//使用时将ui-xxx替换为Xxx（使用iview框架时）

<template>

<ui-row>

<ui-col span="2">

<ui-checkbox v-model="checked" size="large" @on-change="handleCheckChange"></ui-checkbox>

</ui-col>

<ui-col span="21" offset="1">

<ui-select

v-model="selected"

:disabled="disabled"

:multiple="multiple"

clearable

:filterable="filterable"

:loading="loading"

:remote="remote"

:remote-method="remoteMethod"

@on-change="handleSelectChange">

<ui-option v-for="item in list" :value="item.value" :label="item.label" :key="item.value">

<span>{{item.label}}</span>

</ui-option>

</ui-select>

</ui-col>

</ui-row>

</template>

<script type="text/ecmascript-6">

export default {

name: 'complexSelect',

props: {

list: Array,

value: String,

multiple: Boolean,

filterable: Boolean,

loading: Boolean,

remote: Boolean,

remoteMethod: Function,

isCancle: Boolean

},

data () {

return {

checked: false,

selected: [],

disabled: true,

data: this.value,

answer: ':',

tempSelect: ''

};

},

methods: {

handleCheckChange (checked) {

if (checked) {

this.$emit('update:value', this.getValue()); //使用双向绑定的 :value.sync=""方法实现update:value

this.disabled = false;

} else {

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

this.selected = [];

this.disabled = true;

}

},

handleSelectChange (item) {

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

},

getValue () {

let checked = this.checked ? 1 : 0;

return checked + ':0:' + this.selected + this.answer;

},

setValue (val) {

let vals = val.split(':');

if (vals.length === 3) {

this.checked = Boolean(vals[0]);

let tempValue = vals[2].split(',');

this.selected = tempValue;

}

}

},

created () {

this.setValue(this.data);

},

mounted () {

if (this.checked) {

this.disabled = false;

}

if(this.value!=''&&this.value.length==1){

this.checked = true;

this.disabled = false;

this.selected = this.value;

}

if(this.value!=''&&this.value.length>1){

this.tempSelect = this.value;

}

},

watch: {

isCancle: function(newValue,oldValue){

if(newValue){

this.checked = false;

this.selected = [];

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

this.disabled = true;

}

},

// 通过监听select判断修改方式

selected:function (newSelect,oldSelect) {

let vals = this.tempSelect.split(':');

let tempValue = '';

if (vals.length === 3 ||vals.length === 4) {

tempValue = vals[2].split(',');

}

if (tempValue.toString() == '' && newSelect.toString() != ''){

this.answer = ':A';

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

}

if (tempValue.toString() != '' && newSelect.toString() != '' && newSelect.toString() != tempValue){

this.answer = ':U'

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

}

if (oldSelect.toString() != '' && newSelect.toString() == ''){

this.answer = ':D'

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

}

if (tempValue.toString() == newSelect.toString()){

this.answer = ':'

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

}

}

}

};

</script>