菜单,tabs,iframe

2019年11月25日

15:25

* 1. **template**

 <Menu

          id="myMenu"

          v-if="isMenu"

          :open-names="openedNames"

          :active-name="activeName"

          width="101%"

          class="style\_overflow"

          style="overflow: auto;position: relative;background-color: #008CF8;"

          accordion

          @on-select="handleSelection"

        >

          <div>

            <Submenu v-for="(item,index) in menuData" :name="item.code" :key="index">

              <template slot="title">

                <div class="item-one" v-if="!isCollapsed" :title="item.label">

                  <img :src="item.src" style="width: 14px;margin-bottom: -2px;margin-right: 5px" />

                  {{ item.label }}

                </div>

              </template>

              <Submenu

                v-for="(me,index\_me) in item.menuitems"

                :name="me.code"

                :key="index\_me"

                class="menu-two"

              >

                <template slot="title">

                  <div class="item-two" :title="me.label">{{ me.label }}</div>

                </template>

                <div v-if="!isCollapsed" class="menu-three">

                  <menu-item

                    ref="menu"

                    style="padding-left: 90px!important;"

                    v-for="(m,index\_m) in me.meitem"

                    :name="m.name"

                    :key="index\_m"

                    :id="'menu'+index\_m"

                    @click="handleSelection"

                  >{{ m.label }}</menu-item>

                </div>

              </Submenu>

            </Submenu>

          </div>

        </Menu>

* 1. **获取后台数据,并对后台数据进行处理(按数据的level),注意----menuData表示侧栏菜单数据,tabData表示标签页数据**

 getData() {

      var doList = [];

      localStorage.setItem("businessGateway", 11);

      getBusinessMenu({

        deptId: 11,

        userId: localStorage.loginName

      }).then(res => {

        doList = res.data;

        for (var i = 0; i < doList.length; i++) {

          var iconNum = i % 5;

          if (doList[i].level == "1") {

            this.menuData.push({

              label: doList[i].alias,

              name: doList[i].id,

              code: doList[i].code,

              level: doList[i].level,

              remark: doList[i].remark,

              menuitems: [],

              src: require("./../../../assets/images/appIntegration/" +

                doList[i].icon) //添加图标

            });

          }

        }

        for (var i = 0; i < doList.length; i++) {

          if (doList[i].level == "2") {

            for (var z = 0; z < this.menuData.length; z++) {

              // for(var x=0;x<doList.length;x++){

              if (doList[i].pCode === this.menuData[z].code) {

                this.menuData[z].menuitems.push({

                  label: doList[i].alias,

                  pCode: doList[i].pCode,

                  code: doList[i].code,

                  remark: doList[i].remark,

                  name: doList[i].id,

                  level: doList[i].level,

                  meitem: []

                });

                // }

              }

            }

          }

        }

        for (var i = 0; i < doList.length; i++) {

          if (doList[i].level == "3") {

            this.mainTabs.push({

              label: doList[i].alias,

              name: doList[i].id,

              show: false

            });

            for (var m = 0; m < this.menuData.length; m++) {

              for (var b = 0; b < this.menuData[m].menuitems.length; b++) {

                if (doList[i].pCode == this.menuData[m].menuitems[b].code) {

                  this.menuData[m].menuitems[b].meitem.push({

                    label: doList[i].alias,

                    pCode: doList[i].pCode,

                    code: doList[i].code,

                    remark: doList[i].remark,

                    name: doList[i].id,

                    level: doList[i].level

                  });

                }

              }

            }

          }

        }

        this.isMenu = true;

        this.$nextTick(() => {

            this.$refs.menu[0].$el.click();

        });

      });

    },

* 1. **标签页 ,标签页通过item.show决定是否渲染,当点击侧栏菜单时,使对应的item.show为true,移除卡片时,使对应的item.show为false;**

      <Tabs type="card" @on-tab-remove="removeTab" @on-click="clickTab" :value="activeTab">

        <template v-for="(item,index) in mainTabs">

          <tab-pane :label="item.label" closable : name= "item.name" :key="index" v-if="item.show">

            <iframe

              frameborder="0"

              width="100%"

              :height="iframeHeight"

              marginheight="0"

              scrolling="no"

              marginwidth="0"

              :src="src"

              :ref="item.name"

              style="border: 0px solid rgb(227, 227, 227);padding:0px;"

            ></iframe>

          </tab-pane>

        </template>

      </Tabs>

**4.iframe**

 <iframe

              frameborder="0"//边框

              width="100%"//宽

              :height="iframeHeight"//高

              marginheight="0"//上边距

              scrolling="no"//是否可滑动

              marginwidth="0"//左边距

              :src="src"//路径

              :ref="item.name"//ref

              style="border: 0px solid rgb(227, 227, 227);padding:0px;"

            ></iframe>

**iframe的父子通信 :**

**mounted:**

this.$nextTick(() => {

window.addEventListener("message", this.handleMessage);

});

**destroyed:**

window.removeEventListener("message", this.handleMessage);

**methods:**

//监听iframe事件

handleMessage(event) {

const data = event.data;

switch (data.cmd) {

case "add":

break;

case "del":

break;

}

},

**iframe->function**

window.parent.postMessage({

cmd: "del"

});

* 1. **原生页面父子组件间通信**
     1. **触发事件**

// 创建事件

var evt = document.createEvent('Event');

// 定义事件类型

evt.initEvent('tableEvent', true, true);

$("#delete\_tFyTicket")[0].dispatchEvent(evt);

* 1. **绑定事件**

window.addEventListener("tableEvent",handleFyAccount);