Vue-router知识汇总

**一.下载并配置**

**1.下载vue-router插件:npm i vue-router -s**

**2.配置vue-router**

**(1)在router/index.js入口文件中创建VueRouter实例，并引入路由字典；**

import Vue from "vue";

import VueRouter from "vue-router";

import routes from "./routes";

Vue.use(VueRouter);

const router = new VueRouter({

  routes,

});

export default router;

**(2)在router/routes.js中创建路由字典，是一个数组，内部存储所有的路由对象；**

// routes作用是:创建路由字典，并保存所有的路由跳转信息。routes本质是一个数组，每一个路由是一个对象

import home from "../views/vuex练习/home";

import login from "../views/vueRouter练习/login";

import sonLogin from "../views/vueRouter练习/sonLogin";

export default [

  // 设置默认路由，如果路由的地址为/,那么路由会自动跳转至/home路由

  { path: "/", redirect: "/home" },

  { path: "/home", component: home, name: "home" },

  }

];

**(3)在main.js中引入VueRouter,并存储在new Vue中。**

import router from './router'

new Vue({

  router,

  store,

  render: h => h(App)

}).$mount('#app')

**二.new VueRouter({})中的属性介绍。**

const router = new VueRouter({

// 1.routes路由字典，存储所有的路由对象

  routes,

  // 2.mode配置路由url形式：①哈希#；②history

  mode: "history",

  // 3.base:在所有的页面路由前面会加/base

  base:'/base/',

  // ☆如下两个属性会为<router-link增加两个class属性>

  linkActiveClass:'active-link',

  linkExactActiveClass:'exact-active-link',

  // 4.滚动表现形式,to表示当前路由要跳转到哪一个路由；from表示从哪个路由跳转，savepPPosition表示是否记住前一次默认滚动的位置。

  scrollBehavior(to, from, savedPosition) {

    if (savedPosition) {

      return savedPosition;

    } else {

      return { x: 0, y: 0 };

    }

  },

  // 5.当某些浏览器不支持history的路径模式时，会自动切换成哈希#路由模式

  fallback: true

});

**三.routes[{路由对象1}，路由对象2]路由字典中的路由对象的属性介绍。**

export default [

  // 设置默认路由，如果路由的地址为/,那么路由会自动跳转至/home路由

  { path: "/", redirect: "/home" },

  {

    // 1.path代表路由跳转路径

    path: "/login",

    // 2.component:表示当前路由的组件对象

    component: login,

    // 3.表示当前路由组件的名称

    name: "login",

    // 4.添加一些自定义内容

    meta: {

      title: "this is login"

    },

    // 5.在当前组建中路由嵌套

    children: [

      {

        path: "/log/sonlogin",

        component: sonLogin

      }

    ]

  }

];

**四.$router和$route的区别**

**1.$router**

**(1)定义:**表示new VueRouter的路由实例对象，专门执行路由跳转的对象。

**(2)常用API：**

①$router.go(n)：接受一个整数作为参数，类似window.history.go(n)，在浏览器历史记录中前进或后退几步

②$router.push( location )：跳转导航的方法，这种方法会向history栈添加一个新的记录

③$router.replace( location )：和router.push()类似，但是它会替换掉当前的history记录，不会添加新的记录

④$router.back()：相当于router.go(-1)

⑤$router.forward()：相当于router.go(1)

**2.$route**

**(1)定义:**表示当前路由对象，封装着当前路由url中的信息，包括fullPath,path,meta,hash,name,query,params等信息。

**(2)常用的属性:**

①path:表示当前路由的相对路径，和routes路由字典中的path相同。

②query: 一个键值对对象，表示url查询字符串参数， 即？后面的内容，展现形式为{uname:gao,upwd:123}

③params: 一个键值对对象，表示路由参数，如：

<router-link to=“details/1”>,

{path:“/detail/:lid”};

此时lid就是params中的一个属性，即params：{lid:1}。

**五.路由跳转和路由传参**

**1. 路由跳转: 2种:**

**(1). 在html页面中把要跳转到的位置写死。**

<router-link to="/相对路径">文本</router-link>

注解：相对路径指的是在路由字典中该组件的path值，注意不要带参数。

自动翻译成  
 <a href="#/相对路径">文本</a>

Eg：<router-link to="/index">首页</index>

表示点击首页后，跳转到index页面组件。

**(2).在js代码中动态改变跳转路径**

this.$router.push("/相对路径")

例子：当点击登录后页面自动跳转到首页。

var login={

template:`<main>

用户名: <input v-model="uname"><br>

密码: <input type="password" v-model="upwd"><br>

<input type="button" value="登录" @click="login">

</main>`,

data(){

return {

uname:"dingding",

upwd:"123456"

}

},

methods:{

login(){

if(this.uname=="dingding"&&this.upwd=="123456"){

alert("登录成功！自动返回首页");

this.$router.push("/")

}else{

alert("用户名或密码错误！");

}

this.uname="";

this.upwd="";

}

**2.路由传参**

**(1)定义**：当页面中某一文本在实现路由跳转时，需要携带参数，这种情况就被称为路由传参。

**(2)路由传参经典应用**：以京东app为例，当进入众多商品详情页后(detail),点击某一个商品(lid)，然后进入该商品对应的详情页中，这一过程就是路由跳转过程中携带参数，实现跳转的案例。

※接收参数的页面指的就是detail页面组件。

<http://127.0.0.1:5500/vue_practice/2_SPA.html#/details/lid>

**(3)步骤：**

**①在当前页要实现路由跳转的位置的相对路径后加"/参数值" (从哪跳)**

强调“/”/后直接加参数值，: 1. 不要有? 2. 不要写参数名= 3. /直接跟参数值

Notes:假如参数值是一个变量，那么可以给to绑定属性:to=“ ”

<router-link :to="`/detail/${p1.href.split(`=`)[1]}`"></router-link>

注解:此处p1是一个data中的变量，通过切割字符串获取lid编号，实现查看固定编号的商品详情

data(){

  return{

    p1:{

      href:"detailshtml?lid=1"

    }

  }

}

**②在路由字典中，配置接收参数的页面(details)组件的参数;共两步：①在detail/后面加:参数值;②在最后面加prop:true (跳到哪)**

{path:"/details/:lid", component:details, props:true}

变量名

注解：props:true，表示让url中参数(lid)自动成为prop[]数组中的自定义属性

注解：

**③在要接收参数的目标组件(details)中，创建props:[]数组，并向其中添加与路由参数同名的自定义属性变量: prop:[“lid”]**

var details={  
 ...

props:["lid"], 这个lid就可自动接到url中的值，在details页面内，像普通变量一样使用

}

**例如：**

①【index.html中】

<ul>

<li><router-link to="/details/1">查看1号商品</router-link></li>

<li><router-link to="/details/2">查看2号商品</router-link></li>

<li><router-link to="/details/3">查看3号商品</router-link></li>

</ul>

②【router.js中】 {path:"/details/:lid",component:details,props:true},

③【detail.html中】

var details={

template:`<main>

<my-header></my-header>

<h2>这里是商品详情页</h2>

<h3>查询 {{lid}} 号商品的详细信息</h3>

</main>`,

props:["lid"]

}

**7.路由嵌套**

定义：如果想在一个一级路由(页面组件)中显示2级路由，则会用到路由的嵌套。例如在一个商家列表中，当点击点餐，评分，信息按钮时，就会跳转到对应的二级路由进行显示。

步骤:

(1).创建1级路由和2级路由，并在router中配置路由字典信息，创建及配置方式相当于页面组件的方式，唯一的不同就是2级路由要写在1级路由的children中。

import shop from "../views/vueRouter练习/shop/shop";

import shopGoods from "../views/vueRouter练习/shop/shopGoods/shopGoods";

import shopRating from "../views/vueRouter练习/shop/shopRating/shopRating";

import shopInfo from "../views/vueRouter练习/shop/shopInfo/shopInfo";

{

    path: "/shop",

    component: shop,

    children: [

      {

        path: "/shop/goods",

        component: shopGoods

      },

      {

        path: "/shop/info",

        component: shopInfo

      },

      {

        path: "/shop/rating",

        component: shopRating

      }

    ]

  }

(2).在1级路由shop.vue中，创建<router-view></router-view>，用来显示2级路由的内容