组件和路由

2019年11月1日

8:29

**全局组件注册:**

在main.js文件中

导入:

import myComponent from './components/footer'

注册:

Vue.component('firstComponent', myComponent)

**全局组件使用:**

<firstComponent></firstComponent>

**局部组件注册:**

在页面下的script标签下:

import FootNav from "../components/footer";

在export default下:

components:{

        'foot-nav':FootNav

    }

**局部组件使用:**

<foot-nav></foot-nav>

注意:不要使用关键字,类似footer

组件的样式会加上类似[data-v-b45e332e]的选择器,该值是随机的,不能在父组件中写该选择器

加强样式权重

**父子组件之间传参:**

父组件传值给子组件:

子组件:

  props: {

    childValue: {

      type: String,

      default: "this is from child.vue"

    }

父组件: <child :childValue="parentValue"></child>

注意: 父传子参数要写到props属性中

子组件传值给父组件(通过方法的参数传值):

父组件: <child @getChildValue="showValue"></child>

子组件: this.$emit("getChildValue", this.dataValue);

**兄弟组件之间参数:**

1.兄弟组件引入相同的bus.js作为中间仓库

//bus.js内容

import Vue from 'vue'

export default new Vue()

2.兄弟组件都引入到父组件中

import child from "./demo\_child.vue";

import another from "./demo\_another"

3.在需要接收参数的组件的钩子函数mounted中,添加监听事件

  mounted: function() {

    let \_this = this;

    bus.$on("val", name => {

      \_this.name = name;

    });

  },

在另一个需要传递参数的组件中,添加方法,方法中触发事件"val"

    postName: function() {

      bus.$emit("val", this.name);

    }

注意: 当只需要接收一次参数时,可以写$once

mounted: function() {

    let \_this = this;

    bus.$once("val", name => {

      \_this.name = name;

    });

 },

**路由的跳转:**

路由跳转有三种方式:

<**li** v-**for**="article in articles" @click="getDescribe(article.id)">

1. 直接带参数

methods:

getDescribe(id) {  
// 直接调用$router.push 实现携带参数的跳转  
 this.$router.push({  
 path: `/describe/${id}`,  
 })

router:

{  
 path: '/describe/:id',  
 name: 'Describe',  
 component: Describe  
 }

新页面获取参数:

this.$route.params.id

1. params带参数

methods:

**this**.$router.push({  
 name: 'Describe',  
 params: {  
 id: id  
 }  
 })

router:

{

path: '/describe/:id',

name: 'Describe',

component: Describe

}

新页面获取参数:

this.$route.params.id

1. query带参数

methods:

**this**.$router.push({  
 path: '/describe',  
 query: {  
 id: id  
 }  
 })

router:

{  
 path: '/describe',  
 name: 'Describe',  
 component: Describe  
 }

新页面获取参数:

this.$route.query.id

注意:1. 新页面获取参数都是通过this.route , 而不是this.router

1. 为了避免新页面刷新之后params参数消失,需要在router里加上"/:参数",并且,如果请求中不带有相关参数,就不显示页面.
2. query传参,写的是path,query,this.$route.query
3. params传参,写的是name,params,this.$route.params

**路由通过props传参:**

1.路由 router.js

{

path: 'calculation/',

name: 'calculation',

component: calculation,

**props: true,//**设置为true,则params的值将传到props

meta: {

title: 'calculation'

}

2.跳转

this.$router.push({

name: "calculation",

**params: {**

**goodsList: newList,**

**totalPrice: this.totalPrice**

**}**

});

3.新页面

**props: {**

**goodsList: Array,**

**totalPrice: Number**

**},**

**使用localstorage解决页面刷新参数消失的问题**

created () {

const tempData = localStorage.getItem("tempData")

if (tempData) {

this.goodsList = JSON.parse(tempData)

} else {

this.goodsList = this.$route.params.goodsList

}

this.goodsList.forEach(val => {

this.totalPrice += val[2] \* val[3]

})

localStorage.setItem('tempData', JSON.stringify(this.goodsList))

},

beforeDestroy () {

localStorage.removeItem('tempData')

},