = (params) => { return axios.get("/api/Module/get/" + params + "?token="
+ localStorage.getItem('logintoken')) };
// -Module/nodes
export const ModuleNodes = (params) => { return req("post", "/api/Module/nodes", params) };
// -
export const ModuleSave = (params) => { return req("post", "/api/Module/save", params) };
// -
export const ModuleDelete = (params) => { return axios.delete("/api/Module/delete?ids=" +
params + "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const roleList = (params) => { return req("post", "/api/Role/list", params) };
// -

```

```

export const roleSave = (params) => { return req("post", "/api/Role/save", params) };
// -
export const roleDelete = (params) => { return axios.delete("/api/Role/delete?ids=" + params +
"&token=" + localStorage.getItem('logintoken')).then(res => res.data) };
// -
export const RoleRightTree = (params) => { return axios.get("/api/RoleRight/tree/" + params +
"?token=" + localStorage.getItem('logintoken')) };
// -
export const RoleRightSave = (params) => { return req("post", "/api/RoleRight/save", params) };

/**
 *
 **/
// -
export const deptList = (params) => { return req("post", "/api/Dept/list", params) };
// -
export const deptSave = (params) => { return req("post", "/api/Dept/save", params) };
// -
export const deptDelete = (params) => { return axios.get("/api/Dept/delete?ids=" + params +
"&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const variableList = (params) => { return req("post", "/api/Variable/list", params) };
// -
export const variableSave = (params) => { return req("post", "/api/Variable/save", params) };
// -
export const variableDelete = (params) => { return axios.delete("/api/Variable/delete?ids=" +
params + "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

/**
 *
 **/
// -
export const permissionList = (params) => { return req("post", "/api/Permission/list", params) };
// -
export const ermissionSave = (params) => { return req("post", "/api/Permission/save", params) };
// -
export const ermissionDelete = (params) => { return axios.delete("/api/Permission/delete?ids=" +
params + "&token=" + localStorage.getItem('logintoken')).then(res => res.data) };

```

```
// -
export const roleDropDown = () => { return axios.get("/api/Role/dropDown/all?&token=" +
localStorage.getItem('logintoken')).then(res => res.data) };
// -
export const RolePermission = (params) => { return req("post", "/api/RolePermission/save",
params) };
```

4:F:\git\nodejs\vue\vuestep\vueproject\src\App.vue

```
<template>
  <div id="app">
    <router-view/>
  </div>
</template>
```

```
<script>
export default {
  name: "App"
};
</script>
```

```
<style>
html,
body {
  width: 100%;
  height: 100%;
  box-sizing: border-box;
  padding: 0px;
  margin: 0px;
}
#app {
  font-family: "Avenir", Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  color: #2c3e50;
  widows: 100%;
  height: 100%;
}
</style>
```

5:F:\git\nodejs\vue\vuestep\vueproject\src\components\error.vue

```
/**
```

```
* 404
```

```

*/
<template>
  <div class="page-404">
    <h1>404!</h1>
    <h2>Sorry, page not found</h2>
  </div>
</template>

<style scoped>
.page-404 {
  text-align: center;
}
.page-404 h1 {
  font-size: 98px;
  font-weight: 700;
  line-height: 150px;
  text-shadow: rgba(61, 61, 61, 0.3) 1px 1px, rgba(61, 61, 61, 0.2) 2px 2px,
    rgba(61, 61, 61, 0.3) 3px 3px;
}
.page-404 h2 {
  line-height: 30px;
  font-size: 30px;
}
</style>

```

6:F:\git\nodejs\vue\vuestep\vueproject\src\components\leftnav.vue

```

/**
*
*/
<template>
  <el-menu default-active="2" :collapse="collapsed" collapse-transition router :default-
active="$route.path" unique-opened class="el-menu-vertical-demo" background-color="#334157"
text-color="#fff" active-text-color="#ffd04b">
    <div class="logobox">
      
    </div>
    <el-submenu v-for="menu in allmenu" :key="menu.menuid" :index="menu.menuname">
      <template slot="title">
        <i class="iconfont" :class="menu.icon"></i>
        <span>{{menu.menuname}}</span>
      </template>
    <el-menu-item-group>

```



```

      <el-menu-item v-for="chmenu in menu.menus" :index=""/>+chmenu.url"
:key="chmenu.menuid">
      <i class="iconfont" :class="chmenu.icon"></i>
      <span>{{chmenu.menuname}}</span>
    </el-menu-item>
  </el-menu-item-group>
</el-submenu>
</el-menu>
</template>
<script>
import { menu } from '../api/userMG'
export default {
  name: 'leftnav',
  data() {
    return {
      collapsed: false,
      allmenu: []
    }
  },
  // ()
  created() {
    //
    let res = {
      success: true,
      data: [
        {
          menuid: 1,
          icon: 'li-icon-xiangmuguanli',
          menuname: '',
          hasThird: null,
          url: null,
          menus: [
            {
              menuid: 2,
              icon: 'icon-cat-skuQuery',
              menuname: '',
              hasThird: 'N',
              url: 'goods/Goods',
              menus: null
            }
          ]
        }
      ],
    },

```

```
{
  menuid: 33,
  icon: 'li-icon-dingdanguanli',
  menuname: '',
  hasThird: null,
  url: null,
  menus: [
    {
      menuid: 34,
      icon: 'icon-order-manage',
      menuname: '',
      hasThird: 'N',
      url: 'pay/Order',
      menus: null
    }
  ]
},
{
  menuid: 71,
  icon: 'li-icon-xitongguanli',
  menuname: '',
  hasThird: null,
  url: null,
  menus: [
    {
      menuid: 72,
      icon: 'icon-cus-manage',
      menuname: '',
      hasThird: 'N',
      url: 'system/user',
      menus: null
    },
    {
      menuid: 174,
      icon: 'icon-cms-manage',
      menuname: '',
      hasThird: 'N',
      url: 'system/Module',
      menus: null
    },
    {
      menuid: 73,
```

```
    icon: 'icon-news-manage',
    menuname: '',
    hasThird: 'N',
    url: 'system/Role',
    menus: null
  },
  {
    menuid: 74,
    icon: 'icon-cs-manage',
    menuname: '',
    hasThird: 'N',
    url: 'system/Dept',
    menus: null
  },
  {
    menuid: 75,
    icon: 'icon-promotion-manage',
    menuname: '',
    hasThird: 'N',
    url: 'system/Variable',
    menus: null
  },
  {
    menuid: 76,
    icon: 'icon-cms-manage',
    menuname: '',
    hasThird: 'N',
    url: 'system/Permission',
    menus: null
  }
]
},
{
  menuid: 128,
  icon: 'li-icon-shangchengxitongtubiaozitihuayuanwenjian91',
  menuname: '',
  hasThird: null,
  url: null,
  menus: [
    {
      menuid: 129,
      icon: 'icon-provider-manage',
```

```

        menuname: "",
        hasThird: 'N',
        url: 'machine/MachineConfig',
        menus: null
    },
    {
        menuid: 175,
        icon: 'icon-provider-manage',
        menuname: "",
        hasThird: 'N',
        url: 'pay/Config',
        menus: null
    }
]
},
{
    menuid: 150,
    icon: 'li-icon-shangchengxitongtubiaozitihuayuanwenjian91',
    menuname: "",
    hasThird: null,
    url: null,
    menus: [
        {
            menuid: 159,
            icon: 'icon-provider-manage',
            menuname: "",
            hasThird: 'N',
            url: 'charts/statistics',
            menus: null
        }
    ]
}
],
msg: 'success'
}

this.allmenu = res.data

// menu(localStorage.getItem('logintoken'))
// .then(res => {
//     console.log(JSON.stringify(res))
//     if (res.success) {
//         this.allmenu = res.data

```

```

//   } else {
//     this.$message.error(res.msg)
//     return false
//   }
// })
// .catch(err => {
//   this.$message.error("")
// })
//
this.$root.Bus.$on('toggle', value => {
  this.collapsed = !value
})
}
}
</script>
<style>
.el-menu-vertical-demo:not(.el-menu--collapse) {
  width: 240px;
  min-height: 400px;
}
.el-menu-vertical-demo:not(.el-menu--collapse) {
  border: none;
  text-align: left;
}
.el-menu-item-group__title {
  padding: 0px;
}
.el-menu-bg {
  background-color: #1f2d3d !important;
}
.el-menu {
  border: none;
}
.logobox {
  height: 40px;
  line-height: 40px;
  color: #9d9d9d;
  font-size: 20px;
  text-align: center;
  padding: 20px 0px;
}
.logoimg {

```

```
    height: 40px;
  }
</style>
```

7:F:\git\nodejs\vue\vuestep\vueproject\src\components\navcon.vue

```
/**
 *
 */
<template>
  <el-menu class="el-menu-demo" mode="horizontal" background-color="#334157" text-
color="#fff" active-text-color="#fff">
    <el-button class="buttonimg">
      
    </el-button>
    <el-submenu index="2" class="submenu">
      <!-- <template slot="title">{{user.userRealName}}</template> -->
      <template slot="title"></template>
      <el-menu-item index="2-1"></el-menu-item>
      <el-menu-item @click="content()" index="2-2"></el-menu-item>
      <el-menu-item @click="exit()" index="2-3"></el-menu-item>
    </el-submenu>
  </el-menu>
</template>
<script>
import { loginout } from '../api/userMG'
export default {
  name: 'navcon',
  data() {
    return {
      collapsed: true,
      imgshow: require('../assets/img/show.png'),
      imgsq: require('../assets/img/sq.png'),
      user: {}
    }
  },
  // ()
  created() {
    this.user = JSON.parse(localStorage.getItem('userdata'))
  },
  methods: {
    //
    exit() {
```

```

this.$confirm(' ', '?', '', {
  confirmButtonText: '',
  cancelButtonText: '',
  type: 'warning'
})
.then(() => {
  setTimeout(() => {
    this.$store.commit('logout', 'false')
    this.$router.push({ path: '/login' })
    this.$message({
      type: 'success',
      message: '!'
    })
  }, 1000)
  // loginout()
  // .then(res => {
  //   if (res.success) {
  //     //2
  //     setTimeout(() => {
  //       this.$store.commit('logout', 'false')
  //       this.$router.push({ path: '/login' })
  //       this.$message({
  //         type: 'success',
  //         message: '!'
  //       })
  //     }, 1000)
  //   } else {
  //     this.$message.error(res.msg)
  //     this.logining = false
  //     return false
  //   }
  // })
  // .catch(err => {
  //   //
  //   this.getcode()
  //   this.logining = false
  //   this.$message.error('')
  // })
})
.catch(() => {
  this.$message({
    type: 'info',

```

```

        message: "
      })
    })
  },
  //
  toggle(showtype) {
    this.collapsed = !showtype
    this.$root.Bus.$emit('toggle', this.collapsed)
  }
}
}
</script>
<style scoped>
.el-menu-vertical-demo:not(.el-menu--collapse) {
  border: none;
}
.submenu {
  float: right;
}
.buttonimg {
  height: 60px;
  background-color: transparent;
  border: none;
}
.showimg {
  width: 26px;
  height: 26px;
  position: absolute;
  top: 17px;
  left: 17px;
}
.showimg:active {
  border: none;
}
</style>

```

8:F:\git\nodejs\vue\vuestep\vueproject\src\components\Pagination.vue

/**

*

*/

<template>

<el-pagination class="page-box" @size-change="handleSizeChange" @current-


```

change="handleCurrentChange" background :current-page="childMsg.currentPage" :page-
sizes="[10, 20, 30, 40]" :page-size="childMsg.pageSize" layout="total, sizes, prev, pager, next,
jumper" :total="childMsg.total">
  </el-pagination>
</template>
<script>
export default {
  name: 'Pagination',
  props: ['childMsg'],
  data() {
    return {
      pageparm: {
        currentPage: this.childMsg.currentPage,
        pageSize: this.childMsg.pageSize
      }
    }
  },
  created() {},
  methods: {
    handleSizeChange(val) {
      /**
       *
       * 1
       * 2
       */
      this.pageparm.pageSize = val
      this.$emit('callFather', this.pageparm)
    },
    handleCurrentChange(val) {
      /**
       *
       * 1
       * 2
       */
      this.pageparm.currentPage = val
      this.$emit('callFather', this.pageparm)
    }
  }
}
</script>

<style>

```

```
.page-box {  
  margin: 10px auto;  
}  
</style>
```

9:F:\git\nodejs\vue\vuestep\vueproject\src\components\template.vue

```
/**  
*  
**/
```

```
<template>  
  <div>  
  
  </div>  
</template>
```

```
<script>  
//  
import HeaderComponent from "../head.vue";  
//  
export default {  
  //  
  name: "template",  
  //  
  data() {  
    //  
    return {};  
  },  
  //  
  components: {  
    HeaderComponent  
  },  
  //  
  watch: {},  
  //  
  methods: {  
    addFun() {},  
    submitFun() {}  
  },  
  // ()  
  beforeCreate() {},  
  // ()
```

```

    created() {},
    // ()
    beforeMount() {},
    // ()
    mounted() {},
    // ()
    beforeUpdate() {},
    // ()
    updated() {},
    // ()
    beforeDestroy() {},
    // ()
    destroyed() {}
};
</script>
// scoped
<style scoped>
/**
 * css
 * @import "../assets/css/components/index.css";
 */
</style>

```

```

10:F:\git\nodejs\vue\vuestep\vueproject\src\main.js
// The Vue build version to load with the `import` command
// (runtime-only or standalone) has been set in webpack.base.conf with an alias.
import Vue from 'vue';
// element UI
import ElementUI from 'element-ui';
import 'element-ui/lib/theme-chalk/index.css';
import App from './App';
//
import router from './router';
//
import store from './vuex/store';
// icon
import './assets/icon/iconfont.css'
//

// echarts
import echarts from 'echarts'
Vue.prototype.$echarts = echarts

```

```
import axios from 'axios';
Vue.prototype.$axios = axios;
```

```
Vue.config.productionTip = false;
```

```
// element UI
```

```
Vue.use(ElementUI);
```

```
//
```

```
import * as custom from './utils/util'
```

```
Object.keys(custom).forEach(key => {
  Vue.filter(key, custom[key])
})
```

```
//
```

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

```
  if (to.matched.length !== 0) {
```

```
    if (to.meta.requireAuth) { //
```

```
      if (Boolean(localStorage.getItem("userInfo"))) { // vuex stateuser
        next();
```

```
      } else {
```

```
        next({
```

```
          path: '/login',
```

```
          query: { redirect: to.fullPath } // path
```

```
        })
```

```
      }
```

```
    } else {
```

```
      if (Boolean(localStorage.getItem("userInfo"))) { //
```

```
        if (to.path !== "/" && to.path !== "/login") { //
```

```
          next();
```

```
        } else {
```

```
          /**
```

```
          *
```

```
          */
```

```
          next({
```

```
            path: '/goods/Goods'
```

```
          })
```

```
        }
```

```
      } else {
```

```
        next();
```

```
    }
```

```

    }
  } else {
    next({
      path: '/login',
      query: { redirect: to.fullPath } // path
    })
  }
})

```

```

/* eslint-disable no-new */

```

```

new Vue({
  el: '#app',
  router,
  store, //store vuex
  components: { App },
  template: '<App/>',
  data: {
    //
    Bus: new Vue()
  }
}

```

```

})

```

```

11:F:\git\nodejs\vue\vuestep\vueproject\src\router\index.js

```

```

//
import Vue from 'vue';
import Router from 'vue-router';
//
import login from '@/views/login';
//
import index from '@/views/index';
/**
 *
 */
//
import Goods from '@/views/goods/Goods';
//
import Machine from '@/views/machine/Machine';
//
import MachineAisle from '@/views/machine/MachineAisle';
/**
 *

```

```
*/
//
import Order from '@views/pay/Order';
/**
 *
 */
//
import user from '@views/system/user';
//
import Module from '@views/system/Module';
//
import Role from '@views/system/Role';
//
import Dept from '@views/system/Dept';
//
import Variable from '@views/system/Variable';
//
import Permission from '@views/system/Permission';
/**
 *
 */
//
import MachineConfig from '@views/machine/MachineConfig';
//
import Config from '@views/pay/Config';
/**
 *
 */
//
import druidLogin from '@views/druid/login';

//
import statistics from '@views/charts/statistics';

//
Vue.use(Router);

//
export default new Router({
  routes: [{
    path: '/',
    name: "",
```

```
    component: login,
    hidden: true,
    meta: {
      requireAuth: false
    }
  }, {
    path: '/login',
    name: "",
    component: login,
    hidden: true,
    meta: {
      requireAuth: false
    }
  }, {
    path: '/index',
    name: "",
    component: index,
    iconCls: 'el-icon-tickets',
    children: [{
      path: '/goods/Goods',
      name: "",
      component: Goods,
      meta: {
        requireAuth: true
      }
    }], {
      path: '/machine/Machine',
      name: "",
      component: Machine,
      meta: {
        requireAuth: true
      }
    }, {
      path: '/machine/MachineAisle',
      name: "",
      component: MachineAisle,
      meta: {
        requireAuth: true
      }
    }, {
      path: '/pay/Order',
      name: "",
```

```
    component: Order,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/user',
    name: "",
    component: user,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/Module',
    name: "",
    component: Module,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/Role',
    name: "",
    component: Role,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/Dept',
    name: "",
    component: Dept,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/Variable',
    name: "",
    component: Variable,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/system/Permission',
    name: "",
```



```

    component: Permission,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/machine/MachineConfig',
    name: '',
    component: MachineConfig,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/pay/Config',
    name: '',
    component: Config,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/druid/login',
    name: '',
    component: druidLogin,
    meta: {
      requireAuth: true
    }
  }, {
    path: '/charts/statistics',
    name: '',
    component: statistics,
    meta: {
      requireAuth: true
    }
  }
]
})

```

12:F:\git\nodejs\vue\vuestep\vueproject\src\utils\util.js

```
/**
```

```
*
```

```
* @param {*} timestamp
```

```
*/
```

```
const timestampToTime = (timestamp) => {
```

```

let date = new Date(timestamp) //10*1000131000
let Y = date.getFullYear() + '-'
let M =
  (date.getMonth() + 1 < 10 ?
    '0' + (date.getMonth() + 1) :
    date.getMonth() + 1) + '-'
let D =
  (date.getDate() < 10 ? '0' + date.getDate() : date.getDate()) + ' '
let h =
  (date.getHours() < 10 ? '0' + date.getHours() : date.getHours()) + ':'
let m =
  (date.getMinutes() < 10 ? '0' + date.getMinutes() : date.getMinutes()) +
  ':'
let s =
  date.getSeconds() < 10 ? '0' + date.getSeconds() : date.getSeconds()
return Y + M + D + h + m + s
};
/**
 * localStorage
 */
const setStore = (name, content) => {
  if (!name) return;
  if (typeof content !== 'string') {
    content = JSON.stringify(content);
  }
  window.localStorage.setItem(name, content);
}

/**
 * localStorage
 */
const getStore = name => {
  if (!name) return;
  return window.localStorage.getItem(name);
}

/**
 * localStorage
 */
const removeStore = name => {
  if (!name) return;
  window.localStorage.removeItem(name);
}

```

```
}

/**
 * cookie
 **/
function setCookie(name, value, day) {
    let date = new Date();
    date.setDate(date.getDate() + day);
    document.cookie = name + '=' + value + ';expires=' + date;
};
```

```
/**
 * cookie
 **/
function getCookie(name) {
    let reg = RegExp(name + '=(^;]+)');
    let arr = document.cookie.match(reg);
    if (arr) {
        return arr[1];
    } else {
        return "";
    }
};
```

```
/**
 * cookie
 **/
function delCookie(name) {
    setCookie(name, null, -1);
};
```

```
/**
 *
 **/
export {
    timestampToTime,
    setStore,
    getStore,
    removeStore,
    setCookie,
    getCookie,
    delCookie
```

```
}
```

```
13:F:\git\nodejs\vue\vuestep\vueproject\src\views\charts\statistics.vue
```

```
/**
```

```
*
```

```
*/
```

```
<template>
```

```
<!-- -->
```

```
<div class="stbox">
```

```
<!-- -->
```

```
<el-breadcrumb separator-class="el-icon-arrow-right">
```

```
<el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
```

```
<el-breadcrumb-item></el-breadcrumb-item>
```

```
</el-breadcrumb>
```

```
<!-- -->
```

```
<el-row :gutter="23">
```

```
<el-col :span="18">
```

```
<div class="stbgc">
```

```
<el-row :gutter="23">
```

```
<el-col :span="7">
```

```
<el-input size="small" v-model="machineNo" placeholder=""></el-input>
```

```
</el-col>
```

```
<el-col :span="7">
```

```
<el-input size="small" v-model="machineNo" placeholder=""></el-input>
```

```
</el-col>
```

```
<el-col :span="7">
```

```
<el-input size="small" v-model="machineNo" placeholder=""></el-input>
```

```
</el-col>
```

```
<el-col :span="3" class="stsearch">
```

```
<el-button size="small" type="primary"></el-button>
```

```
</el-col>
```

```
</el-row>
```

```
</div>
```

```
</el-col>
```

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

```
<div class="stbgc">
```

```
<el-row>
```

```
<el-col :span="8" class="text-c">
```

```
<el-radio v-model="type" label="day"></el-radio>
```

```
</el-col>
```

```
<el-col :span="8" class="text-c">
```

```
<el-radio v-model="type" label="month"></el-radio>
```

```

    </el-col>
    <el-col :span="8" class="text-c">
      <el-radio v-model="type" label="years"></el-radio>
    </el-col>
  </el-row>
</div>
</el-col>
</el-row>
<!-- -->
<el-row :gutter="23">
  <el-col :span="8" class="text-c">
    <div class="st-gbox">
      <div class="cavasbox" ref="SCEchart"></div>
    </div>
  </el-col>
  <el-col :span="8" class="text-c">
    <div class="st-gbox">
      <div class="cavasbox" ref="SUMEchart"></div>
    </div>
  </el-col>
  <el-col :span="8" class="text-c">
    <div class="st-gbox">
      <div class="cavasbox" ref="ClickEchart"></div>
    </div>
  </el-col>
</el-row>
<!-- -->
<div>
  <el-row :gutter="23">
    <el-col :span="12" class="text-c">
      <div class="paybox">
        <div class="cavasbox" ref="payEchart"></div>
      </div>
    </el-col>
    <el-col :span="12" class="text-c">
      <div class="paybox">
        <div class="cavasbox" ref="payNumEchart"></div>
      </div>
    </el-col>
  </el-row>
</div>
</div>

```

```
</template>
<script type="text/ecmascript-6">
import Chart from 'echarts'
export default {
  name: "welcome",
  data() {
    return {
      machineNo: "",
      type: 'day',
      //
      SCEoption: {
        tooltip: {
          trigger: 'item',
          formatter: "{a} <br/>{b} : {c}"
        },
        legend: {
          data: [{
            name: "",
            icon: 'rect'
          }],
          top: 1,
          left: 1,
          itemGap: 10,
          itemWidth: 12,
          itemHeight: 12,
          textStyle: {
            fontSize: 12,
            color: "#323232"
          }
        },
        grid: {
          left: 50,
          right: 10,
          top: 30,
          bottom: 30,
          borderWidth: 1
        },
        xAxis: {
          type: 'category',
          data: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'],
          axisLine: {
            lineStyle: {
```

```

        color: "#999999",
        width: 1
    }
},
axisLabel: {
    margin: 14,
    height: 70,
    interval: 0,
    textStyle: {
        fontSize: 10,
        color: "#999999"
    }
}
},
yAxis: {
    type: 'value',
    axisLine: {
        lineStyle: {
            color: "#999999",
            width: 1
        }
    },
    axisLabel: {
        margin: 14,
        textStyle: {
            fontSize: 10,
            color: "#999999"
        }
    }
},
series: [{
    name: "",
    type: 'bar',
    barGap: 0,
    data: [50000, 70000, 80000, 40000, 50000, 30000, 40000, 60000, 50000, 40000, 60000,
40000],
    barWidth: 10,
    itemStyle: {
        normal: {
            color: new Chart.graphic.LinearGradient(
                0, 0, 0, 1,
                [

```

```

        { offset: 0, color: '#83bff6' },
        { offset: 0.5, color: '#188df0' },
        { offset: 1, color: '#188df0' }
    ]
)
},
emphasis: {
    color: new Chart.graphic.LinearGradient(
        0, 0, 0, 1,
        [
            { offset: 0, color: '#2378f7' },
            { offset: 0.7, color: '#2378f7' },
            { offset: 1, color: '#83bff6' }
        ]
    )
}
}
}]
},
//
SUMoption: {
    tooltip: {
        trigger: 'item',
        formatter: "{a} <br/>{b} : {c}"
    },
    legend: {
        data: [{
            name: "",
            icon: 'rect'
        }],
        top: 1,
        left: 1,
        itemGap: 10,
        itemWidth: 12,
        itemHeight: 12,
        textStyle: {
            fontSize: 12,
            color: "#323232"
        }
    },
    grid: {
        left: 50,

```



```
right: 10,
top: 30,
bottom: 30,
borderWidth: 1
},
xAxis: {
  type: 'category',
  data: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'],
  axisLine: {
    lineStyle: {
      color: "#999999",
      width: 1
    }
  },
  axisLabel: {
    margin: 14,
    height: 70,
    interval: 0,
    textStyle: {
      fontSize: 10,
      color: "#999999"
    }
  }
},
yAxis: {
  type: 'value',
  axisLine: {
    lineStyle: {
      color: "#999999",
      width: 1
    }
  },
  axisLabel: {
    margin: 14,
    textStyle: {
      fontSize: 10,
      color: "#999999"
    }
  }
},
series: [{
  name: "",
```

```

    // type: 'bar',
    type: 'line',
    barGap: 0,
    data: [50000, 70000, 80000, 40000, 50000, 30000, 40000, 60000, 50000, 40000, 60000,
40000],
    barWidth: 10,
    itemStyle: {
        color: "#108ff9"
    }
}]
},
//
Clickoption: {
    tooltip: {
        trigger: 'item',
        formatter: "{a} <br/>{b} : {c}"
    },
    legend: {
        data: [{
            name: "",
            icon: 'rect'
        }],
        top: 1,
        left: 1,
        itemGap: 10,
        itemWidth: 12,
        itemHeight: 12,
        textStyle: {
            fontSize: 12,
            color: "#323232"
        }
    },
    grid: {
        left: 50,
        right: 10,
        top: 30,
        bottom: 30,
        borderWidth: 1
    },
    xAxis: {
        type: 'category',
        data: ['1', '2', '3', '4', '5', '6', '7', '8', '9', '10', '11', '12'],

```

```
axisLine: {
  lineStyle: {
    color: "#999999",
    width: 1
  }
},
axisLabel: {
  margin: 14,
  height: 70,
  interval: 0,
  textStyle: {
    fontSize: 10,
    color: "#999999"
  }
},
yAxis: {
  type: 'value',
  axisLine: {
    lineStyle: {
      color: "#999999",
      width: 1
    }
  },
  axisLabel: {
    margin: 14,
    textStyle: {
      fontSize: 10,
      color: "#999999"
    }
  }
},
series: [{
  name: "",
  type: 'bar',
  barGap: 0,
  data: [50000, 10000, 80000, 30000, 50000, 60000, 40000, 80000, 50000, 20000, 60000,
40000],
  barWidth: 10,
  itemStyle: {
    color: "#48cefd"
  }
}
```

```
    ]
  },
  //
  payoption: {
    backgroundColor: '#2c343c',
    title: {
      text: '()',
      left: 10,
      top: 5,
      textStyle: {
        fontSize: 12,
        color: '#ccc'
      }
    },
  },

  tooltip: {
    trigger: 'item',
    formatter: "{a} <br/>{b} : {c} ({d}%)"
  },

  visualMap: {
    show: false,
    min: 80,
    max: 600,
    inRange: {
      colorLightness: [0, 1]
    }
  },
  series: [
    {
      name: '()',
      type: 'pie',
      radius: '55%',
      center: ['50%', '50%'],
      data: [
        { value: 335, name: " " },
        { value: 310, name: " " },
        { value: 274, name: " " },
        { value: 235, name: " " },
        { value: 100, name: 'Pos' }
      ],
      sort: function (a, b) { return a.value - b.value; },
      roseType: 'radius',
    }
  ]
}
```

```

label: {
  normal: {
    textStyle: {
      color: 'rgba(255, 255, 255, 0.3)'
    }
  },
  labelLine: {
    normal: {
      lineStyle: {
        color: 'rgba(255, 255, 255, 0.3)'
      },
      smooth: 0.2,
      length: 10,
      length2: 20
    }
  },
  itemStyle: {
    normal: {
      color: '#c23531',
      shadowBlur: 200,
      shadowColor: 'rgba(0, 0, 0, 0.5)'
    }
  },
  animationType: 'scale',
  animationEasing: 'elasticOut',
  animationDelay: function (idx) {
    return Math.random() * 200;
  }
},
payNumoption: {
  backgroundColor: '#2c343c',
  title: {
    text: '()',
    left: 10,
    top: 5,
    textStyle: {
      fontSize: 12,
      color: '#ccc'
    }
  }
}

```

```
}  
},
```

```
tooltip: {  
  trigger: 'item',  
  formatter: "{a} <br/>{b} : {c} ({d}%)"  
},
```

```
visualMap: {  
  show: false,  
  min: 80,  
  max: 600,  
  inRange: {  
    colorLightness: [0, 1]  
  }  
},
```

```
series: [  
  {  
    name: '()',  
    type: 'pie',  
    radius: '55%',  
    center: ['50%', '50%'],  
    data: [  
      { value: 335, name: " },  
      { value: 310, name: " },  
      { value: 274, name: " },  
      { value: 235, name: " },  
      { value: 100, name: 'Pos' }  
    ].sort(function (a, b) { return a.value - b.value; } ),  
    roseType: 'radius',  
    label: {  
      normal: {  
        textStyle: {  
          color: 'rgba(255, 255, 255, 0.3)'  
        }  
      }  
    },  
    labelLine: {  
      normal: {  
        lineStyle: {  
          color: 'rgba(255, 255, 255, 0.3)'  
        }  
      },
```

```

        smooth: 0.2,
        length: 10,
        length2: 20
    }
},
itemStyle: {
    normal: {
        color: '#c23531',
        shadowBlur: 200,
        shadowColor: 'rgba(0, 0, 0, 0.5)'
    }
},

    animationType: 'scale',
    animationEasing: 'elasticOut',
    animationDelay: function (idx) {
        return Math.random() * 200;
    }
}
]
},
}
},
//
components: {
    //
},
// ()
created() { },
// ()
mounted() {
    this.getSCE()
    this.getSUM()
    this.getClick()
    this.getpay()
    this.getpayNum()
},
//
methods: {
    //
    getSCE() {
        this.chart = Chart.init(this.$refs.SCEchart)
    }
}

```

```

        this.chart.setOption(this.SCEoption)
    },
    //
    getSUM() {
        this.chart = Chart.init(this.$refs.SUMEchart)
        this.chart.setOption(this.SUMoption)
    },
    //
    getClick() {
        this.chart = Chart.init(this.$refs.ClickEchart)
        this.chart.setOption(this.Clickoption)
    },
    //
    getpay() {
        this.chart = Chart.init(this.$refs.payEchart)
        this.chart.setOption(this.payoption)
    },
    //
    getpayNum() {
        this.chart = Chart.init(this.$refs.payNumEchart)
        this.chart.setOption(this.payNumoption)
    }

}

};
</script>
<style>
.stbox {
    width: 100%;
    height: 100%;
    box-sizing: border-box;
}
.stbgc {
    background-color: #fff;
    height: 60px;
    line-height: 60px;
    border-radius: 5px;
    padding: 0px 16px;
}
.stsearch {
    text-align: center;
}

```



```
.text-c {
  text-align: center;
}
.st-gbox {
  background-color: #fff;
  margin-top: 20px;
  border-radius: 5px;
  height: 30vh;
  box-sizing: border-box;
  padding: 10px;
}
.cavasbox {
  box-sizing: border-box;
  width: 100%;
  height: 100%;
}
.paybox {
  width: 100%;
  background-color: #fff;
  box-sizing: border-box;
  border-radius: 5px;
  margin-top: 20px;
  height: 32vh;
}
</style>
```

14:F:\git\nodejs\vue\vuestep\vueproject\src\views\druid\login.vue

```
/**
 *
 */
```

```
<template>
  <div> </div>
</template>
```

```
<script>
export default {
```

```
}
</script>
```

```
<style>
```

</style>

15:F:\git\nodejs\vue\vuestep\vueproject\src\views\goods\Goods.vue

/**

*

*/

<template>

<div>

<!-- -->

<el-breadcrumb separator-class="el-icon-arrow-right">

<el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>

<el-breadcrumb-item></el-breadcrumb-item>

</el-breadcrumb>

<!-- -->

<el-form :inline="true" :model="formInline" class="user-search">

<el-form-item label="">

<el-input size="small" v-model="formInline.deptName" placeholder=""></el-input>

</el-form-item>

<el-form-item label="">

<el-input size="small" v-model="formInline.deptNo" placeholder=""></el-input>

</el-form-item>

<el-form-item>

<el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>

<el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>

</el-form-item>

</el-form>

<!-->

<el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-loading-text="" style="width: 100%;">

<el-table-column align="center" type="selection" width="60">

</el-table-column>

<el-table-column sortable prop="deptName" label="" width="300">

</el-table-column>

<el-table-column sortable prop="deptNo" label="" width="300">

</el-table-column>

<el-table-column sortable prop="editTime" label="" width="300">

<template slot-scope="scope">

<div>{{scope.row.editTime|timestampToTime}}</div>

</template>

</el-table-column>

```

<el-table-column sortable prop="editUser" label="" width="300">
</el-table-column>
<el-table-column align="center" label="" min-width="300">
  <template slot-scope="scope">
    <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
    <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click="closeDialog">
  <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
    <el-form-item label="" prop="deptName">
      <el-input size="small" v-model="editForm.deptName" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="deptNo">
      <el-input size="small" v-model="editForm.deptNo" auto-complete="off" placeholder=""></el-
input>
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button size="small" @click="closeDialog"></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
  </div>
</el-dialog>
</div>
</template>

```

```

<script>
import { deptList, deptSave, deptDelete } from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //

```

```

editFormVisible: false, //
title: "",
editForm: {
  deptId: "",
  deptName: "",
  deptNo: "",
  token: localStorage.getItem('logintoken')
},
// rules
rules: {
  deptName: [
    { required: true, message: "", trigger: 'blur' }
  ],
  deptNo: [{ required: true, message: "", trigger: 'blur' }]
},
formInline: {
  page: 1,
  limit: 10,
  varLable: "",
  varName: "",
  token: localStorage.getItem('logintoken')
},
//
seletedata: {
  ids: "",
  token: localStorage.getItem('logintoken')
},
userparm: [], //
listData: [], //
//
pageparm: {
  currentPage: 1,
  pageSize: 10,
  total: 10
}
}
},
//
components: {
  Pagination
},
/**

```

```

*
*/

/**
*
*/
created() {
  this.getdata(this.formInline)
},

/**
*
*/
methods: {
  //
  getdata(parameter) {
    this.loading = true
    //
    let res = {
      code: 0,
      msg: null,
      count: 5,
      data: [
        {
          addUser: null,
          editUser: null,
          addTime: 1521062371000,
          editTime: 1526700200000,
          deptId: 2,
          deptName: 'XX',
          deptNo: '1',
          parentId: 1
        },
        {
          addUser: null,
          editUser: null,
          addTime: 1521063247000,
          editTime: 1526652291000,
          deptId: 3,
          deptName: '',
          deptNo: '02',
          parentId: 1
        }
      ]
    }
  }
}

```

```

    },
    {
      addUser: null,
      editUser: null,
      addTime: 1526349555000,
      editTime: 1526349565000,
      deptId: 12,
      deptName: '1',
      deptNo: '11',
      parentId: 1
    },
    {
      addUser: null,
      editUser: null,
      addTime: 1526373178000,
      editTime: 1526373178000,
      deptId: 13,
      deptName: '5',
      deptNo: '5',
      parentId: 1
    },
    {
      addUser: null,
      editUser: null,
      addTime: 1526453107000,
      editTime: 1526453107000,
      deptId: 17,
      deptName: 'v',
      deptNo: 'v',
      parentId: 1
    }
  ]
}

this.loading = false
this.listData = res.data
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/****
*
```

```

*/
// deptList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page
//     this.pageparm.pageSize = this.formInline.limit
//     this.pageparm.total = res.count
//   }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = "
    this.editForm.deptId = row.deptId
    this.editForm.deptName = row.deptName
    this.editForm.deptNo = row.deptNo
  } else {
    this.title = "

```

```

    this.editForm.deptId = "
    this.editForm.deptName = "
    this.editForm.deptNo = "
  }
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      deptSave(this.editForm)
        .then(res => {
          this.editFormVisible = false
          this.loading = false
          if (res.success) {
            this.getdata(this.formInline)
            this.$message({
              type: 'success',
              message: "
            })
          } else {
            this.$message({
              type: 'info',
              message: res.msg
            })
          }
        })
    }
    .catch(err => {
      this.editFormVisible = false
      this.loading = false
      this.$message.error("")
    })
  } else {
    return false
  }
})
},
//
deleteUser(index, row) {
  this.$confirm('?', "", {
    confirmButtonText: "",
    cancelButtonText: "",
    type: 'warning'
  })
}

```



```

    })
    .then(() => {
        deptDelete(row.deptId)
        .then(res => {
            if (res.success) {
                this.$message({
                    type: 'success',
                    message: '!'
                })
                this.getdata(this.formInline)
            } else {
                this.$message({
                    type: 'info',
                    message: res.msg
                })
            }
        })
    })
    .catch(err => {
        this.loading = false
        this.$message.error("")
    })
})
.catch(() => {
    this.$message({
        type: 'info',
        message: ""
    })
})
},
//
closeDialog() {
    this.editFormVisible = false
}
}
}
</script>

```

```

<style scoped>
.user-search {
    margin-top: 20px;
}
.userRole {

```

```
width: 100%;  
}  
</style>
```

16:F:\git\nodejs\vue\vuestep\vueproject\src\views\index.vue

```
<template>  
  <!-- <el-container class="index-con">  
    <el-header class="index-header">  
      <navcon></navcon>  
    </el-header>  
    <el-container class="index-con">  
      <el-aside :class="showclass">  
        <leftnav></leftnav>  
      </el-aside>  
      <el-container class="main-con">  
        <el-main class="index-main">  
          <router-view></router-view>  
        </el-main>  
      </el-container>  
    </el-container>  
  </el-container> -->
```

```
<el-container class="index-con">  
  <el-aside :class="showclass">  
    <leftnav></leftnav>  
  </el-aside>  
  <el-container class="main-con">  
    <el-header class="index-header">  
      <navcon></navcon>  
    </el-header>  
    <el-main class="index-main">  
      <router-view></router-view>  
    </el-main>  
  </el-container>  
</el-container>  
</template>  
<script>  
  //  
  import navcon from '../components/navcon.vue'
```

```
import leftnav from '../components/leftnav.vue'
```

```
export default {
```

```
  name: 'index',
```

```
  data() {
```

```
    return {
```

```
      showclass: 'asideshow',
```

```
      showtype: false
```

```
    }
```

```
  },
```

```
  //
```

```
  components: {
```

```
    navcon,
```

```
    leftnav
```

```
  },
```

```
  methods: {},
```

```
  created() {
```

```
    //
```

```
    this.$root.Bus.$on('toggle', value => {
```

```
      if (value) {
```

```
        this.showclass = 'asideshow'
```

```
      } else {
```

```
        setTimeout(() => {
```

```
          this.showclass = 'aside'
```

```
        }, 300)
```

```
      }
```

```
    })
```

```
  },
```

```
  beforeUpdate() {},
```

```
  // ()
```

```
  beforeMount() {
```

```
    //
```

```
    this.$message({
```

```
      message: "",
```

```
      type: 'success'
```

```
    })
```

```
  }
```

```
}
```

```
</script>
```

```
<style >
```

```
.index-con {
```

```
  height: 100%;
```

```
  width: 100%;
```

```

    box-sizing: border-box;
}

.aside {
  width: 64px !important;
  height: 100%;
  background-color: #334157;
  margin: 0px;
}

.asideshow {
  width: 240px !important;
  height: 100%;
  background-color: #334157;
  margin: 0px;
}

.index-header,
.index-main {
  padding: 0px;
  border-left: 2px solid #333;
}
</style>

```

17:F:\git\nodejs\vue\vuestep\vueproject\src\views\login.vue

```

<template>
  <div class="login-wrap">
    <el-form label-position="left" :model="ruleForm" :rules="rules" ref="ruleForm" label-width="0px"
class="demo-ruleForm login-container">
      <h3 class="title"></h3>
      <el-form-item prop="username">
        <el-input type="text" v-model="ruleForm.username" auto-complete="off" placeholder=""></el-
input>
      </el-form-item>
      <el-form-item prop="password">
        <el-input type="password" v-model="ruleForm.password" auto-complete="off" placeholder="
"></el-input>
      </el-form-item>
      <el-row>
        <el-col :span="12">
          <el-form-item prop="code">
            <el-input type="text" v-model="ruleForm.code" auto-complete="off" placeholder=""
@keyup.enter.native="submitForm('ruleForm')"></el-input>
          </el-form-item>

```

```

    </el-col>
    <el-col :span="12" class="code-box">
      
    </el-col>
  </el-row>
  <el-checkbox class="remember" v-model="rememberpwd"></el-checkbox>
  <el-form-item style="width:100%;">
    <el-button type="primary" style="width:100%;" @click="submitForm('ruleForm')"
:loading="logining"></el-button>
  </el-form-item>
</el-form>
</div>
</template>
<script type="text/ecmascript-6">
import { login } from '../api/userMG'
import { setCookie, getCookie, delCookie } from '../utils/util'
import md5 from 'js-md5'
export default {
  name: 'login',
  data() {
    return {
      //loadingfalse
      logining: false,
      //
      rememberpwd: false,
      ruleForm: {
        //usernamepassword
        username: "",
        password: "",
        code: "",
        randomStr: "",
        codeimg: ""
      },
      //rules
      rules: {
        username: [{ required: true, message: "", trigger: 'blur' }],
        password: [{ required: true, message: "", trigger: 'blur' }],
        code: [{ required: true, message: "", trigger: 'blur' }]
      }
    }
  },
  // ()

```

```

created() {
  this.$message({
    message: "",
    type: 'success'
  })
  //
  this.getcode()
  //
  this.getuserpwd()

},
//
methods: {
  //
  getuserpwd() {
    if (getCookie('user') != "" && getCookie('pwd') != "") {
      this.ruleForm.username = getCookie('user')
      this.ruleForm.password = getCookie('pwd')
      this.rememberpwd = true
    }
  },
  //info
  submitForm(formName) {
    this.$refs[formName].validate(valid => {
      if (valid) {
        this.logining = true
        //
        setTimeout(() => {
          this.logining = false
          this.$store.commit('login', 'true')
          this.$router.push({ path: '/goods/Goods' })
        }, 1000)
        //
        // login(this.ruleForm).then(res => {
        //   if (res.success) {
        //     if (this.rememberpwd) {
        //       //cookie7
        //       setCookie('user', this.ruleForm.username, 7)
        //       //cookie7
        //       setCookie('pwd', this.ruleForm.password, 7)
        //     } else {
        //       delCookie('user')

```


EtLSBHZW5lcmF0b3I6IFNrZXRjaCA0Ni4yICg0NDQ5NikgLSBodHRwOi8vd3d3LmJvaGVtaWFuY
29kaW5nLnNvbS9za2V0Y2ggLS0+CiAglCA8dGI0bGU+R3JvdXAglMjE8L3RpdGxIPgogICAglPGR
lc2M+Q3JIYXRIZCB3aXRoIFNrZXRjaC48L2Ric2M+CiAglCA8ZGVmcz48L2RIZnM+CiAglCA8ZyB
pZD0iQW50LURlc2lnbi1Qcm8tMy4wliBzdHJva2U9Im5vbmUiIHNoem9rZS13aWR0aD0iMSlgZmls
bD0ibm9uZSIgZmlsbC1ydWxlPSJldmVub2Rklj4KICAgICAglCA8ZyBpZD0i6LSm5oi35a+G56CB5
5m75b2VLeagoemqjClgdHJhbnNmb3JtPSJ0cmFuc2xhdGUoLTc5LjAwMDAwMCwgLTgyLjAwMD
AwMCkiPgogICAglCAglCAglCA8ZyBpZD0iR3JvdXAAtMjEilHRyYW5zZm9ybT0idHJhbnNsYXRlKD
c3LjAwMDAwMCwgNzMumMDAwMDAwKSI+CiAglCAglCAglCAglCAglCA8ZyBpZD0iR3JvdXAAtMT
gilG9wYWNpdHk9ljAuOCIGdHJhbnNmb3JtPSJ0cmFuc2xhdGUoNzQuOTAxNDE2LCA1NjkuNjk5
MTU4KSByb3Rh dGUoLTcuMDAwMDAwKSB0cmFuc2xhdGUoLTc0LjkwMTQxNiwiLTU2OS42O
TkxNTgplHRyYW5zbGF0ZSg0LjkwMTQxNiwiNTI1LjE5OTE1OCKiPgogICAglCAglCAglCAglCAgl
CAglDXlbGxpcHNlIGlkPSJPdmFsLTE4LiBmaWxsPSIjQ0ZEQUU2liBvcGFjaXR5PSlwLjllBiJeD0iN
jMuNTc0ODc5MilgY3k9ljMyLjQ2ODM2NyIgcn9ljxLjc4MzA0NzkilHJ5PSlyMS43NjYwMDgiPjwvZ
WxsaXBzZT4KICAgICAglCAglCAglCAglCAglCA8ZWxsaXBzZSBpZD0iT3Zhbc0zliBmaWxsPSIj
Q0ZEQUU2liBvcGFjaXR5PSlwLjU5OTk5OTk2NCIgY3g9ljUuOTg3NDY0NzkilGN5PSIxMy44NjY4
NjAxliByeD0iNS4yMTczOTEzliByeT0iNS4yMTMzMdk5Nyl+PC9lbGxpcHNlPgogICAglCAglCAglC
AgICAglCAglCAglDXwYXRolGQ9Ik0zOC4xMzU0NTE0LDg4LjM1MjAyMTUgQzQzLjg5ODQyMjcsODg
uMzUyMDIxNSA0OC41NzAyMzQsODMuNjgzODY0NyA0OC41NzAyMzQsNzc0OTI1NDAXNSBD
NDguNTcwMjM0LDcyLjE2NjkzODMGNDMuODk4NDIlYnyw2Ny40OTg3ODE2IDM4LjEzNTQ1MTQ
sNjcuNDk4NzgxiBDMzluMzcyNDgwMSw2Ny40OTg3ODE2IDI3LjcwMDY2ODgsNzluMTY2OTM
4MyAyNy43MDA2Njg4LDc3LjkyNTQwMTUgQzI3LjcwMDY2ODgsODMuNjgzODY0NyAzMi4zNzlO
ODAXLDg4LjM1MjAyMTUgMzguMTM1NDUxNCw4OC4zNTIwMjE1FoilGlkPSJPdmFsLTMTQ29w
eSlgZmlsbD0iI0NGREFFNilgb3BhY2I0eT0iMC40NSI+PC9wYXRoPgogICAglCAglCAglCAglCAgl
CAglDXwYXRolGQ9Ik02NC4yNzc1NTgyLDMzLjE3MDQ5NjMgTdExOS4xODU4MzYsMTYuNTY1
NDkxNSIgaWQ9IIBhdGgtMTIilHN0cm9rZT0iI0NGREFFNilgc3Ryb2tlLXdpcZHRoPSIxLjczOTEzMD
QzliBzdHJva2UtbGluZWNhcd0icm91bmQiIHNoem9rZS13aW5lam9pbj0icm91bmQiPjwvcGF0aD4
KICAgICAglCAglCAglCAglCAglCA8cGF0aCBkPSJNNDIuMTQzMTCwOCwyNi41MDAYnjgxIEw3
LjcxMTkwMTYyLDE0LjU2NDA3MDIilGlkPSJQYXRolTE2liBzdHJva2U9liNFMEI0QjciIHNoem9rZ
S13aWR0aD0iMC43MDI2Nzg5NjQilG9wYWNpdHk9ljAuNyIgcn9ljxLjc4MzA0NzkilHJ5PSlyMS43NjYwMDgiPjwvZ
BzdHJva2UtbGluZWpvaW49InJvdW5kliBzdHJva2UtZGFzaGFycmF5PSIxLjQwNTM1Nzg5OTg3M
ze1MywyLjEwODAznjk1MzQ2OTk4MSI+PC9wYXRoPgogICAglCAglCAglCAglCAglCAglDXwYX
RolGQ9Ik02My45MjYyMTg3LDMzLjUyMTU2MSBMNDMuNjcyMTMyNiwi2OS4zMjUwOTUxliBpZD
0iUGF0aC0xNSIgc3Ryb2tlPSIjQkFDQUQ5liBzdHJva2Utd2lkdg9ljAuNzAyNjc4OTY0liBzdHJva2
UtbGluZWNhcd0icm91bmQiIHNoem9rZS13aW5lam9pbj0icm91bmQiIHNoem9rZS1kYXNoYXJyY
Xk9ljEuNDA1MzU3ODk5ODczMTUzLDluMTA4MDM2OTUzNDY5OTgxLj48L3BhdGg+CiAglCAgl
CAglCAglCAglCAglPGcgawQ9Ikdyb3VwL TE3liB0cmFuc2Zvc m09InRyYW5zbGF0ZSgxMjYu
ODUwOTIyLCAXMy41NDM2NTQpIHJvdGF0ZSgzMC4wMDAwMDApIHRyYW5zbGF0ZSgtMTI2Lj
g1MDkyMiwiLT E zLjU0MzY1NCKgdHJhbnNsYXRlKDExNy4yODU3MDUsIDQuMzgxODg5KSlgZ
mlsbD0iI0NGREFFNil+CiAglCAglCAglCAglCAglCAglCAglCAglDXlbGxpcHNlIGlkPSJPdmFsLTQi
IG9wYWNpdHk9ljAuNDUIlGN4PSI5LjEzNDgyNjUzliBjeT0iOS4xMjc2ODA3Nilgcng9ljkuMTM0ODI
2NTMIlHJ5PSI5LjEyNzY4MDc2lj48L2VsbGlwc2U+CiAglCAglCAglCAglCAglCAglCAglCAglDXwY
XRolGQ9Ik0xOC4yNjk2NTMxLDE4LjI1NTM2MTUgQzE4LjI2OTY1MzEsMTMuMjE0MjgyNiAxNC4

[illegible]

[illegible]

```
/* background-color: #112346; */
background-repeat: no-repeat;
background-position: center right;
background-size: 100%;
}

.login-container {
border-radius: 10px;
margin: 0px auto;
width: 350px;
padding: 30px 35px 15px 35px;
background: #fff;
border: 1px solid #eaeaea;
text-align: left;
```

```
    box-shadow: 0 0 20px 2px rgba(0, 0, 0, 0.1);
  }
  .title {
    margin: 0px auto 40px auto;
    text-align: center;
    color: #505458;
  }
  .remember {
    margin: 0px 0px 35px 0px;
  }
  .code-box {
    text-align: right;
  }
  .codeimg {
    height: 40px;
  }
</style>
```

18:F:\git\nodejs\vue\vuestep\vueproject\src\views\machine\Machine.vue

/**

*

*/

<template>

<div> </div>

</template>

<script>

export default {

}

</script>

<style>

</style>

19:F:\git\nodejs\vue\vuestep\vueproject\src\views\machine\MachineAisle.vue

/**

*

*/

```
<template>
  <div> </div>
</template>
```

```
<script>
export default {

}
</script>
```

```
<style>
```

```
</style>
```

20:F:\git\nodejs\vue\vuestep\vueproject\src\views\machine\MachineConfig.vue

```
/**
 *
 */
<template>
  <div>
    <!-- -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
      <el-breadcrumb-item></el-breadcrumb-item>
    </el-breadcrumb>
    <!-- -->
    <el-form :inline="true" :model="formInline" class="user-search">
      <el-form-item label="">
        <el-input size="small" v-model="formInline.machineNo" placeholder=""></el-input>
      </el-form-item>
      <el-form-item>
        <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>
        <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>
      </el-form-item>
    </el-form>
    <!------>
    <el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-
loading-text="" style="width: 100%;">
      <el-table-column align="center" type="selection" width="60">
        </el-table-column>
```

```

<el-table-column sortable prop="machineNo" label="" width="200">
</el-table-column>
<el-table-column sortable prop="payType" label="" width="200">
</el-table-column>
<el-table-column sortable prop="configName" label="" width="200">
</el-table-column>
<el-table-column sortable prop="payOpen" label="" width="200">
</el-table-column>
<el-table-column sortable prop="editTime" label="" width="200">
  <template slot-scope="scope">
    <div>{{scope.row.editTime|timestampToTime}}</div>
    </el-switch>
  </template>
</el-table-column>
<el-table-column sortable prop="editUser" label="" width="150">
</el-table-column>
<el-table-column align="center" label="" min-width="300">
  <template slot-scope="scope">
    <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
    <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click="closeDialog">
  <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
    <el-form-item label="" prop="machineNo">
      <el-input size="small" v-model="editForm.machineNo" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="payType">
      <el-select size="small" v-model="editForm.payType" placeholder="" class="userRole">
        <el-option v-for="type in payType" :label="type.key" :value="type.value"
:key="type.value"></el-option>
      </el-select>
    </el-form-item>
    <el-form-item label="" prop="configId">
      <el-input size="small" v-model="editForm.configId" auto-complete="off" placeholder=""></el-
input>

```

```

    </el-form-item>
    <el-form-item label="" prop="configName">
      <el-input size="small" v-model="editForm.configName" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="payOpen">
      <el-select size="small" v-model="editForm.payOpen" placeholder="" class="userRole">
        <el-option label="" value="T"></el-option>
        <el-option label="" value="N"></el-option>
      </el-select>
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button size="small" @click="closeDialog"></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
  </div>
</el-dialog>
</div>
</template>

```

```

<script>
import {
  MachineConfigList,
  MachineConfigSave,
  MachineConfigDelete
} from '../api/payMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      editFormVisible: false, //
      title: "",
      payType: [
        { key: "", value: 1 },
        { key: "", value: 2 },
        { key: "", value: 3 },
        { key: 'POS', value: 4 },
        { key: "", value: 5 },

```

```

    { key: 'POSCB', value: 6 },
    { key: '', value: 8 },
    { key: '', value: 9 }
  ],
  editForm: {
    tcId: '',
    machineNo: '',
    payType: '',
    configId: '',
    configName: '',
    payOpen: '',
    token: localStorage.getItem('logintoken')
  },
  // rules
  rules: {
    machineNo: [
      { required: true, message: '', trigger: 'blur' }
    ],
    payType: [
      { required: true, message: '', trigger: 'blur' }
    ],
    configId: [
      { required: true, message: '', trigger: 'blur' }
    ],
    configName: [
      { required: true, message: '', trigger: 'blur' }
    ],
    payOpen: [{ required: true, message: '', trigger: 'blur' }]
  },
  formInline: {
    page: 1,
    limit: 10,
    varLable: '',
    varName: '',
    token: localStorage.getItem('logintoken')
  },
  //
  seletedata: {
    ids: '',
    token: localStorage.getItem('logintoken')
  },
  userparm: [], //

```



```

    listData: [], //
    //
    pageparm: {
      currentPage: 1,
      pageSize: 10,
      total: 10
    }
  },
  //
  components: {
    Pagination
  },
  /**
   *
   */

  /**
   *
   */
  created() {
    this.getdata(this.formInline)
  },

  /**
   *
   */
  methods: {
    //
    getdata(parameter) {
      this.loading = true
      //
      let res = {
        code: 0,
        msg: null,
        count: 5,
        data: [
          {
            addUser: null,
            editUser: null,
            addTime: null,
            editTime: 1524046759000,

```

```
tclId: 1,
deptId: 1,
machineNo: '564565656666',
payType: 3,
payOpen: 'T',
configId: 63,
configName: "",
posNo: '098'
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  tclId: 2,
  deptId: 1,
  machineNo: '66666666',
  payType: 2,
  payOpen: 'T',
  configId: 64,
  configName: "",
  posNo: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  tclId: 3,
  deptId: 1,
  machineNo: '93066545645546500791',
  payType: 6,
  payOpen: 'T',
  configId: 67,
  configName: "",
  posNo: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
```

```

    tclId: 4,
    deptId: 1,
    machineNo: '65545656565',
    payType: 6,
    payOpen: 'T',
    configId: 67,
    configName: "",
    posNo: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1527409037000,
    editTime: 1527409037000,
    tclId: 6,
    deptId: 1,
    machineNo: '565654545454545',
    payType: 6,
    payOpen: 'T',
    configId: 96,
    configName: "",
    posNo: null
  }
]
}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
// MachineConfigList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',

```

```

//      message: res.msg
//    })
//  } else {
//    this.listData = res.data
//    //
//    this.pageparm.currentPage = this.formInline.page
//    this.pageparm.pageSize = this.formInline.limit
//    this.pageparm.total = res.count
//  }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = "
    this.editForm.tclId = row.tclId
    this.editForm.machineNo = row.machineNo
    this.editForm.configId = row.configId
    this.editForm.payType = row.payType
    this.editForm.configName = row.configName
    this.editForm.payOpen = row.payOpen
  } else {
    this.title = "
    this.editForm.tclId = "
    this.editForm.machineNo = "
    this.editForm.payType = "
    this.editForm.configId = "

```

```

    this.editForm.configName = "
    this.editForm.payOpen = "
  }
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      MachineConfigSave(this.editForm)
        .then(res => {
          this.editForm.Visible = false
          this.loading = false
          if (res.success) {
            this.getdata(this.formInline)
            this.$message({
              type: 'success',
              message: "
            })
          } else {
            this.$message({
              type: 'info',
              message: res.msg
            })
          }
        })
      })
      .catch(err => {
        this.editForm.Visible = false
        this.loading = false
        this.$message.error("")
      })
    } else {
      return false
    }
  })
},
//
deleteUser(index, row) {
  this.$confirm('?', "", {
    confirmButtonText: "",
    cancelButtonText: "",
    type: 'warning'
  })
}

```

```

.then(() => {
  MachineConfigDelete(row.deptId)
  .then(res => {
    if (res.success) {
      this.$message({
        type: 'success',
        message: '!'
      })
      this.getdata(this.formInline)
    } else {
      this.$message({
        type: 'info',
        message: res.msg
      })
    }
  })
  .catch(err => {
    this.loading = false
    this.$message.error("")
  })
})
.catch(() => {
  this.$message({
    type: 'info',
    message: "
  })
})
},
//
closeDialog() {
  this.editFormVisible = false
}
}
}
</script>

```

```

<style scoped>
.user-search {
  margin-top: 20px;
}
.userRole {
  width: 100%;

```

```
}  
</style>
```

21:F:\git\nodejs\vue\vuestep\vueproject\src\views\pay\Config.vue

```
/**  
 *  
 */  
<template>  
  <div>  
    <!-- -->  
    <el-breadcrumb separator-class="el-icon-arrow-right">  
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>  
      <el-breadcrumb-item></el-breadcrumb-item>  
    </el-breadcrumb>  
    <!-- -->  
    <el-form :inline="true" :model="formInline" class="user-search">  
      <el-form-item label="">  
        <el-input size="small" v-model="formInline.name" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-select size="small" v-model="formInline.payType" placeholder="">  
          <el-option v-for="type in payType" :label="type.key" :value="type.value"  
:key="type.value"></el-option>  
        </el-select>  
      </el-form-item>  
      <el-form-item label="">  
        <el-input size="small" v-model="formInline.partner" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>  
        <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>  
      </el-form-item>  
    </el-form>  
    <!-- ---->  
    <el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-  
loading-text="" style="width: 100%;">  
      <el-table-column align="center" type="index" width="60">  
      </el-table-column>  
      <el-table-column sortable prop="name" label="" width="200" show-overflow-tooltip>
```

```

</el-table-column>
<el-table-column sortable prop="payType" label="" width="120" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="partner" label="" width="100" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="subMchId" label="" width="140" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="appid" label="ID" width="100" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="signType" label="" width="120" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="partnerKey" label="" width="180" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="notifyUrl" label="" width="140" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="sellerUserId" label="" width="150" show-overflow-tooltip>
</el-table-column>
<el-table-column sortable prop="certPath" label="" width="150" show-overflow-tooltip>
</el-table-column>
<el-table-column align="center" label="" min-width="150">
  <template slot-scope="scope">
    <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
    <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%"
@click="closeDialog('editForm')">
  <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
    <el-form-item label="" prop="name">
      <el-input size="small" v-model="editForm.name" auto-complete="off" placeholder=""></el-
input>
    </el-form-item>
    <el-form-item label="" prop="payType">
      <el-select size="small" v-model="editForm.payType" placeholder="" class="userRole">
        <el-option v-for="type in payType" :label="type.key" :value="type.value"
:key="type.value"></el-option>
      </el-select>

```



```
</el-form-item>
<el-form-item label="" prop="partner">
  <el-input size="small" v-model="editForm.partner" auto-complete="off" placeholder=""></el-
input>
</el-form-item>
<el-form-item label="" prop="subMchId">
  <el-input size="small" v-model="editForm.subMchId" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="ID" prop="appid">
  <el-input size="small" v-model="editForm.appid" auto-complete="off" placeholder="ID"></el-
input>
</el-form-item>
<el-form-item label="" prop="notifyUrl">
  <el-input size="small" v-model="editForm.notifyUrl" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="signType">
  <el-input size="small" v-model="editForm.signType" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="partnerKey">
  <el-input size="small" v-model="editForm.partnerKey" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="sellerUserId">
  <el-input size="small" v-model="editForm.sellerUserId" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="certPath">
  <el-input size="small" v-model="editForm.certPath" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="certPassword">
  <el-input size="small" v-model="editForm.certPassword" auto-complete="off" placeholder="
"></el-input>
</el-form-item>
<el-form-item label="" prop="rsaKey">
  <el-input size="small" v-model="editForm.rsaKey" auto-complete="off" placeholder=""></el-
input>
</el-form-item>
</el-form>
```

```
<div slot="footer" class="dialog-footer">
  <el-button size="small" @click="closeDialog('editForm')"></el-button>
  <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
</div>
</el-dialog>
</div>
</template>
```

```
<script>
import { ConfigList, ConfigSave, ConfigDelete } from '../api/payMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      loading: false, //
      editFormVisible: false, //
      title: "",
      payType: [
        { key: "", value: 0 },
        { key: "", value: 1 },
        { key: "", value: 2 },
        { key: "", value: 3 },
        { key: 'POS', value: 4 },
        { key: "", value: 5 },
        { key: 'POSCB', value: 6 },
        { key: "", value: 8 },
        { key: "", value: 9 }
      ],
      editForm: {
        id: "",
        name: "",
        payType: 1,
        partner: "",
        subMchId: "",
        appid: "",
        notifyUrl: "",
        signType: "",
        partnerKey: "",
        sellerUserId: "",
        certPath: "",
        certPassword: "",
```

```
rsaKey: "",
token: localStorage.getItem('logintoken')
},
// rules
rules: {
  name: [{ required: true, message: "", trigger: 'blur' }],
  payType: [
    { required: true, message: "", trigger: 'blur' }
  ],
  partner: [{ required: true, message: "", trigger: 'blur' }],
  subMchId: [
    { required: true, message: "", trigger: 'blur' }
  ],
  appid: [{ required: true, message: 'ID', trigger: 'blur' }],
  notifyUrl: [
    { required: true, message: "", trigger: 'blur' }
  ],
  signType: [
    { required: true, message: "", trigger: 'blur' }
  ],
  partnerKey: [
    { required: true, message: "", trigger: 'blur' }
  ],
  sellerUserId: [
    { required: true, message: "", trigger: 'blur' }
  ],
  certPath: [
    { required: true, message: "", trigger: 'blur' }
  ],
  certPassword: [
    { required: true, message: "", trigger: 'blur' }
  ],
  rsaKey: [
    { required: true, message: "", trigger: 'blur' }
  ]
},
formInline: {
  page: 1,
  limit: 10,
  name: "",
  payType: 0,
  partner: "",
```

```

        token: localStorage.getItem('logintoken')
    },
    //
    seletedata: {
        ids: "",
        token: localStorage.getItem('logintoken')
    },
    userparm: [], //
    listData: [], //
    //
    pageparm: {
        currentPage: 1,
        pageSize: 10,
        total: 10
    }
    }
},
//
components: {
    Pagination
},
/**
 *
 */

/**
 *
 */
created() {
    this.getdata(this.formInline)
},

/**
 *
 */
methods: {
    //
    getdata(parameter) {
        this.loading = true
        //
        let res = {
            code: 0,

```

```
msg: null,
count: 207,
data: [
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 1,
    name: '2.0',
    payType: 1,
    partner: '2015122801047567',
    subMchId: '',
    appId: '2015122801047567',
    notifyUrl: 'r/pay/alipay/notify',
    signType: 'RSA',
    partnerKey: '==',
    sellerUserId: '2088121360144859',
    certPath: '',
    certPassword: '',
    rsaKey: '',
    deptId: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 2,
    name: 'zzzzzz',
    payType: 2,
    partner: '1250856201',
    subMchId: null,
    appId: 'wx3ef7713adf0a97b8',
    notifyUrl: null,
    signType: 'NATIVE',
    partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
    sellerUserId: null,
    certPath:
      '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
      SNAPSHOT/conf/apiclient_cert_1250856201.p12',
    certPassword: '1250856201',
```

```
    rsaKey: null,
    deptId: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 3,
    name: ' ',
    payType: 2,
    partner: '1271942301',
    subMchId: '1273729701',
    appId: 'wx3ef7713adf0a97b8',
    notifyUrl: '/pay/wx/notify',
    signType: 'NATIVE',
    partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
    sellerUserId: '',
    certPath:
      '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
    certPassword: '1271942301',
    rsaKey: null,
    deptId: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 5,
    name: '301',
    payType: 2,
    partner: '1271942301',
    subMchId: '1341564201',
    appId: 'wx3ef7713adf0a97b8',
    notifyUrl: 'er/pay/wx/notify',
    signType: 'NATIVE',
    partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
    sellerUserId: null,
    certPath:
      '/usr/local/tomcat_provider/webapps/---0.0.1-
```

```
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
  certPassword: '1271942301',
  rsaKey: null,
  deptId: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  id: 6,
  name: '301',
  payType: 2,
  partner: '1271942301',
  subMchId: '1284797101',
  appId: 'wx3ef7713adf0a97b8',
  notifyUrl:
    'http://180.166.211.210:8114/machine-pay-consumer/pay/wx/notify',
  signType: 'NATIVE',
  partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
  sellerUserId: null,
  certPath:
    '/usr/local/tomcat_provider/webapps/--provider-0.0.1-
```

```
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
  certPassword: '1271942301',
  rsaKey: null,
  deptId: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  id: 7,
  name: '301',
  payType: 2,
  partner: '1271942301',
  subMchId: '1277531101',
  appId: 'wx3ef7713adf0a97b8',
  notifyUrl:
    'http://180.166.211.210:8114/machine-pay-consumer/pay/wx/notify',
  signType: 'NATIVE',
```

```
partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
sellerUserId: null,
certPath:
  '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
certPassword: '1271942301',
rsaKey: null,
deptId: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  id: 8,
  name: '301',
  payType: 2,
  partner: '1271942301',
  subMchId: '1276485301',
  appId: 'wx3ef7713adf0a97b8',
  notifyUrl:
    'http://180.166.211.210:8114/machine-pay-consumer/pay/wx/notify',
  signType: 'NATIVE',
  partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
  sellerUserId: null,
  certPath:
    '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
  certPassword: '1271942301',
  rsaKey: null,
  deptId: null
},
{
  addUser: null,
  editUser: null,
  addTime: null,
  editTime: null,
  id: 9,
  name: '',
  payType: 2,
  partner: '1347158201',
  subMchId: '1351034701',
```



```
    appid: 'wx83a7489c10c9c952',
    notifyUrl: '',
    signType: 'NATIVE',
    partnerKey: 'f174607ba704632b6cad2df8b04650d6',
    sellerUserId: null,
    certPath:
      '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1347158201.p12',
    certPassword: '1347158201',
    rsaKey: null,
    deptId: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 10,
    name: '-301',
    payType: 2,
    partner: '1271942301',
    subMchId: '1357984702',
    appid: 'wx3ef7713adf0a97b8',
    notifyUrl:
      'http://180.166.211.210:8114/machine-pay-consumer/pay/wx/notify',
    signType: 'NATIVE',
    partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
    sellerUserId: null,
    certPath:
      '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
    certPassword: '1271942301',
    rsaKey: null,
    deptId: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: null,
    editTime: null,
    id: 11,
    name: '-301',
```

```

    payType: 2,
    partner: '1271942301',
    subMchId: '1357972202',
    appId: 'wx3ef7713adf0a97b8',
    notifyUrl:
      'http://180.166.211.210:8114/machine-pay-consumer/pay/wx/notify',
    signType: 'NATIVE',
    partnerKey: '2e3cdaf5aa051c16563c0b8916184d5d',
    sellerUserId: null,
    certPath:
      '/usr/local/tomcat_provider/webapps/machine-service-provider-0.0.1-
SNAPSHOT/conf/apiclient_cert_1271942301.p12',
    certPassword: '1271942301',
    rsaKey: null,
    deptId: null
  }
]
}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
// ConfigList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page
//     this.pageparm.pageSize = this.formInline.limit

```

```

//    this.pageparm.total = res.count
//  }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = ""
    this.editForm.id = row.id
    this.editForm.name = row.name
    this.editForm.payType = row.payType
    this.editForm.partner = row.partner
    this.editForm.subMchId = row.subMchId
    this.editForm.appid = row.appid
    this.editForm.notifyUrl = row.notifyUrl
    this.editForm.signType = row.signType
    this.editForm.partnerKey = row.partnerKey
    this.editForm.sellerUserId = row.sellerUserId
    this.editForm.certPath = row.certPath
    this.editForm.certPassword = row.certPassword
    this.editForm.rsaKey = row.rsaKey
  } else {
    this.title = ""
    this.editForm.id = ""
    this.editForm.name = ""
    this.editForm.payType = ""
    this.editForm.partner = ""

```

```

this.editForm.subMchId = "
this.editForm.appid = "
this.editForm.notifyUrl = "
this.editForm.signType = "
this.editForm.partnerKey = "
this.editForm.sellerUserId = "
this.editForm.certPath = "
this.editForm.certPassword = "
this.editForm.rsaKey = "
}
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      ConfigSave(this.editForm)
        .then(res => {
          this.editFormVisible = false
          this.loading = false
          if (res.success) {
            this.getdata(this.formInline)
            this.$message({
              type: 'success',
              message: "
            })
          } else {
            this.$message({
              type: 'info',
              message: res.msg
            })
          }
        })
      .catch(err => {
        this.editFormVisible = false
        this.loading = false
        this.$message.error("")
      })
    } else {
      return false
    }
  })
},

```

```

//
deleteUser(index, row) {
  this.$confirm('?', '', {
    confirmButtonText: '',
    cancelButtonText: '',
    type: 'warning'
  })
  .then(() => {
    ConfigDelete(row.deptId)
      .then(res => {
        if (res.success) {
          this.$message({
            type: 'success',
            message: '!'
          })
          this.getdata(this.formInline)
        } else {
          this.$message({
            type: 'info',
            message: res.msg
          })
        }
      })
    })
  .catch(err => {
    this.loading = false
    this.$message.error('')
  })
})
.catch(() => {
  this.$message({
    type: 'info',
    message: ''
  })
})
},
//
closeDialog(formName) {
  this.editFormVisible = false
  this.$refs[formName].resetFields()
}
}
}

```

```
</script>
```

```
<style scoped>
```

```
.user-search {  
  margin-top: 20px;  
}
```

```
.userRole {  
  width: 100%;  
}
```

```
</style>
```

22:F:\git\nodejs\vue\vuestep\vueproject\src\views\pay\Order.vue

```
/**  
 *  
 */  
<template>  
  <div>  
    <!-- -->  
    <el-breadcrumb separator-class="el-icon-arrow-right">  
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>  
      <el-breadcrumb-item></el-breadcrumb-item>  
    </el-breadcrumb>  
    <!-- -->  
    <el-form :inline="true" :model="formInline" class="user-search">  
      <el-form-item label="">  
        <el-input size="small" v-model="formInline.machineNo" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-input size="small" v-model="formInline.orderNo" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-input size="small" v-model="formInline.transId" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-select size="small" v-model="formInline.payType" placeholder="">  
          <el-option v-for="type in payType" :label="type.key" :value="type.value"  
:key="type.value"></el-option>  
        </el-select>  
      </el-form-item>
```

```

<el-form-item>
  <el-select size="small" v-model="formInline.orderStatus" placeholder="">
    <el-option v-for="type in payway" :label="type.key" :value="type.value"
:key="type.value"></el-option>
  </el-select>
</el-form-item>
<el-form-item>
  <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>
</el-form-item>
</el-form>
<!-->
<el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-
loading-text="" style="width: 100%;">
  <el-table-column align="center" type="index" width="60">
</el-table-column>
  <el-table-column sortable prop="machineNo" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="orderNo" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="transId" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="payType" label="" width="140" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="transType" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="goodsPrice" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="payAmount" label="" width="180" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="goodsName" label="" width="140" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="orderStatus" label="" width="120" show-overflow-tooltip>
</el-table-column>
  <el-table-column sortable prop="addTime" label="" width="180" show-overflow-tooltip>
    <template slot-scope="scope">
      <div>{{scope.row.addTime|timestampToTime}}</div>
    </template>
  </el-table-column>
  <el-table-column align="center" label="" min-width="150">
    <template slot-scope="scope">
      <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
      <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-

```

```

button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="50%"
@click="closeDialog('editForm')">
  <el-form label-width="120px" :model="editForm" ref="editForm">
    <el-row>
      <el-col :span="12">
        <el-form-item label="">
          <el-input size="small" v-model="editForm.deptId" auto-complete="off" placeholder=""
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.orderNo" auto-complete="off" placeholder=""
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.payType" auto-complete="off" placeholder=""
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.transType" auto-complete="off" placeholder=""
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.goodsNo" auto-complete="off" placeholder="ID"
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.payAmount" auto-complete="off" placeholder="
" disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.aisleNo" auto-complete="off" placeholder=""
disabled></el-input>
        </el-form-item>
        <el-form-item label="">
          <el-input size="small" v-model="editForm.openId" auto-complete="off" placeholder=""

```



```
disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.subMchId" auto-complete="off" placeholder=""
disabled></el-input>
  </el-form-item>
</el-col>
<el-col :span="12">
  <el-form-item label="">
    <el-input size="small" v-model="editForm.deptName" auto-complete="off" placeholder=""
disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.transId" auto-complete="off" placeholder=""
disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.subPayType" auto-complete="off"
placeholder="" disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.machineNo" auto-complete="off" placeholder=""
" disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.goodsPrice" auto-complete="off" placeholder=""
ID" disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.goodsName" auto-complete="off"
placeholder="" disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.orderStatus" auto-complete="off" placeholder=""
" disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.mchId" auto-complete="off" placeholder=""
disabled></el-input>
  </el-form-item>
  <el-form-item label="">
    <el-input size="small" v-model="editForm.editUser" auto-complete="off" placeholder=""
```

```

disabled></el-input>
    </el-form-item>
</el-col>
</el-row>
<el-form-item label="">
    <el-input size="small" v-model="editForm.remark" auto-complete="off" placeholder=""
disabled></el-input>
    </el-form-item>
</el-form>
</el-dialog>
</div>
</template>

```

```

<script>
import { OrderList, OrderRefund, OrderDelete } from '../api/payMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      loading: false, //
      editFormVisible: false, //
      title: "",
      payType: [
        { key: "", value: 0 },
        { key: "", value: 1 },
        { key: "", value: 2 },
        { key: "", value: 3 },
        { key: 'POS', value: 4 },
        { key: "", value: 5 },
        { key: 'POSCB', value: 6 },
        { key: "", value: 8 },
        { key: "", value: 9 }
      ],
      payway: [
        { key: "", value: 0 },
        { key: "", value: 1 },
        { key: "", value: 2 },
        { key: "", value: 3 },
        { key: "", value: 4 },
        { key: "", value: 5 },
        { key: "", value: 11 },
        { key: "", value: 12 },

```

```
{ key: ", value: 13 },
{ key: ", value: 14 },
{ key: ", value: 10 }
],
editForm: {
  id: ",
  name: ",
  payType: 1,
  partner: ",
  subMchId: ",
  appid: ",
  notifyUrl: ",
  signType: ",
  partnerKey: ",
  sellerUserId: ",
  certPath: ",
  certPassword: ",
  rsaKey: ",
  token: localStorage.getItem('logintoken')
},
formInline: {
  page: 1,
  limit: 10,
  machineNo: ",
  orderNo: ",
  transId: ",
  payType: 0,
  orderStatus: 0,
  token: localStorage.getItem('logintoken')
},
//
seletedata: {
  ids: ",
  token: localStorage.getItem('logintoken')
},
userparm: [], //
listData: [], //
//
pageparm: {
  currentPage: 1,
  pageSize: 10,
  total: 10
```

```

    }
  }
},
//
components: {
  Pagination
},
/**
 *
 */

/**
 *
 */
created() {
  this.getdata(this.formInline)
},

/**
 *
 */
methods: {
  //
  getdata(parameter) {
    this.loading = true
    //
    let res = {
      code: 0,
      msg: null,
      count: 23,
      data: [
        {
          addUser: null,
          editUser: null,
          addTime: 1526380193000,
          editTime: 1526380193000,
          orderId: 109,
          deptId: 1,
          deptName: 'xxxx',
          orderNo: 'xxxx',
          transId: 'xxxx',
          payType: 6,

```

```
subPayType: 'WXPAY',
transType: '',
machineNo: '111111',
goodsNo: '123456',
goodsPrice: 0.01,
payAmount: -0.01,
goodsName: '',
aisleNo: null,
orderStatus: 14,
openId: null,
mchId: '1111111111111111',
subMchId: null,
remark: 'IP: 101.81.251.226'
},
{
  addUser: null,
  editUser: null,
  addTime: 1526380176000,
  editTime: 1526380176000,
  orderId: 108,
  deptId: 1,
  deptName: 'xxxxxx',
  orderNo: 'xxxx',
  transId: 'xxxxx',
  payType: 6,
  subPayType: 'WXPAY',
  transType: '',
  machineNo: 'J1AX904002',
  goodsNo: '123456',
  goodsPrice: 0.01,
  payAmount: -0.01,
  goodsName: '',
  aisleNo: null,
  orderStatus: 14,
  openId: null,
  mchId: '898310154990338',
  subMchId: null,
  remark: 'IP: 101.81.251.226'
},
{
  addUser: null,
  editUser: null,
```

```
    addTime: 1524921444000,
    editTime: 1524894094000,
    orderId: 107,
    deptId: 1,
    deptName: 'xxxxxx',
    orderNo: 'J1AX90400220180428101723945',
    transId: '4200000137201804287543647891',
    payType: 6,
    subPayType: 'WXPay',
    transType: '',
    machineNo: 'J1AX904002',
    goodsNo: '123456',
    goodsPrice: 0.01,
    payAmount: 0.01,
    goodsName: '',
    aisleNo: null,
    orderStatus: 7,
    openId: null,
    mchId: '898310154990338',
    subMchId: null,
    remark: ''
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1521307596000,
    editTime: 1524641207000,
    orderId: 20,
    deptId: 1,
    deptName: 'xxxx',
    orderNo: '9300079120180318142634440',
    transId: null,
    payType: 0,
    subPayType: '0',
    transType: '',
    machineNo: '1111111111111111',
    goodsNo: '123456',
    goodsPrice: 0.01,
    payAmount: 0.01,
    goodsName: '',
    aisleNo: null,
    orderStatus: 7,
```

```
    openId: null,
    mchId: null,
    subMchId: null,
    remark: '11111111111111111111111111111111'
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1520195909000,
    editTime: 1520195909000,
    orderId: 19,
    deptId: 1,
    deptName: 'xxxx',
    orderNo: '9300079120180305183828606',
    transId: null,
    payType: 0,
    subPayType: '0',
    transType: '',
    machineNo: '93000791',
    goodsNo: '123456',
    goodsPrice: 0.01,
    payAmount: 0.01,
    goodsName: '',
    aisleNo: null,
    orderStatus: 1,
    openId: null,
    mchId: null,
    subMchId: null,
    remark: null
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1520035180000,
    editTime: 1520035180000,
    orderId: 18,
    deptId: 1,
    deptName: 'xxxx',
    orderNo: '4200000056201803031934477774',
    transId: '9300079120180303170851281',
    payType: 6,
    subPayType: 'WXPAY',
```

```
transType: "",
machineNo: '22222222222222222222',
goodsNo: '123456',
goodsPrice: 0.01,
payAmount: 0.01,
goodsName: "",
aisleNo: null,
orderStatus: 8,
openId: null,
mchId: '898310154990338',
subMchId: null,
remark: null
},
{
  addUser: null,
  editUser: null,
  addTime: 1520020261000,
  editTime: 1520185478000,
  orderId: 17,
  deptId: 1,
  deptName: 'xxxx',
  orderNo: '9300079120180303175059985',
  transId: '4200000072201803031887274444',
  payType: 6,
  subPayType: 'WXPAY',
  transType: "",
  machineNo: '93000791',
  goodsNo: '123456',
  goodsPrice: 0.01,
  payAmount: 0.01,
  goodsName: "",
  aisleNo: null,
  orderStatus: 7,
  openId: null,
  mchId: '898310154990338',
  subMchId: null,
  remark: 'IP: 116.247.119.165'
},
{
  addUser: null,
  editUser: null,
  addTime: 1520019911000,
```



```
editTime: 1520020075000,  
orderId: 16,  
deptId: 1,  
deptName: 'xxxx',  
orderNo: '9300079120180303174511778',  
transId: '4200000055201803031949877221',  
payType: 6,  
subPayType: 'WXPay',  
transType: '',  
machineNo: '93000791',  
goodsNo: '123456',  
goodsPrice: 0.01,  
payAmount: 0.01,  
goodsName: '',  
aisleNo: null,  
orderStatus: 1,  
openId: null,  
mchId: '898310154990338',  
subMchId: null,  
remark: null  
},  
{  
  addUser: null,  
  editUser: null,  
  addTime: 1520019776000,  
  editTime: 1520019776000,  
  orderId: 15,  
  deptId: 1,  
  deptName: 'xxxx',  
  orderNo: '9300079120180303174256156',  
  transId: null,  
  payType: 0,  
  subPayType: '0',  
  transType: '',  
  machineNo: '93000791',  
  goodsNo: '123456',  
  goodsPrice: 0.01,  
  payAmount: 0.01,  
  goodsName: '',  
  aisleNo: null,  
  orderStatus: 1,  
  openId: null,
```

```

        mchId: null,
        subMchId: null,
        remark: null
    },
    {
        addUser: null,
        editUser: null,
        addTime: 1520019729000,
        editTime: 1520019729000,
        orderId: 14,
        deptId: 1,
        deptName: 'XX',
        orderNo: '9300079120180303174208429',
        transId: null,
        payType: 0,
        subPayType: '0',
        transType: '',
        machineNo: '93000791',
        goodsNo: '123456',
        goodsPrice: 0.01,
        payAmount: 0.01,
        goodsName: '',
        aisleNo: null,
        orderStatus: 1,
        openId: null,
        mchId: null,
        subMchId: null,
        remark: null
    }
]
}
this.loading = false
this.listData = res.data
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/****
*
*/

```

```

// OrderList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page
//     this.pageparm.pageSize = this.formInline.limit
//     this.pageparm.total = res.count
//   }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  this.editForm = row
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      ConfigSave(this.editForm)
      .then(res => {

```

```

    this.editFormVisible = false
    this.loading = false
    if (res.success) {
        this.getdata(this.formInline)
        this.$message({
            type: 'success',
            message: "
        })
    } else {
        this.$message({
            type: 'info',
            message: res.msg
        })
    }
})
.catch(err => {
    this.editFormVisible = false
    this.loading = false
    this.$message.error("")
})
} else {
    return false
}
})
},
//
deleteUser(index, row) {
    this.$confirm('?', "", {
        confirmButtonText: "",
        cancelButtonText: "",
        type: 'warning'
    })
    .then(() => {
        ConfigDelete(row.deptId)
        .then(res => {
            if (res.success) {
                this.$message({
                    type: 'success',
                    message: '!'
                })
                this.getdata(this.formInline)
            } else {

```

```

        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
    .catch(err => {
      this.loading = false
      this.$message.error("")
    })
  })
  .catch(() => {
    this.$message({
      type: 'info',
      message: ""
    })
  })
},
//
closeDialog(formName) {
  this.editFormVisible = false
  this.$refs[formName].resetFields()
}
}
}
</script>

```

```

<style scoped>
.user-search {
  margin-top: 20px;
}
.userRole {
  width: 100%;
}
</style>

```

```

/**
 *
 */
<template>
  <div>
    <!-- -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
      <el-breadcrumb-item></el-breadcrumb-item>
    </el-breadcrumb>
    <!-- -->
    <el-form :inline="true" :model="formInline" class="user-search">
      <el-form-item label="">
        <el-input size="small" v-model="formInline.deptName" placeholder=""></el-input>
      </el-form-item>
      <el-form-item label="">
        <el-input size="small" v-model="formInline.deptNo" placeholder=""></el-input>
      </el-form-item>
      <el-form-item>
        <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>
        <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>
      </el-form-item>
    </el-form>
    <!------>
    <el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-loading-text="" style="width: 100%;">
      <el-table-column align="center" type="selection" width="60">
      </el-table-column>
      <el-table-column sortable prop="deptName" label="" width="300">
      </el-table-column>
      <el-table-column sortable prop="deptNo" label="" width="300">
      </el-table-column>
      <el-table-column sortable prop="editTime" label="" width="300">
        <template slot-scope="scope">
          <div>{{scope.row.editTime|timestampToTime}}</div>
        </template>
      </el-table-column>
      <el-table-column sortable prop="editUser" label="" width="300">
      </el-table-column>
      <el-table-column align="center" label="" min-width="300">
        <template slot-scope="scope">
          <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>

```

```

        <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
    </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click="closeDialog">
    <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
        <el-form-item label="" prop="deptName">
            <el-input size="small" v-model="editForm.deptName" auto-complete="off" placeholder="
"></el-input>
        </el-form-item>
        <el-form-item label="" prop="deptNo">
            <el-input size="small" v-model="editForm.deptNo" auto-complete="off" placeholder=""></el-
input>
        </el-form-item>
    </el-form>
    <div slot="footer" class="dialog-footer">
        <el-button size="small" @click="closeDialog"></el-button>
        <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
    </div>
</el-dialog>
</div>
</template>

```

```

<script>
import { deptList, deptSave, deptDelete } from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      editFormVisible: false, //
      title: "",
      editForm: {
        deptId: "",
        deptName: "",

```

```

    deptNo: "",
    token: localStorage.getItem('logintoken')
  },
  // rules
  rules: {
    deptName: [
      { required: true, message: "", trigger: 'blur' }
    ],
    deptNo: [{ required: true, message: "", trigger: 'blur' }]
  },
  formInline: {
    page: 1,
    limit: 10,
    varLable: "",
    varName: "",
    token: localStorage.getItem('logintoken')
  },
  //
  seletedata: {
    ids: "",
    token: localStorage.getItem('logintoken')
  },
  userparm: [], //
  listData: [], //
  //
  pageparm: {
    currentPage: 1,
    pageSize: 10,
    total: 10
  }
}
//
components: {
  Pagination
},
/**
 *
 */

/**
 *

```



```

*/
created() {
  this.getdata(this.formInline)
},

/**
 *
 */
methods: {
  //
  getdata(parameter) {
    this.loading = true
    //
    let res = {
      code: 0,
      msg: null,
      count: 5,
      data: [
        {
          addUser: null,
          editUser: null,
          addTime: 1521062371000,
          editTime: 1526700200000,
          deptId: 2,
          deptName: "",
          deptNo: '1',
          parentId: 1
        },
        {
          addUser: null,
          editUser: null,
          addTime: 1521063247000,
          editTime: 1526652291000,
          deptId: 3,
          deptName: "",
          deptNo: '02',
          parentId: 1
        },
        {
          addUser: null,
          editUser: null,
          addTime: 1526349555000,

```

```

        editTime: 1526349565000,
        deptId: 12,
        deptName: '1',
        deptNo: '11',
        parentId: 1
    },
    {
        addUser: null,
        editUser: null,
        addTime: 1526373178000,
        editTime: 1526373178000,
        deptId: 13,
        deptName: '5',
        deptNo: '5',
        parentId: 1
    },
    {
        addUser: null,
        editUser: null,
        addTime: 1526453107000,
        editTime: 1526453107000,
        deptId: 17,
        deptName: 'v',
        deptNo: 'v',
        parentId: 1
    }
]
}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
// deptList(parameter)
// .then(res => {
//     this.loading = false

```

```

// if (res.success == false) {
//   this.$message({
//     type: 'info',
//     message: res.msg
//   })
// } else {
//   this.listData = res.data
//   //
//   this.pageparm.currentPage = this.formInline.page
//   this.pageparm.pageSize = this.formInline.limit
//   this.pageparm.total = res.count
// }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = "
    this.editForm.deptId = row.deptId
    this.editForm.deptName = row.deptName
    this.editForm.deptNo = row.deptNo
  } else {
    this.title = "
    this.editForm.deptId = "
    this.editForm.deptName = "
    this.editForm.deptNo = "
  }
}

```

```

},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      deptSave(this.editForm)
        .then(res => {
          this.editFormVisible = false
          this.loading = false
          if (res.success) {
            this.getdata(this.formInline)
            this.$message({
              type: 'success',
              message: "
            })
          } else {
            this.$message({
              type: 'info',
              message: res.msg
            })
          }
        })
      .catch(err => {
        this.editFormVisible = false
        this.loading = false
        this.$message.error("")
      })
    } else {
      return false
    }
  })
},
//
deleteUser(index, row) {
  this.$confirm('?', "", {
    confirmButtonText: "",
    cancelButtonText: "",
    type: 'warning'
  })
  .then(() => {
    deptDelete(row.deptId)
      .then(res => {

```

```

        if (res.success) {
            this.$message({
                type: 'success',
                message: '!'
            })
            this.getdata(this.formInline)
        } else {
            this.$message({
                type: 'info',
                message: res.msg
            })
        }
    })
    .catch(err => {
        this.loading = false
        this.$message.error("")
    })
})
.catch(() => {
    this.$message({
        type: 'info',
        message: ""
    })
})
},
//
closeDialog() {
    this.editFormVisible = false
}
}
}
</script>

```

```

<style scoped>
.user-search {
    margin-top: 20px;
}
.userRole {
    width: 100%;
}
</style>

```

24:F:\git\nodejs\vue\vuestep\vueproject\src\views\system\Module.vue

```
/**
 *
 */
<template>
  <div>
    <!-- -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
      <el-breadcrumb-item></el-breadcrumb-item>
    </el-breadcrumb>
    <!-- -->
    <el-row>
      <!-- title -->
      <div class="modt-box"></div>
      <el-col :span="4"> </el-col>
      <el-col :span="6">
        <div class="mod-btnbox">
          <el-button size="small" type="primary"></el-button>
          <el-button size="small" type="primary"></el-button>
        </div>
        <el-tree class="treeclass" ref="tree" :data="treeData" default-expand-all=""
:props="defaultProps" @node-click="nodeclick" @check-change="handleClick" check-strictly
node-key="id" show-checkbox></el-tree>
      </el-col>
      <el-col :span="18">
        <div class="mod-btnbox">
          <el-button size="small" type="primary" icon="el-icon-plus" @click="addModule"></el-
button>
        </div>
        <el-form ref="form" :model="form" label-width="80px" :rules="rules">
          <el-form-item label="" prop="parentId">
            <el-select size="small" v-model="form.parentId" placeholder="" class="selectw">
              <el-option v-for="parm in fmenu" :key="parm.id" :label="parm.name"
:value="parm.id"></el-option>
            </el-select>
          </el-form-item>
          <el-form-item label="" prop="moduleName">
            <el-input size="small" v-model="form.moduleName"></el-input>
          </el-form-item>
        </el-form>
      </el-col>
    </el-row>
  </div>
</template>
```

```

    <el-form-item label="" prop="moduleIcon">
      <el-input size="small" v-model="form.moduleIcon"></el-input>
    </el-form-item>
    <el-form-item label="URL" prop="moduleUrl">
      <el-input size="small" v-model="form.moduleUrl"></el-input>
    </el-form-item>
    <el-form-item label="" prop="moduleOrder">
      <el-input size="small" v-model="form.moduleOrder"></el-input>
    </el-form-item>
    <el-form-item>
      <el-button size="small" type="primary" @click="saveModule('form')"></el-button>
      <el-button size="small" type="primary" v-show="showdelete" @click="deleteModule"></el-
button>
    </el-form-item>
  </el-form>
</el-col>
</el-row>
</div>
</template>

```

```

<script>
import {
  ModuleList,
  ModuleGet,
  ModuleNodes,
  ModuleSave,
  ModuleDelete
} from '../api/userMG'
export default {
  data() {
    return {
      showdelete: false,
      treeData: [],
      defaultProps: {
        children: 'children',
        label: 'name'
      },
    },
    form: {
      addUser: "",
      editUser: "",
      addTime: "",
      editTime: "",

```

```

    moduleId: "",
    parentId: "",
    moduleLevel: "",
    systemNo: "",
    isLeaf: "",
    fullIndex: "",
    moduleIcon: "",
    moduleOrder: "",
    moduleName: "",
    moduleNotes: "",
    moduleUrl: ""
  },
  // rules
  rules: {
    parentId: [
      { required: true, message: "", trigger: 'blur' }
    ],
    moduleName: [
      { required: true, message: "", trigger: 'blur' }
    ],
    moduleIcon: [
      { required: true, message: "", trigger: 'blur' }
    ],
    moduleUrl: [{ required: true, message: 'URL', trigger: 'blur' }],
    moduleOrder: [
      { required: true, message: "", trigger: 'blur' }
    ]
  },
  fmenu: []
}
},
/**
 *
 */

created() {
  this.getdata()
  this.getmenu()
},

/**
 *

```



```
*/  
methods: {  
  //  
  getdata() {  
    ModuleList()  
    .then(res => {  
      this.treeData = [  
        {  
          id: 1,  
          pld: 0,  
          name: "",  
          open: true,  
          checked: false  
        },  
        {  
          id: 2,  
          pld: 1,  
          name: "",  
          open: true,  
          checked: false  
        },  
        {  
          id: 3,  
          pld: 1,  
          name: "",  
          open: true,  
          checked: false  
        },  
        {  
          id: 12,  
          pld: 1,  
          name: "",  
          open: true,  
          checked: false  
        },  
        {  
          id: 13,  
          pld: 1,  
          name: "",  
          open: true,  
          checked: false  
        },  
      ]  
    })  
  }  
}
```

```
{
  id: 17,
  pld: 1,
  name: "",
  open: true,
  checked: false
},
{
  id: 4,
  pld: 2,
  name: "",
  open: true,
  checked: false
},
{
  id: 5,
  pld: 2,
  name: "",
  open: true,
  checked: false
},
{
  id: 6,
  pld: 2,
  name: "",
  open: true,
  checked: false
},
{
  id: 7,
  pld: 2,
  name: "",
  open: true,
  checked: false
}
]
})
.catch(err => {
  this.loading = false
  this.$message.error("")
})
},
```

```

//
addModule() {
  this.showdelete = false
  this.form.addUser = ""
  this.form.editUser = ""
  this.form.addTime = ""
  this.form.editTime = ""
  this.form.moduleId = ""
  this.form.parentId = ""
  this.form.moduleLevel = ""
  this.form.systemNo = ""
  this.form.isLeaf = ""
  this.form.fullIndex = ""
  this.form.moduleIcon = ""
  this.form.moduleOrder = ""
  this.form.moduleName = ""
  this.form.moduleNotes = ""
  this.form.moduleUrl = ""
},
//
getmenu() {
  let parm = {
    page: '1',
    limit: '10'
  }
  ModuleNodes(parm)
    .then(res => {
      this.fmenu = res.data
    })
    .catch(err => {
      this.loading = false
      this.$message.error("")
    })
},
//
handleClick(data, checked, node) {
  if (checked) {
    this.$refs.tree.setCheckedNodes([])
    this.$refs.tree.setCheckedNodes([data])
    this.showdelete = true
  } else {
  }
}

```

```

},
//
nodeclick(arr, node, self) {
  ModuleGet(arr.id)
    .then(res => {
      console.log(JSON.stringify(res))
      this.form = res.data.data
      this.$refs.tree.setCheckedNodes([])
      this.$refs.tree.setCheckedNodes([arr])
    })
    .catch(err => {
      this.loading = false
      this.$message.error("")
    })
},
//
saveModule(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      ModuleSave(this.form)
        .then(res => {
          this.getdata()
          this.getmenu()
        })
        .catch(err => {
          this.$message.error("")
        })
    } else {
      return false
    }
  })
},
//
deleteModule() {
  ModuleDelete(this.form.moduleId)
    .then(res => {
      this.getdata()
      this.getmenu()
      this.$message.error("")
    })
    .catch(err => {
      this.$message.error("")
    })
}

```

```
    })
  }
}
}
</script>
```

```
<style>
.user-search {
  margin-top: 20px !important;
}
.user-search td {
  margin-top: 20px !important;
}
.userRole {
  width: 100%;
}
.el-table__expanded-cell {
  padding: 0px !important;
  margin: 0px !important;
}
.modt-box {
  width: 100%;
  height: 60px;
  line-height: 60px;
  background-color: #f0f2f5;
  text-align: center;
  color: #334157;
  margin-top: 20px;
  margin-bottom: 20px;
  font-size: 24px;
}
.mod-btnbox {
  margin-bottom: 20px;
}
.treeclass {
  border: 1px solid #3f3f3f;
  padding-top: 20px;
  padding-bottom: 20px;
}
.selectw {
  width: 100%;
}
```

</style>

25:F:\git\nodejs\vue\vuestep\vueproject\src\views\system\Permission.vue

/**

*

*/

<template>

<div>

<!-- -->

<el-breadcrumb separator-class="el-icon-arrow-right">

<el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>

<el-breadcrumb-item></el-breadcrumb-item>

</el-breadcrumb>

<!-- -->

<el-form :inline="true" :model="formInline" class="user-search">

<el-form-item label="">

<el-input size="small" v-model="formInline.permissionName" placeholder=""></el-input>

</el-form-item>

<el-form-item label="">

<el-input size="small" v-model="formInline.permission" placeholder="CODE"></el-input>

</el-form-item>

<el-form-item label="">

<el-select size="small" v-model="formInline.roleId" placeholder="">

<el-option selected label="" value="0"></el-option>

<el-option v-for="parm in userparm" :key="parm.roleId" :label="parm.roleName"

:value="parm.roleId"></el-option>

</el-select>

</el-form-item>

<el-form-item>

<el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>

<el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>

<el-button size="small" type="primary" icon="el-icon-plus" @click="RolePermission()"></el-

button>

</el-form-item>

</el-form>

<!-- -->

<el-table size="small" @selection-change="selectChange" :data="listData" highlight-current-row

v-loading="loading" border element-loading-text="" style="width: 100%;">

<el-table-column align="center" type="selection" width="60">

</el-table-column>

```

<el-table-column sortable prop="permissionName" label="" width="300">
</el-table-column>
<el-table-column sortable prop="permission" label="CODE" width="300">
</el-table-column>
<el-table-column sortable prop="editTime" label="" width="300">
  <template slot-scope="scope">
    <div>{{scope.row.editTime|timestampToTime}}</div>
  </template>
</el-table-column>
<el-table-column sortable prop="editUser" label="" width="300">
</el-table-column>
<el-table-column align="center" label="" min-width="300">
  <template slot-scope="scope">
    <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
    <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click="closeDialog">
  <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
    <el-form-item label="" prop="permissionName">
      <el-input size="small" v-model="editForm.permissionName" auto-complete="off"
placeholder=""></el-input>
    </el-form-item>
    <el-form-item label="CODE" prop="permission">
      <el-input size="small" v-model="editForm.permission" auto-complete="off" placeholder="
CODE"></el-input>
    </el-form-item>
  </el-form>
  <div slot="footer" class="dialog-footer">
    <el-button size="small" @click="closeDialog"></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
  </div>
</el-dialog>
</div>
</template>

```

```

<script>
import {
  permissionList,
  ermissionSave,
  ermissionDelete,
  roleDropDown,
  RolePermission
} from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      editFormVisible: false, //
      title: "",
      editForm: {
        permissionId: "",
        permissionName: "",
        permission: "",
        token: localStorage.getItem('logintoken')
      },
      // rules
      rules: {
        permissionName: [
          { required: true, message: "", trigger: 'blur' }
        ],
        permission: [
          { required: true, message: 'CODE', trigger: 'blur' }
        ]
      },
      formInline: {
        page: 1,
        limit: 10,
        permissionName: "",
        permission: "",
        roleId: '0',
        token: localStorage.getItem('logintoken')
      },
      //
      selectdata: [],

```



```
    userparm: [], //
    listData: [], //
    //
    pageparm: {
        currentPage: 1,
        pageSize: 10,
        total: 10
    }
},
//
components: {
    Pagination
},
/**
 *
 */

watch: {},

/**
 *
 */

created() {
    this.getdata(this.formInline)
    this.getAccsee()
},

/**
 *
 */

methods: {
    //
    getdata(parameter) {
        this.loading = true
        //
        let res = {
            code: 0,
            msg: null,
            count: 0,
```

```
data: [
  {
    addUser: null,
    editUser: null,
    addTime: 1519728609000,
    editTime: 1522585700000,
    permissionId: 1,
    permissionName: '-',
    permission: 'system:User:list',
    lay_CHECKED: false,
    LAY_CHECKED: false
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1519728667000,
    editTime: 1522585706000,
    permissionId: 3,
    permissionName: '-',
    permission: 'system:User:save',
    lay_CHECKED: false,
    LAY_CHECKED: false
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1519728669000,
    editTime: 1522256096000,
    permissionId: 4,
    permissionName: '-',
    permission: 'system:User:delete',
    lay_CHECKED: false,
    LAY_CHECKED: false
  },
  {
    addUser: 'root',
    editUser: 'root',
    addTime: 1520249365000,
    editTime: 1522256085000,
    permissionId: 5,
    permissionName: '::',
    permission: 'system:Role:list',
```

```
lay_CHECKED: false,
LAY_CHECKED: false
},
{
  addUser: 'root',
  editUser: 'root',
  addTime: 1520249588000,
  editTime: 1520249588000,
  permissionId: 7,
  permissionName: 'system:Role:save',
  permission: 'system:Role:save',
  lay_CHECKED: false,
  LAY_CHECKED: false
},
{
  addUser: 'root',
  editUser: 'root',
  addTime: 1520249588000,
  editTime: 1520249588000,
  permissionId: 8,
  permissionName: 'system:Role:delete',
  permission: 'system:Role:delete',
  lay_CHECKED: false,
  LAY_CHECKED: false
},
{
  addUser: 'root',
  editUser: 'root',
  addTime: 1520251014000,
  editTime: 1520251014000,
  permissionId: 9,
  permissionName: 'system:Variable:',
  permission: 'system:Variable:list',
  lay_CHECKED: false,
  LAY_CHECKED: false
},
{
  addUser: 'root',
  editUser: 'root',
  addTime: 1520251014000,
  editTime: 1520251014000,
  permissionId: 11,
```

```

        permissionName: 'system:Variable:',
        permission: 'system:Variable:save',
        lay_CHECKED: false,
        LAY_CHECKED: false
    },
    {
        addUser: 'root',
        editUser: 'root',
        addTime: 1520251014000,
        editTime: 1520251014000,
        permissionId: 12,
        permissionName: 'system:Variable:',
        permission: 'system:Variable:delete',
        lay_CHECKED: false,
        LAY_CHECKED: false
    }
]
}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
//
// permissionList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page

```

```

//    this.pageparm.pageSize = this.formInline.limit
//    this.pageparm.total = res.count
//  }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
getAccsee() {
  roleDropDown()
  .then(res => {
    if (res.success == false) {
      this.$message({
        type: 'info',
        message: res.msg
      })
    } else {
      this.userparm = res.data
    }
  })
  .catch(err => {
    this.$message.error("")
  })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  //
  //this.editForm = Object.assign({}, row)
  if (row != undefined && row != 'undefined') {

```

```

    this.title = ""
    this.editForm.permissionId = row.permissionId
    this.editForm.permissionName = row.permissionName
    this.editForm.permission = row.permission
  } else {
    this.title = ""
    this.editForm.permissionId = ""
    this.editForm.permissionName = ""
    this.editForm.permission = ""
  }
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      permissionSave(this.editForm)
        .then(res => {
          this.editFormVisible = false
          this.loading = false
          if (res.success) {
            this.getdata(this.formInline)
            this.$message({
              type: 'success',
              message: ""
            })
          } else {
            this.$message({
              type: 'info',
              message: res.msg
            })
          }
        })
    }
    .catch(err => {
      this.editFormVisible = false
      this.loading = false
      this.$message.error("")
    })
  } else {
    return false
  }
})
},

```

```

//
deleteUser(index, row) {
  this.$confirm('?', '', {
    confirmButtonText: '',
    cancelButtonText: '',
    type: 'warning'
  })
  .then(() => {
    ermissionDelete(row.permissionId)
    .then(res => {
      if (res.success) {
        this.$message({
          type: 'success',
          message: ''
        })
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
  })
  .catch(err => {
    this.loading = false
    this.$message.error(',')
  })
})
.catch(() => {
  this.$message({
    type: 'info',
    message: ''
  })
})
},
//
selectChange(val) {
  this.selectdata = val
},
//
RolePermission() {
  let parms = {

```

```

    roleId: "",
    permissionIds: ""
  }
  if (this.formInline.roleId == '0') {
    this.$message({
      type: 'info',
      message: ""
    })
    return false
  }
  parms.roleId = this.formInline.roleId
  let len = this.selectdata
  let ids = []
  if (len != 0) {
    for (let i = 0; i < len.length; i++) {
      ids.push(len[i].permissionId)
    }
  }
  parms.permissionIds = ids.join(',')
  RolePermission(parms)
    .then(res => {
      if (res.success) {
        this.$message({
          type: 'success',
          message: ""
        })
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
    .catch(err => {
      this.loading = false
      this.$message.error(',')
    })
  },
  //
  closeDialog() {
    this.editFormVisible = false
  }
}

```



```
    }  
  }  
}  
</script>
```

```
<style scoped>  
.user-search {  
  margin-top: 20px;  
}  
.userRole {  
  width: 100%;  
}  
</style>
```

26:F:\git\nodejs\vue\vuestep\vueproject\src\views\system\Role.vue

```
/**  
 *  
 */  
<template>  
  <div>  
    <!-- -->  
    <el-breadcrumb separator-class="el-icon-arrow-right">  
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>  
      <el-breadcrumb-item></el-breadcrumb-item>  
    </el-breadcrumb>  
    <!-- -->  
    <el-form :inline="true" :model="formInline" class="user-search">  
  
      <el-form-item label="">  
        <el-input size="small" v-model="formInline.roleName" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item label="">  
        <el-input size="small" v-model="formInline.roleNo" placeholder=""></el-input>  
      </el-form-item>  
      <el-form-item>  
        <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>  
        <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>  
      </el-form-item>  
    </el-form>  
  <!------>
```

```

<el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-
loading-text="" style="width: 100%;">
  <el-table-column align="center" type="selection" width="60">
  </el-table-column>
  <el-table-column sortable prop="roleName" label="" width="300">
  </el-table-column>
  <el-table-column sortable prop="roleNo" label="" width="300">
  </el-table-column>
  <el-table-column sortable prop="editTime" label="" width="300">
    <template slot-scope="scope">
      <div>{{scope.row.editTime|timestampToTime}}</div>
    </template>
  </el-table-column>
  <el-table-column sortable prop="editUser" label="" width="300">
  </el-table-column>
  <el-table-column align="center" label="" min-width="300">
    <template slot-scope="scope">
      <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
      <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
      <el-button size="mini" type="success" @click="menuAccess(scope.$index, scope.row)">
</el-button>
    </template>
  </el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click='closeDialog("edit")'>
  <el-form label-width="120px" :model="editForm" ref="editForm" :rules="rules">
    <el-form-item label="" prop="systemNo">
      <el-input size="small" v-model="editForm.systemNo" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="roleName">
      <el-input size="small" v-model="editForm.roleName" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="roleNo">
      <el-input size="small" v-model="editForm.roleNo" auto-complete="off" placeholder=""></el-
input>
    </el-form-item>

```

```

</el-form>
<div slot="footer" class="dialog-footer">
  <el-button size="small" @click='closeDialog("edit")'></el-button>
  <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
</div>
</el-dialog>
<!-- -->
<el-dialog title="" :visible.sync="menuAccesssshow" width="30%" @click='closeDialog("perm")'>
  <el-tree ref="tree" default-expand-all="" :data="RoleRight" :props="RoleRightProps" :default-
checked-keys="checkmenu" node-key="id" show-checkbox>
    </el-tree>
    <div slot="footer" class="dialog-footer">
      <el-button size="small" @click='closeDialog("perm")'></el-button>
      <el-button size="small" type="primary" :loading="loading" class="title"
@click="menuPermSave"></el-button>
    </div>
  </el-dialog>
</div>
</template>

```

```

<script>
import {
  roleList,
  roleSave,
  roleDelete,
  rolePwd,
  RoleRightTree,
  RoleRightSave
} from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      editFormVisible: false, //
      menuAccesssshow: false, //
      title: "",
      editForm: {
        roleId: "",

```

```

    systemNo: "",
    roleNo: "",
    roleName: "",
    token: localStorage.getItem('logintoken')
  },
  // rules
  rules: {
    systemNo: [
      { required: true, message: "", trigger: 'blur' }
    ],
    roleNo: [
      { required: true, message: "", trigger: 'blur' }
    ],
    roleName: [
      { required: true, message: "", trigger: 'blur' }
    ]
  },
  formInline: {
    page: 1,
    limit: 10,
    varLable: "",
    varName: "",
    token: localStorage.getItem('logintoken')
  },
  //
  seletedata: {
    ids: "",
    token: localStorage.getItem('logintoken')
  },
  userparm: [], //
  listData: [], //
  //
  RoleRight: [],
  RoleRightProps: {
    children: 'children',
    label: 'name'
  },
  //
  checkmenu: [],
  //role
  saveroleId: "",
  //

```

```
    pageparm: {
      currentPage: 1,
      pageSize: 10,
      total: 10
    }
  },
  //
  components: {
    Pagination
  },
  /**
   *
  */

  watch: {},

  /**
   *
  */

  created() {
    this.getdata(this.formInline)
  },

  /**
   *
  */

  methods: {
    //
    getdata(parameter) {
      //
      let res = {
        code: 0,
        msg: null,
        count: 6,
        data: [
          {
            addUser: 'root',
            editUser: 'root',
            addTime: 1519182004000,
```

```
editTime: 1520288426000,
roleId: 1,
systemNo: 'pmd',
roleNo: 'Administrator',
roleName: "
},
{
  addUser: null,
  editUser: null,
  addTime: 1521111376000,
  editTime: 1520678191000,
  roleId: 2,
  systemNo: 'order',
  roleNo: 'admin',
  roleName: "
},
{
  addUser: null,
  editUser: null,
  addTime: 1520678221000,
  editTime: 1520678221000,
  roleId: 95,
  systemNo: 'pm',
  roleNo: 'common',
  roleName: "
},
{
  addUser: null,
  editUser: null,
  addTime: 1526349853000,
  editTime: 1526349853000,
  roleId: 96,
  systemNo: '1',
  roleNo: '1',
  roleName: '1'
},
{
  addUser: null,
  editUser: null,
  addTime: 1526349942000,
  editTime: 1526437443000,
  roleId: 97,
```

```

        systemNo: '2',
        roleNo: '2',
        roleName: '2'
    },
    {
        addUser: null,
        editUser: null,
        addTime: 1526652148000,
        editTime: 1526652148000,
        roleId: 101,
        systemNo: 'test',
        roleNo: 'demo',
        roleName: ''
    }
]
}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
// roleList(parameter)
// .then(res => {
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page
//     this.pageparm.pageSize = this.formInline.limit
//     this.pageparm.total = res.count
//   }

```

```

// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = ""
    this.editForm.roleId = row.roleId
    this.editForm.systemNo = row.systemNo
    this.editForm.roleNo = row.roleNo
    this.editForm.roleName = row.roleName
  } else {
    this.title = ""
    this.editForm.roleId = ""
    this.editForm.systemNo = ""
    this.editForm.roleNo = ""
    this.editForm.roleName = ""
  }
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {
      roleSave(this.editForm)
        .then(res => {
          this.editFormVisible = false
          this.loading = false
          if (res.success) {

```



```

        this.getdata(this.formInline)
        this.$message({
          type: 'success',
          message: "
        })
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
    .catch(err => {
      this.editFormVisible = false
      this.loading = false
      this.$message.error("")
    })
  } else {
    return false
  }
})
},
//
deleteUser(index, row) {
  this.$confirm('?', "", {
    confirmButtonText: "",
    cancelButtonText: "",
    type: 'warning'
  })
  .then(() => {
    roleDelete(row.roleId)
    .then(res => {
      if (res.success) {
        this.$message({
          type: 'success',
          message: "
        })
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg

```

```

        })
    }
})
.catch(err => {
    this.loading = false
    this.$message.error("")
})
})
.catch(() => {
    this.$message({
        type: 'info',
        message: ""
    })
})
},
//
menuAccess: function(index, row) {
    this.menuAccessshow = true
    this.saveroleId = row.roleId
    RoleRightTree(row.roleId)
    .then(res => {
        if (res.data.success) {
            this.$message({
                type: 'success',
                message: ""
            })
            this.changemenu(res.data.data)
            this.RoleRight = this.changeArr(res.data.data)
        } else {
            this.$message({
                type: 'info',
                message: res.data.msg
            })
        }
    })
    .catch(err => {
        this.loading = false
        this.$message.error("")
    })
},
//
changemenu(arr) {

```

```

let change = []
for (let i = 0; i < arr.length; i++) {
  if (arr[i].checked) {
    change.push(arr[i].id)
  }
}
this.checkmenu = change
},
// tree
changeArr(list) {
  var temptree = [],
    tree = [],
    items = []
  for (var i in list) {
    if (!temptree[list[i].id]) {
      var trow = {
        id: list[i].id,
        pld: list[i].pld,
        name: list[i].name,
        open: list[i].open,
        checked: list[i].checked,
        children: []
      }
      temptree[list[i].id] = trow
      items.push(trow)
    }
    if (list[i].uid > 0) {
      temptree[list[i].id]['children'].push({
        id: list[i].id,
        pld: list[i].pld,
        name: list[i].name,
        open: list[i].open,
        checked: list[i].checked,
        children: []
      })
    }
  }
}

for (var j in items) {
  if (temptree[items[j].pld]) {
    temptree[items[j].pld]['children'].push(temptree[items[j].id])
  } else {

```

```

        tree.push(temptree[items[j].id])
    }
}
temptree = null
items = null
return tree
},
// -
menuPermSave() {
    let parm = {
        roleId: this.saveroleId,
        moduleIds: "
    }
    let node = this.$refs.tree.getCheckedNodes()
    let moduleIds = []
    if (node.length != 0) {
        for (let i = 0; i < node.length; i++) {
            moduleIds.push(node[i].id)
        }
        parm.moduleIds = JSON.stringify(moduleIds)
    }
    RoleRightSave(parm)
    .then(res => {
        if (res.success) {
            this.$message({
                type: 'success',
                message: "
            })
            this.menuAccessshow = false
            this.getdata(this.formInline)
        } else {
            this.$message({
                type: 'info',
                message: res.msg
            })
        }
    })
    .catch(err => {
        this.loading = false
        this.$message.error("")
    })
},

```

```
//
closeDialog(dialog) {
  if (dialog == 'edit') {
    this.editFormVisible = false
  } else if (dialog == 'perm') {
    this.menuAccessshow = false
  }
}
}
}
}
</script>
```

```
<style scoped>
.user-search {
  margin-top: 20px;
}
.userRole {
  width: 100%;
}
</style>
```

27:F:\git\nodejs\vue\vuestep\vueproject\src\views\system\user.vue

```
/**
 *
 */
<template>
  <div>
    <!-- -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
      <el-breadcrumb-item></el-breadcrumb-item>
    </el-breadcrumb>
    <!-- -->
    <el-form :inline="true" :model="formInline" class="user-search">
      <el-form-item label="">
        <el-select size="small" v-model="formInline.isLock" placeholder="">
          <el-option label="" value=""></el-option>
          <el-option label="" value="N"></el-option>
          <el-option label="" value="Y"></el-option>
        </el-select>
```

```

</el-form-item>
<el-form-item label="">
  <el-input size="small" v-model="formInline.userName" placeholder=""></el-input>
</el-form-item>
<el-form-item label="">
  <el-input size="small" v-model="formInline.userMobile" placeholder=""></el-input>
</el-form-item>
<el-form-item>
  <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>
  <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>
  <el-button size="small" type="primary" @click="handleunit()"></el-button>
</el-form-item>
</el-form>
<!-->
<el-table size="small" @selection-change="selectChange" :data="userData" highlight-current-
row v-loading="loading" border element-loading-text="" style="width: 100%;">
  <el-table-column align="center" type="selection" width="50">
  </el-table-column>
  <el-table-column align="center" sortable prop="deptName" label="" width="120">
  </el-table-column>
  <el-table-column align="center" sortable prop="userName" label="" width="120">
  </el-table-column>
  <el-table-column align="center" sortable prop="userRealName" label="" width="120">
  </el-table-column>
  <el-table-column align="center" sortable prop="userMobile" label="" width="120">
  </el-table-column>
  <el-table-column align="center" sortable prop="userSex" label="" min-width="50">
  </el-table-column>
  <el-table-column align="center" sortable prop="userEmail" label="" min-width="120">
  </el-table-column>
  <el-table-column align="center" sortable prop="editTime" label="" min-width="120">
    <template slot-scope="scope">
      <div>{{scope.row.editTime|timestampToTime}}</div>
    </template>
  </el-table-column>
  <el-table-column align="center" sortable prop="isLock" label="" min-width="50">
    <template slot-scope="scope">
      <el-switch v-model="scope.row.isLock=='N'?nshow:fshow" active-color="#13ce66" inactive-
color="#ff4949" @change="editType(scope.$index, scope.row)">
      </el-switch>
    </template>
  </el-table-column>

```

```

<el-table-column label="" min-width="300">
  <template slot-scope="scope">
    <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
    <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
    <el-button size="mini" type="success" @click="resetpwd(scope.$index, scope.row)"></el-
button>
    <el-button size="mini" type="success" @click="dataAccess(scope.$index, scope.row)"></el-
button>
    <el-button size="mini" type="success" @click="offlineUser(scope.$index, scope.row)"></el-
button>
    <el-button size="mini" type="success" @click="refreshCache(scope.$index, scope.row)">
</el-button>
  </template>
</el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog :title="title" :visible.sync="editFormVisible" width="30%" @click='closeDialog("edit")'>
  <el-form label-width="80px" ref="editForm" :model="editForm" :rules="rules">
    <el-form-item label="" prop="userName">
      <el-input size="small" v-model="editForm.userName" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="userRealName">
      <el-input size="small" v-model="editForm.userRealName" auto-complete="off"
placeholder=""></el-input>
    </el-form-item>
    <el-form-item label="" prop="roleId">
      <el-select size="small" v-model="editForm.roleId" placeholder="" class="userRole">
        <el-option label="" value="1"></el-option>
        <el-option label="" value="2"></el-option>
      </el-select>
    </el-form-item>
    <el-form-item label="" prop="userMobile">
      <el-input size="small" v-model="editForm.userMobile" placeholder=""></el-input>
    </el-form-item>
    <el-form-item label="" prop="userEmail">
      <el-input size="small" v-model="editForm.userEmail" placeholder=""></el-input>
    </el-form-item>
    <el-form-item label="" prop="userSex">

```

```

    <el-radio v-model="editForm.userSex" label=""></el-radio>
    <el-radio v-model="editForm.userSex" label=""></el-radio>
  </el-form-item>
</el-form>
<div slot="footer" class="dialog-footer">
  <el-button size="small" @click='closeDialog("edit")'></el-button>
  <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
</div>
</el-dialog>
<!-- -->
<el-dialog title="" :visible.sync="dataAccessshow" width="30%" @click='closeDialog("perm")'>
  <el-tree ref="tree" default-expand-all="" :data="UserDept" :props="defaultProps" :default-
checked-keys="checkmenu" node-key="id" show-checkbox>
  </el-tree>
  <div slot="footer" class="dialog-footer">
    <el-button size="small" @click='closeDialog("perm")'></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="menuPermSave"></el-button>
  </div>
</el-dialog>
<!-- -->
<el-dialog title="" :visible.sync="unitAccessshow" width="30%" @click='closeDialog("unit")'>
  <el-tree ref="tree" default-expand-all="" :data="UserDept" :props="defaultProps" @check-
change="handleClick" :default-checked-keys="checkmenu" node-key="id" show-checkbox check-
strictly>
  </el-tree>
  <div slot="footer" class="dialog-footer">
    <el-button size="small" @click='closeDialog("unit")'></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="unitPermSave"></el-button>
  </div>
</el-dialog>
</div>
</template>

```

```

<script>
//
import {
  userList,
  userSave,
  userDelete,

```



```

userPwd,
userExpireToken,
userFlashCache,
userLock,
UserDeptTree,
UserDeptSave,
UserDeptdeptTree,
UserChangeDept
} from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      title: "",
      editFormVisible: false, //
      dataAccessshow: false, //
      unitAccessshow: false, //
      //
      editForm: {
        userId: "",
        userName: "",
        userRealName: "",
        roleId: "",
        userMobile: "",
        userEmail: "",
        userSex: "",
        token: localStorage.getItem('logintoken')
      },
      //
      unitparm: {
        userIds: "",
        deptId: "",
        deptName: ""
      },
      //
      selectdata: [],
      // rules
      rules: {
        userName: [

```

```

    { required: true, message: "", trigger: 'blur' }
  ],
  userRealName: [
    { required: true, message: "", trigger: 'blur' }
  ],
  roleId: [{ required: true, message: "", trigger: 'blur' }],
  userMobile: [
    { required: true, message: "", trigger: 'blur' },
    {
      pattern: /^1(3\d|47|5((?!4)\d)|7(0|1|[[6-8]])|8\d)\d{8,8}$/,
      required: true,
      message: "",
      trigger: 'blur'
    }
  ],
  userEmail: [
    { required: true, message: "", trigger: 'blur' },
    {
      pattern: /^[A-Za-z0-9\u4e00-\u9fa5]+\@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+$/,
      required: true,
      message: "",
      trigger: 'blur'
    }
  ],
  userSex: [{ required: true, message: "", trigger: 'blur' }
],
//
seletedata: {
  ids: "",
  token: localStorage.getItem('logintoken')
},
//
resetpsd: {
  userId: "",
  token: localStorage.getItem('logintoken')
},
//
offline: {
  token: localStorage.getItem('logintoken')
},
//
formInline: {

```

```
    page: 1,
    limit: 10,
    deptId: "",
    userName: "",
    userMobile: "",
    isLock: ""
  },
  //
  userData: [],
  //
  UserDept: [],
  defaultProps: {
    children: 'children',
    label: 'name'
  },
  //
  checkmenu: [],
  //role
  saveroleId: "",
  //
  pageparm: {
    currentPage: 1,
    pageSize: 10,
    total: 10
  }
}
},
//
components: {
  Pagination
},

/**
 *
 */
watch: {},

/**
 *
 */
created() {
  this.getdata(this.formInline)
```

```
},

/**
 *
 */
methods: {
  //
  getdata(parameter) {
    this.loading = true
    //
    let res = {
      code: 0,
      msg: null,
      count: 12,
      data: [
        {
          addUser: '1',
          editUser: '1',
          addTime: null,
          editTime: 1527411068000,
          userId: 1,
          systemNo: 'pmd',
          userName: 'root',
          userPassword: 'e10adc3949ba59abbe56e057f20f883e',
          userRealName: '',
          userSex: '',
          userMobile: '138123456789',
          userEmail: '111@qq.com',
          isLock: 'N',
          deptId: 1,
          deptName: 'xxxx',
          roleId: 1
        },
        {
          addUser: '1',
          editUser: '1',
          addTime: null,
          editTime: 1527410579000,
          userId: 3,
          systemNo: 'mc',
          userName: 'zengzhuo',
          userPassword: 'e10adc3949ba59abbe56e057f20f883e',
```

```
userRealName: "",
userSex: 'M',
userMobile: '18616988966',
userEmail: '222@qq.com',
isLock: 'N',
deptId: 2,
deptName: 'xxxx',
roleId: 101
},
{
  addUser: '1',
  editUser: '4',
  addTime: null,
  editTime: 1527411586000,
  userId: 4,
  systemNo: 'ec',
  userName: 'admin',
  userPassword: '59ba8b7dda9ff79186311a5a9fa155ca',
  userRealName: "",
  userSex: "",
  userMobile: '138123456789',
  userEmail: 'huangxuekun@founder.com',
  isLock: 'N',
  deptId: 2,
  deptName: 'xxxx',
  roleId: 3
},
{
  addUser: null,
  editUser: null,
  addTime: 1526275128000,
  editTime: 1526284402000,
  userId: 28,
  systemNo: null,
  userName: 'eee111',
  userPassword: 'e10adc3949ba59abbe56e057f20f883e',
  userRealName: '123111',
  userSex: "",
  userMobile: '12354342345',
  userEmail: '111232@qq.com',
  isLock: 'N',
  deptId: 4,
```

```
deptName: 'zxxxxx',
roleId: 1
},
{
  addUser: null,
  editUser: null,
  addTime: 1526284533000,
  editTime: 1526284533000,
  userId: 37,
  systemNo: null,
  userName: 'ces',
  userPassword: 'e10adc3949ba59abbe56e057f20f883e',
  userRealName: 'sesfg',
  userSex: '',
  userMobile: '12312312312',
  userEmail: '1221111111',
  isLock: 'N',
  deptId: 1,
  deptName: 'xxxx',
  roleId: 1
},
{
  addUser: null,
  editUser: null,
  addTime: 1526285228000,
  editTime: 1526285228000,
  userId: 43,
  systemNo: null,
  userName: '22',
  userPassword: 'e10adc3949ba59abbe56e057f20f883e',
  userRealName: '22',
  userSex: '',
  userMobile: '222',
  userEmail: '222',
  isLock: 'N',
  deptId: 1,
  deptName: 'xxxx',
  roleId: 1
},
{
  addUser: null,
  editUser: null,
```

```
    addTime: 1526448593000,
    editTime: 1526448593000,
    userId: 58,
    systemNo: null,
    userName: '1',
    userPassword: 'e10adc3949ba59abbe56e057f20f883e',
    userRealName: '1',
    userSex: '',
    userMobile: '13607118810',
    userEmail: '1@qq.com',
    isLock: 'N',
    deptId: 1,
    deptName: 'xxxx',
    roleId: 1
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1526452698000,
    editTime: 1526520341000,
    userId: 60,
    systemNo: null,
    userName: '222222222',
    userPassword: 'e10adc3949ba59abbe56e057f20f883e',
    userRealName: '222222222222',
    userSex: '',
    userMobile: '13607118810',
    userEmail: '111@qq.com',
    isLock: 'N',
    deptId: 1,
    deptName: 'xxxx',
    roleId: 1
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1526452731000,
    editTime: 1526452731000,
    userId: 61,
    systemNo: null,
    userName: '33333333333333',
    userPassword: 'e10adc3949ba59abbe56e057f20f883e',
```

```

        userRealName: '44444444444444444444',
        userSex: '',
        userMobile: '13607118810',
        userEmail: 'qqq@qq.com',
        isLock: 'N',
        deptId: 1,
        deptName: 'xxxx',
        roleId: 2
    },
    {
        addUser: null,
        editUser: null,
        addTime: 1526452756000,
        editTime: 1527128981000,
        userId: 62,
        systemNo: null,
        userName: '211111111',
        userPassword: 'e10adc3949ba59abbe56e057f20f883e',
        userRealName: '21111111111',
        userSex: '',
        userMobile: '13601478451',
        userEmail: '222222@qq.com',
        isLock: 'N',
        deptId: 17,
        deptName: 'v',
        roleId: 2
    }
]
}
this.loading = false
this.userData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
//
// userList(parameter).then(res => {

```



```

// this.loading = false
// if (res.success == false) {
//   this.$message({
//     type: 'info',
//     message: res.msg
//   })
// } else {
//   this.userData = res.data
//   //
//   this.pageparm.currentPage = this.formInline.page
//   this.pageparm.pageSize = this.formInline.limit
//   this.pageparm.total = res.count
// }
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {
  this.getdata(this.formInline)
},
// type
editType: function(index, row) {
  this.loading = true
  let parm = {
    lock: "",
    userId: "",
    token: localStorage.getItem('logintoken')
  }
  parm.userId = row.userId
  let lock = row.isLock
  if (lock == 'N') {
    parm.lock = 'Y'
  } else {
    parm.lock = 'N'
  }
  //
  userLock(parm).then(res => {

```

```

this.loading = false
if (res.success == false) {
  this.$message({
    type: 'info',
    message: res.msg
  })
} else {
  this.$message({
    type: 'success',
    message: "
  })
  this.getdata(this.formInline)
}
})
},
//
handleEdit: function(index, row) {
  this.editFormVisible = true
  if (row != undefined && row != 'undefined') {
    this.title = "
    this.editForm.userId = row.userId
    this.editForm.userName = row.userName
    this.editForm.userRealName = row.userRealName
    this.editForm.roleId = row.roleId
    this.editForm.userMobile = row.userMobile
    this.editForm.userEmail = row.userEmail
    this.editForm.userSex = row.userSex
  } else {
    this.title = "
    this.editForm.userId = "
    this.editForm.userName = "
    this.editForm.userRealName = "
    this.editForm.roleId = "
    this.editForm.userMobile = "
    this.editForm.userEmail = "
    this.editForm.userSex = "
  }
},
//
submitForm(editData) {
  this.$refs[editData].validate(valid => {
    if (valid) {

```

```

//
userSave(this.editForm)
.then(res => {
  this.editFormVisible = false
  this.loading = false
  if (res.success) {
    this.getdata(this.formInline)
    this.$message({
      type: 'success',
      message: "
    })
  } else {
    this.$message({
      type: 'info',
      message: res.msg
    })
  }
})
.catch(err => {
  this.editFormVisible = false
  this.loading = false
  this.$message.error("")
})
} else {
  return false
}
})
},
//
handleunit: function(index, row) {
  this.unitAccessshow = true
  let parms = 0
  UserDeptdeptTree(parms)
  .then(res => {
    if (res.data.success) {
      this.UserDept = this.changeArr(res.data.data)
    } else {
      this.$message({
        type: 'info',
        message: res.msg
      })
    }
  })
}

```

```

    })
    .catch(err => {
      this.loading = false
      this.$message.error(',')
    })
  },
  handleClick(data, checked, node) {
    if (checked) {
      this.$refs.tree.setCheckedNodes([])
      this.$refs.tree.setCheckedNodes([data])
      this.unitparm.deptId = data.id
      this.unitparm.deptName = data.name
      //
    } else {
    }
  },
  //
  unitPermSave() {
    let len = this.selectdata
    let ids = []
    if (len != 0) {
      for (let i = 0; i < len.length; i++) {
        ids.push(len[i].userId)
      }
    }
    this.unitparm.userIds = ids.join(',')
    UserChangeDept(this.unitparm)
    .then(res => {
      this.unitAccessshow = false
      if (res.success) {
        this.$message({
          type: 'success',
          message: "
        })
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
  }
})

```

```

        .catch(err => {
            this.loading = false
            this.$message.error(',')
        })
    },
    //
    selectChange(val) {
        this.selectdata = val
    },
    //
    closeDialog(dialog) {
        if (dialog == 'edit') {
            this.editFormVisible = false
        } else if (dialog == 'perm') {
            this.dataAccessshow = false
        } else if (dialog == 'unit') {
            this.unitAccessshow = false
        }
    },
    //
    deleteUser(index, row) {
        this.$confirm('?', '', {
            confirmButtonText: '',
            cancelButtonText: '',
            type: 'warning'
        })
        .then(() => {
            //
            userDelete(row.id)
                .then(res => {
                    if (res.success) {
                        this.$message({
                            type: 'success',
                            message: '!'
                        })
                        this.getdata(this.formInline)
                    } else {
                        this.$message({
                            type: 'info',
                            message: res.msg
                        })
                    }
                })
        })
    }
}

```

```

    })
    .catch(err => {
      this.loading = false
      this.$message.error("")
    })
  })
  .catch(() => {
    this.$message({
      type: 'info',
      message: ""
    })
  })
},
//
resetpwd(index, row) {
  this.resetpsd.userId = row.userId
  this.$confirm('?', "", {
    confirmButtonText: "",
    cancelButtonText: "",
    type: 'warning'
  })
  .then(() => {
    userPwd(this.resetpsd)
    .then(res => {
      if (res.success) {
        this.$message({
          type: 'success',
          message: ""
        })
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
  })
  .catch(err => {
    this.loading = false
    this.$message.error("")
  })
})

```

```

        .catch(() => {
            this.$message({
                type: 'info',
                message: "
            })
        })
    },
    //
    dataAccess: function(index, row) {
        this.dataAccessshow = true
        this.saveroleId = row.userId
        UserDeptTree(row.userId)
        .then(res => {
            if (res.data.success) {
                this.checkmenu = this.changemenu(res.data.data)
                this.UserDept = this.changeArr(res.data.data)
            } else {
                this.$message({
                    type: 'info',
                    message: res.data.msg
                })
            }
        })
        .catch(err => {
            this.loading = false
            this.$message.error("")
        })
    },
    //
    changeArr(data) {
        var pos = {}
        var tree = []
        var i = 0
        while (data.length != 0) {
            if (data[i].pId == 0) {
                tree.push({
                    id: data[i].id,
                    name: data[i].name,
                    pId: data[i].pId,
                    open: data[i].open,
                    checked: data[i].checked,
                    children: []
                })
            }
            data.splice(i, 1)
            i = 0
        }
    }
}

```

```

    })
    pos[data[i].id] = [tree.length - 1]
    data.splice(i, 1)
    i--
  } else {
    var posArr = pos[data[i].pId]
    if (posArr !== undefined) {
      var obj = tree[posArr[0]]
      for (var j = 1; j < posArr.length; j++) {
        obj = obj.children[posArr[j]]
      }

      obj.children.push({
        id: data[i].id,
        name: data[i].name,
        pId: data[i].pId,
        open: data[i].open,
        checked: data[i].checked,
        children: []
      })
      pos[data[i].id] = posArr.concat([obj.children.length - 1])
      data.splice(i, 1)
      i--
    }
  }
  i++
  if (i > data.length - 1) {
    i = 0
  }
}
return tree
},
//
changemenu(arr) {
  let change = []
  for (let i = 0; i < arr.length; i++) {
    if (arr[i].checked) {
      change.push(arr[i].id)
    }
  }
  return change
},

```



```

// -
menuPermSave() {
  let parm = {
    userId: this.saveroleId,
    deptIds: ""
  }
  let node = this.$refs.tree.getCheckedNodes()
  let moduleIds = []
  if (node.length !== 0) {
    for (let i = 0; i < node.length; i++) {
      moduleIds.push(node[i].id)
    }
    parm.deptIds = JSON.stringify(moduleIds)
  }
  UserDeptSave(parm)
    .then(res => {
      if (res.success) {
        this.$message({
          type: 'success',
          message: ""
        })
        this.dataAccessshow = false
        this.getdata(this.formInline)
      } else {
        this.$message({
          type: 'info',
          message: res.msg
        })
      }
    })
    .catch(err => {
      this.loading = false
      this.$message.error("")
    })
  },
  //
  offlineUser(index, row) {
    this.$confirm("" + row.userName + '?', "", {
      confirmButtonText: "",
      cancelButtonText: "",
      type: 'warning'
    })
  }
}

```

```

.then(() => {
  userExpireToken(row.userName)
  .then(res => {
    if (res.success) {
      this.$message({
        type: 'success',
        message: " + row.userName + "
      })
      this.getdata(this.formInline)
    } else {
      this.$message({
        type: 'info',
        message: res.msg
      })
    }
  })
  .catch(err => {
    this.loading = false
    this.$message.error("")
  })
})
.catch(() => {
  this.$message({
    type: 'info',
    message: ""
  })
})
},
//
refreshCache(index, row) {
  userFlashCache(row.userName)
  .then(res => {
    if (res.success) {
      this.$message({
        type: 'success',
        message: ""
      })
      this.getdata(this.formInline)
    } else {
      this.$message({
        type: 'info',
        message: res.msg
      })
    }
  })
}

```

```

    })
  }
})
.catch(err => {
  this.loading = false
  this.$message.error("")
})
}
}
}
</script>

```

```

<style scoped>
.user-search {
  margin-top: 20px;
}
.userRole {
  width: 100%;
}
</style>

```

28:F:\git\nodejs\vue\vuestep\vueproject\src\views\system\Variable.vue

```

/**
 *
 */
<template>
  <div>
    <!-- -->
    <el-breadcrumb separator-class="el-icon-arrow-right">
      <el-breadcrumb-item :to="{ path: '/' }"></el-breadcrumb-item>
      <el-breadcrumb-item></el-breadcrumb-item>
    </el-breadcrumb>
    <!-- -->
    <el-form :inline="true" :model="formInline" class="user-search">

      <el-form-item label="">
        <el-input size="small" v-model="formInline.varLable" placeholder=""></el-input>
      </el-form-item>
      <el-form-item label="">
        <el-input size="small" v-model="formInline.varName" placeholder=""></el-input>

```

```

</el-form-item>
<el-form-item>
  <el-button size="small" type="primary" icon="el-icon-search" @click="search"></el-button>
  <el-button size="small" type="primary" icon="el-icon-plus" @click="handleEdit()"></el-button>
</el-form-item>
</el-form>
<!-->
<el-table size="small" :data="listData" highlight-current-row v-loading="loading" border element-
loading-text="" style="width: 100%;">
  <el-table-column align="center" type="selection" width="60">
</el-table-column>
  <el-table-column sortable prop="varLable" label="" width="200">
</el-table-column>
  <el-table-column sortable prop="varName" label="" width="200">
</el-table-column>
  <el-table-column sortable prop="varValue" label="" width="300">
</el-table-column>
  <el-table-column sortable prop="editTime" label="" width="200">
    <template slot-scope="scope">
      <div>{{scope.row.editTime|timestampToTime}}</div>
    </template>
</el-table-column>
  <el-table-column sortable prop="editUser" label="" width="200">
</el-table-column>
  <el-table-column align="center" label="" min-width="300">
    <template slot-scope="scope">
      <el-button size="mini" @click="handleEdit(scope.$index, scope.row)"></el-button>
      <el-button size="mini" type="danger" @click="deleteUser(scope.$index, scope.row)"></el-
button>
    </template>
  </el-table-column>
</el-table>
<!-- -->
<Pagination v-bind:child-msg="pageparm" @callFather="callFather"></Pagination>
<!-- -->
<el-dialog title="" :visible.sync="editFormVisible" width="30%" @click="closeDialog">
  <el-form label-width="120px" :model="editForm" :rules="rules" ref="editForm">
    <el-form-item label="" prop="varLable">
      <el-input size="small" v-model="editForm.varLable" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="varName">

```

```

        <el-input size="small" v-model="editForm.varName" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
    <el-form-item label="" prop="varValue">
        <el-input size="small" v-model="editForm.varValue" auto-complete="off" placeholder="
"></el-input>
    </el-form-item>
</el-form>
<div slot="footer" class="dialog-footer">
    <el-button size="small" @click="closeDialog"></el-button>
    <el-button size="small" type="primary" :loading="loading" class="title"
@click="submitForm('editForm')"></el-button>
</div>
</el-dialog>
</div>
</template>

```

```

<script>
import { variableList, variableSave, variableDelete } from '../api/userMG'
import Pagination from '../components/Pagination'
export default {
  data() {
    return {
      nshow: true, //switch
      fshow: false, //switch
      loading: false, //
      editFormVisible: false, //
      title: "",
      editForm: {
        varId: "",
        varLable: "",
        varName: "",
        varValue: "",
        token: localStorage.getItem('logintoken')
      },
      // rules
      rules: {
        varLable: [
          { required: true, message: "", trigger: 'blur' }
        ],
        varName: [
          { required: true, message: "", trigger: 'blur' }
        ]
      }
    }
  }
}

```

```

    ],
    varValue: [
        { required: true, message: "", trigger: 'blur' }
    ]
},
formInline: {
    page: 1,
    limit: 10,
    varLable: "",
    varName: "",
    token: localStorage.getItem('logintoken')
},
//
seletedata: {
    ids: "",
    token: localStorage.getItem('logintoken')
},
userparm: [], //
listData: [], //
//
pageparm: {
    currentPage: 1,
    pageSize: 10,
    total: 10
}
}
},
//
components: {
    Pagination
},
/**
 *
 */

watch: {},

/**
 *
 */

created() {

```

```

    this.getdata(this.formInline)
  },

  /**
   *
  */

  methods: {
    //
    getdata(parameter) {
      this.loading = true
      //
      let res = {
        code: 0,
        msg: null,
        count: 6,
        data: [
          {
            addUser: 'root',
            editUser: 'root',
            addTime: 1519385529000,
            editTime: 1520619747000,
            varId: 190,
            deptId: 1,
            varLable: "",
            varName: 'domainName',
            varValue: 'http://127.0.0.1:8080/iot-service-pay'
          },
          {
            addUser: 'root',
            editUser: 'root',
            addTime: 1519451541000,
            editTime: 1519451547000,
            varId: 191,
            deptId: 1,
            varLable: "",
            varName: 'productImgURL',
            varValue: 'http://ymwllw.com/pic'
          },
          {
            addUser: 'root',
            editUser: 'root',

```

```
    addTime: 1519452658000,
    editTime: 1519452661000,
    varId: 192,
    deptId: 1,
    varLabel: "",
    varName: 'groupDomain',
    varValue: 'http://ymwlwl.com'
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1519879624000,
    editTime: 1519879628000,
    varId: 193,
    deptId: 1,
    varLabel: "",
    varName: 'envType',
    varValue: 'false'
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1526374014000,
    editTime: 1526374031000,
    varId: 198,
    deptId: null,
    varLabel: 'sgh',
    varName: 'jhjj',
    varValue: 'jhjhgs'
  },
  {
    addUser: null,
    editUser: null,
    addTime: 1526453243000,
    editTime: 1526453243000,
    varId: 199,
    deptId: null,
    varLabel: 'v',
    varName: 'v',
    varValue: 'v'
  }
]
```



```

}
this.loading = false
this.listData = res.data
//
this.pageparm.currentPage = this.formInline.page
this.pageparm.pageSize = this.formInline.limit
this.pageparm.total = res.count
//

/**
 *
 */
// variableList(parameter)
// .then(res => {
//   console.log(JSON.stringify(res))
//   this.loading = false
//   if (res.success == false) {
//     this.$message({
//       type: 'info',
//       message: res.msg
//     })
//   } else {
//     this.listData = res.data
//     //
//     this.pageparm.currentPage = this.formInline.page
//     this.pageparm.pageSize = this.formInline.limit
//     this.pageparm.total = res.count
//   }
// })
// .catch(err => {
//   this.loading = false
//   this.$message.error("")
// })
},
//
callFather(parm) {
  this.formInline.page = parm.currentPage
  this.formInline.limit = parm.pageSize
  this.getdata(this.formInline)
},
//
search() {

```

```

    this.getdata(this.formInline)
  },
  //
  handleEdit: function(index, row) {
    this.editFormVisible = true
    if (row != undefined && row != 'undefined') {
      this.title = "
      this.editForm.varId = row.varId
      this.editForm.varLable = row.varLable
      this.editForm.varName = row.varName
      this.editForm.varValue = row.varValue
    } else {
      this.title = "
      this.editForm.varId = "
      this.editForm.varLable = "
      this.editForm.varName = "
      this.editForm.varValue = "
    }
  },
  //
  submitForm(editData) {
    this.$refs[editData].validate(valid => {
      if (valid) {
        variableSave(this.editForm)
          .then(res => {
            this.editFormVisible = false
            this.loading = false
            if (res.success) {
              this.getdata(this.formInline)
              this.$message({
                type: 'success',
                message: "
              })
            } else {
              this.$message({
                type: 'info',
                message: res.msg
              })
            }
          })
      }
    })
    .catch(err => {
      this.editFormVisible = false

```

```

        this.loading = false
        this.$message.error("")
    })
} else {
    return false
}
})
},
//
deleteUser(index, row) {
    this.$confirm('?', "", {
        confirmButtonText: "",
        cancelButtonText: "",
        type: 'warning'
    })
    .then(() => {
        variableDelete(row.varId)
        .then(res => {
            if (res.success) {
                this.$message({
                    type: 'success',
                    message: ""
                })
                this.getdata(this.formInline)
            } else {
                this.$message({
                    type: 'info',
                    message: res.msg
                })
            }
        })
    })
    .catch(err => {
        this.loading = false
        this.$message.error("")
    })
})
.catch(() => {
    this.$message({
        type: 'info',
        message: ""
    })
})
})

```

```
    },  
    //  
    closeDialog() {  
        this.editFormVisible = false  
    }  
}  
}  
</script>
```

```
<style scoped>  
.user-search {  
    margin-top: 20px;  
}  
.userRole {  
    width: 100%;  
}  
</style>
```