**user.vue中**

<template>

<div class='user-list'>

这是user组件

<div>

<router-link :to="'/user/'+**item.tip**+'/'+**item.id**" key='index' v-for="item,index in userList">{{item.userName}}</router-link>

</div>

<div class='userInfo' v-if='**userInfo**'>

<div v-if='userInfo.id' >

<p>年龄：{{userInfo.age}}</p>

<p>性別：{{userInfo.gender}}</p>

<p>愛好：{{userInfo.hobby}}</p>

</div>

</div>

</template>

<script>

let data=[

{

id: 1,

tip:'vip',

userName: '张三',

age: 38,

gender: '男',

hobby: '唱歌'

},

{

id: 2,

tip:'vip',

userName: '李四',

age: 39,

gender: '男',

hobby: '吃鸡'

},

{

id:3,

tip:'commom',

userName: '王麻子',

age:40,

gender: '男',

hobby: '打麻将'

}

]

export default {

data (){

return {

**userList:**data,

**userInfo**:{}

}

},

watch : {//watch监控$route

$route(){//url发生变化，$route会重新赋值

let id=this.$route.params.userId;

if(id){

this.userInfo=this.userList.filter((item)=>{

return item.id == id;

})[0]

}else{

this.userInfo = {};

}

}

},

created (){//user组件第一次创建时其生命周期里执行的钩子函数。

//复用user组件created不会再次执行，所以要用到watch。

//url一点发生变化$route,重新生成一个路由信息对象。

let id=this.$route.params.userId;

console.log(this.$route.params.userId);

if(id){

this.userInfo=**this.userList.filter((item)=>{**

**return item.id == id;**

**})[0]**

}else{

this.userInfo = {};

}

}

}

**在路由中：**

**..................**

**{**

**path: '/user/:tip?/:userId?',//动态路径的写法 问号表示匹配一次或者零次**

**当匹配到一个路由时，参数值会被设置到 this.$route.params**

**component:user**

**}**

**...................**

上面拿到data数据，在router-link中将数据循环出来，classs名字为userInfo的**div** 用来显示具体某个用户的所有数据：

1.在钩子函数created中，获得当前路由的路由信息对象里用户的id值

this.$route.params.userId;

2.创建一个userInfo对象来接收具体id对应的用户的数据

let id=this.$route.params.userId;

if(id){

this.userInfo=**this.userList.filter((item)=>{**

**return item.id == id;**

**})[0]**

}else{

this.userInfo = {};

}

过滤userList数组中的数据，filter会根据条件返回一个新数组

3.watch监控$route

组件生成一次，再去组件里操作，就是复用之前的组件，组件不会重新生成，组件生成一次就会执行一下这个生命周期里的**钩子函数。**

如上面：有三个用户共用一个user组件，当user组件生成一次，就会调用一次钩子函数created，当再去访问具体某个用户的时候，只会复用之前的user组件，不会重新生成，所以也不会重新执行生命周期里的钩子函数created

当切换不同用户的时候，虽然组件不会重新生成，但是url地址是在变化的，

所以这里使用watch来监控$route路由信息对象。

**优化代码user.vue中的代码**

**上面可以知道，**watch 和 created中的方法是一样的，可以将这个方法写到，公共函数里。

export default {

data (){

return {

**userList:**data,

**userInfo**:{}

}

},

watch : {

***$route***(){//监控路由信息对象里的userId，一变化就执行下面代码。

**this.getDate()**

}

},

created (){//user组件第一次创建时其生命周期里执行的钩子函数

**this.getDate()**

} ,

methods:{

**getDate()**{

let id=this.$route.params.userId;

if(id){

this.userInfo=this.userList.filter((item)=>{

return item.id == id;

})[0]

}else{

this.userInfo = {};

}

}

}

}