Vue项目步骤第03天

# 1、图片详情组件

1.1、缩略图静态界面的实现

在PhotoInfo.vue文件中的template中，增加如下标签

<div class="mui-content">

<ul class="mui-table-view mui-grid-view mui-grid-9">

<li class="mui-table-view-cell mui-media mui-col-xs-4 mui-col-sm-3">

<router-link to="/news/newslist">

<span class="mui-icon mui-icon-home"></span>

<div class="mui-media-body">新闻资讯</div>

</router-link>

</li>

</ul>

</div>

配置样式，去掉边框以及背景色

.mui-content,.mui-content ul{

background-color: #fff;

}

.mui-grid-view.mui-grid-9 .mui-table-view-cell{

border-right:0px;

border-bottom:0px;

}

.mui-grid-view.mui-grid-9{

border-top:0px;

border-left:0px;

}

去掉文字和图标，增加img标签，使用测试图片，并且将router-link去掉

<div class="mui-content">

<ul class="mui-table-view mui-grid-view mui-grid-9">

<li class="mui-table-view-cell mui-media mui-col-xs-4 mui-col-sm-3">

<img src="../../../static/test.jpg" alt="">

</li>

</ul>

</div>

配置好img标签的大小

.mui-content img{

width: 100px;

height: 100px;

}

1.2、在缩略图数据的获取

a、在data中定义好数据

data:function(){

return {

id:0, //图片的id

photoinfo :{} , //图片的详情描述数据对象

imageList:[]

}

},

b、在methods实现获取数据的方法

methods:{

getImages:function(){

var url = this.$urlPrefix + '/api/getthumimages/'+this.id;

this.$http.get(url).then(function(response){

this.imageList = response.body.message;

});

}

}

c、在created中调用b中定义的方法

created:function(){

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

this.getinfo();

this.getImages();

},

d、通过v-for的方法对缩略图进行标签填充

<div class="mui-content">

<ul class="mui-table-view mui-grid-view mui-grid-9">

<li v-for="item in imageList" class="mui-table-view-cell mui-media mui-col-xs-4 mui-col-sm-3">

<a to="#">

<img v-bind:src="item.src" alt="">

</a>

</li>

</ul>

</div>

e、注意，由于服务器返回的图片路径是相对路径，为了测试，可以手动更改成网络上的图片地址，代码如下：

getImages:function(){

var url = this.$urlPrefix + '/api/getthumimages/'+this.id;

this.$http.get(url).then(function(response){

var tempUrlList = [

"http://www.people.com.cn/mediafile/pic/20150917/23/10030234897474881435.jpg",

"http://jiangsu.china.com.cn/uploadfile/2016/0413/1460530872607118.jpg",

"http://www.people.com.cn/mediafile/pic/20150917/23/10030234897474881435.jpg",

"http://jiangsu.china.com.cn/uploadfile/2016/0413/1460530872607118.jpg",

"http://www.people.com.cn/mediafile/pic/20150917/23/10030234897474881435.jpg",

"http://jiangsu.china.com.cn/uploadfile/2016/0413/1460530872607118.jpg",

"http://www.people.com.cn/mediafile/pic/20150917/23/10030234897474881435.jpg",

"http://jiangsu.china.com.cn/uploadfile/2016/0413/1460530872607118.jpg"

];

this.imageList = response.body.message;

this.imageList.forEach(function(item,index){

item.src = tempUrlList[index];

});

});

}

1.3、缩略图放大效果的实现

1. 安装vue-preview

cnpm i vue-preview --save-dev

1. 在main.js中使用vue-preview

import VuePreview from 'vue-preview';

Vue.use(VuePreview);

1. 在PhotoInfo.vue文件中对img标签进行改造

<div class="mui-content">

<ul class="mui-table-view mui-grid-view mui-grid-9">

<li v-for="(item,index) in imageList" class="mui-table-view-cell mui-media mui-col-xs-4 mui-col-sm-3">

<img class="preview-img" :src="item.src" height="100px" @click="$preview.open(index, imageList)" >

</li>

</ul>

</div>

1. 配置svg文件类型的解析

{

test: /\.(png|jpg|gif|ttf|svg)$/,

loader:'url-loader?limit=20000'

}

1. 设置图片的宽度高度

this.imageList.forEach(function(item,index){

item.src = tempUrlList[index];

var img = document.createElement('img');

img.src = item.src;

item.h = img.height;

item.w = img.width;

});

# 2、商品列表的实现

* 1. 商品列表静态界面的实现

1. **标题静态效果**

使用mui中的图文表格，复制代码到GoodsList.vue文件的template中

<template>

<div class="templateDiv">

<div id="mui-content" class="mui-content" style="background-color:#fff">

<ul class="mui-table-view mui-grid-view">

<li class="mui-table-view-cell mui-media mui-col-xs-6">

<a href="#">

<img class="mui-media-object" src="../../../static/test.jpg">

<div class="mui-media-body">标题</div>

<div class="desc">

<p>

<span>￥价格1</span>

<s>￥价格2</s>

</p>

<p>

<h6 class="left">热卖中</h6>

<h6 class="right">剩余2件</h6>

</p>

</div>

</a>

</li>

</ul>

</div>

</div>

</template>

1. **样式设定**

<style scoped>

#mui-content li{

border: 1px solid rgba(0,0,0,0.4);

-webkit-box-shadow: 0 0 4px #000;

-moz-box-shadow: 0 0 4px #000 ;

box-shadow: 0 0 4px #000 ;

margin-left: 4px;

padding: 5px;

}

#mui-content .mui-media-body{

color: #0094ff;

}

#mui-content .desc{

height: 60px;

background-color: rgba(0,0,0,0.1);

margin-top: 10px;

padding: 5px;

text-align: left;

}

#mui-content .desc span{

color: red;

margin-right: 15px;

}

#mui-content a{

padding-left: 0px;

margin-left: 0px;

}

#mui-content .desc .right{

position: absolute;

right: 10px;

bottom: 0px;

}

#mui-content .desc .left{

position: absolute;

left: 10px;

bottom: 0px;

}

</style>

* 1. 商品列表界面数据的获取

**a、商品列表数据获取**

<script>

export default {

data:function(){

return {

goodsList:[]

}

},

methods:{

getGoodsList(){

var url = this.$urlPrefix + '/api/getgoods?pageindex=1';

this.$http.get(url).then(function(res){

this.goodsList = res.body.message;

});

}

},

created:function(){

this.getGoodsList();

}

}

</script>

**b、使用v-for动态填充li的数量**

<li v-for="item in goodsList" class="mui-table-view-cell mui-media mui-col-xs-6">

<a href="#">

<img class="mui-media-object" :src="item.img\_url">

<div class="mui-media-body">{{item.title}}</div>

<div class="desc">

<p>

<span>￥{{item.sell\_price}}</span>

<s>￥{{item.market\_price}}</s>

</p>

<p>

<h6 class="left">热卖中</h6>

<h6 class="right">剩余{{item.stock\_quantity}}件</h6>

</p>

</div>

</a>

</li>

# 商品详情的实现

3.1、商品详情入口的实现

1. 创建GoodsInfo.vue文件，建立好基本的代码结构

<template>

<div class="templateDiv">

商品详情

</div>

</template>

<script>

export default {

data:function(){

return {

}

},

methods:{

},

created:function(){

}

}

</script>

<style scoped>

</style>

1. 在GoodsList中，将a标签替换成router-link标签，href变为to，to的值为/goods/goodsinfo
2. 定义路由规则
3. 传递参数

指定哪些路径需要参数

{

path:"/goods/goodsinfo/:goodsid",

component:GoodsInfo

}

在router-link中传递参数，注意to的写法

<router-link v-bind="{to:'/goods/goodsinfo/'+item.id}">

在目标组件中接收参数

<script>

export default {

data:function(){

return {

goodsId:0

}

},

methods:{

},

created:function(){

this.goodsId = this.$route.params.goodsid;

}

}

</script>

3.2、商品轮播图的实现

a、创建slide.vue这个文件，这个组件专门来实现轮播图的功能，这个是个子组件，在父组件中使用某一个标签就可以显示slide.vue这个组件所表示的内容

<template>

<div>

图片轮播

</div>

</template>

<script>

export default {

data:function(){

return {

}

},

methods:{

},

created:function(){

}

}

</script>

<style scoped>

</style>

b、在GoodsInfo.vue文件中进行使用

<template>

<div class="templateDiv">

<div class="silder">

<silder></silder>

</div>

</div>

</template>

<script>

import slider from "../../components/subcomponent/Slide.vue"

export default {

components:{silder:slider},

data:function(){

return {

goodsId:0

}

},

methods:{

},

created:function(){

this.goodsId = this.$route.params.goodsid;

}

}

</script>

1. 传值给Slide.vue组件

在Slide.vue定义有哪些值可以从父组件传递过来

props:["imgs"],

在GoodsInfo.vue中，使用slide标签的时候，将值进行传递

<silder :imgs="imgs"></silder>

同时，在GoodsInfo.vue文件的data中增加属性imgs

data:function(){

return {

goodsId:0,

imgs:[]

}

}

1. 在GoodsInfo.vue文件中获取数据，对imgs进行赋值

methods:{

getimgs:function(){

var url = this.$urlPrefix + '/api/getthumimages/'+this.goodsId;

this.$http.get(url).then(function(res){

this.imgs =res.body.message;

//由于服务器返回的是相对路径，为了方便，将imgs中的内容替换成网络图片

var tempUrlList = [ "http://www.people.com.cn/mediafile/pic/20150917/23/10030234897474881435.jpg", "http://jiangsu.china.com.cn/uploadfile/2016/0413/1460530872607118.jpg"

];

this.imgs = tempUrlList;

});

}

},

created:function(){

this.goodsId = this.$route.params.goodsid;

this.getimgs();

e、实现Slide.vue的效果

<template>

<div>

<mt-swipe :auto="2000">

<mt-swipe-item v-for="item in imgs">

<img :src="item">

</mt-swipe-item>

</mt-swipe>

</div>

</template>

<script>

export default {

props:["imgs"],

data:function(){

return {

}

},

methods:{

},

created:function(){

}

}

</script>

<style scoped>

.mint-swipe{

height: 200px;

}

.mint-swipe-item img{

width: 100%;

}

.mint-swipe-item{

width: 100%;

height: 300px;

}

</style>

f、将home的轮播图也替换成Slide.vue组件

3.3、商品详情界面的静态效果

a、静态标签

<div class="templateDiv">

<div class="silder">

<silder :imgs="imgs"></silder>

</div>

<div id="buy">

<h4>标题</h4>

<p class="line"></p>

<ul>

<li class="price">

市场价: <s>￥</s>

销售价：<span>￥</span>

</li>

<li class="inputli">

购买数量： 3

</li>

<li>

<mt-button type="primary" size="small">立即购买</mt-button>

<mt-button type="danger" size="small">加入购物车</mt-button>

</li>

</ul>

</div>

<div id="params">

<h6>商品参数</h6>

<p class="line"></p>

<ul>

<li>商品货号：</li>

<li>库存情况：</li>

<li>上架时间：</li>

</ul>

</div>

<div id="other">

<mt-button class="imgdesc" type="primary" size="large">图文详情</mt-button>

<mt-button type="danger" size="large">商品评论</mt-button>

</div>

</div>

b、样式设置

<style scoped>

.silder{

border:1px solid rgba(0,0,0,0.4);

border-radius: 5px;

margin: 5px;

}

#buy,#params,#other{

margin: 5px;

padding: 5px;

border:1px solid rgba(0,0,0,0.4);

border-radius: 5px;

}

.line{

height: 1px;

border: 1px solid rgba(0,0,0,0.2);

}

#buy ul,#params ul{

padding-left: 0px;

}

#buy h4{

color:#0094ff;

padding: 5px;

}

#buy li,#params li{

list-style: none;

padding: 8px;

}

#buy .price span{

color:red;

}

#other .imgdesc{

margin-bottom: 20px;

}

.inputli{

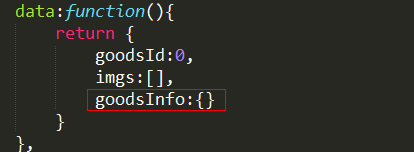
position: relative;

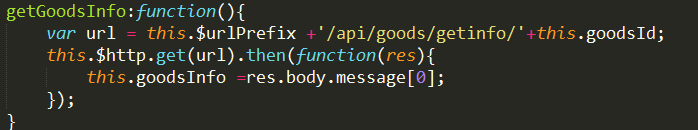
}

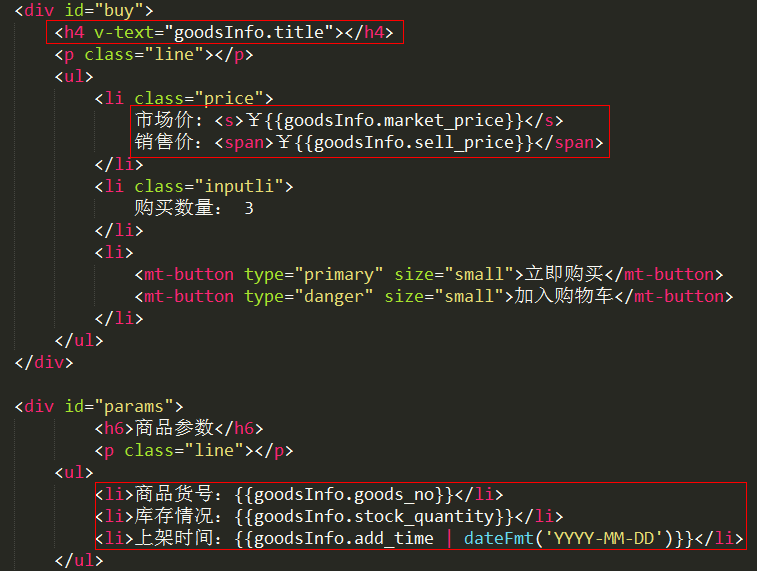
</style>

3.4、商品详情数据的获取

1. 获取数据
2. 赋值
3. 在标签中使用数据







3.5、商品详情图文详情的实现

a、创建GoodsDesc.vue

<template>

<div class="templateDiv">

图文详情

</div>

</template>

<script>

export default {

data:function(){

return {

}

},

methods:{

},

created:function(){

}

}

</script>

<style scoped>

</style>

b、增加入口

<router-link v-bind="{to:'/goods/goodsdesc/'+goodsId}">

<mt-button class="imgdesc" type="primary" size="large">图文详情</mt-button>

</router-link>

c、配置路由

import GoodsDesc from "./components/goods/GoodsDesc.vue";

{

path:'/goods/goodsdesc/:id',

component:GoodsDesc

},

d、获取参数

export default {

data:function(){

return {

goodsId:0

}

},

methods:{

},

created:function(){

this.goodsId = this.$route.params.id;

}

}

e、获取数据

export default {

data:function(){

return {

goodsId:0,

info:{}

}

},

methods:{

getinfo:function(){

var url =this.$urlPrefix +'/api/goods/getdesc/'+this.goodsId;

this.$http.get(url).then(function(res){

this.info = res.body.message[0];

});

}

},

created:function(){

this.goodsId = this.$route.params.id;

this.getinfo();

}

}

f、界面和数据绑定

<template>

<div class="templateDiv">

<h4 v-text="info.title"></h4>

<p class="line"></p>

<p v-html="info.content"></p>

</div>

</template>

3.6、商品详情评论功能实现

用GoodsComment.vue进行包装,原因是因为Comment.vue这个组件中定义了props[],这里面的值必须由父组件使用标签的时候进行传值，因此使用GoodsComment.vue进行一次包装

a、创建GoodComment.vue文件，使用Comment.vue组件，

b、增加入口router-link

c、配置路由规则

d、接收参数

<router-link v-bind="{to:'/goods/goodscomment/'+goodsId}">

<mt-button type="danger" size="large">商品评论</mt-button>

</router-link>

<template>

<div class="templateDiv">

<comment v-bind:mynewsid="id"></comment>

</div>

</template>

<script>

import Comment from "../subcomponent/Comment.vue";

export default {

data:function(){

return {

id:0

}

},

components:{

comment:Comment//前面是我将要使用的标签的名称，后面是组件的对象

},

methods:{

},

created:function(){

var goodsId = this.$route.params.id;

this.id = goodsId;

}

}

</script>

import GoodsComment from "./components/goods/GoodsComment.vue";

{

path:'/goods/goodscomment/:id',

component:GoodsComment

}

3.7、商品详情数据购买数量功能的实现

将数量增减的操作放到InputNumber.vue组件中，改变数量会影响到的是用户购买商品或者添加商品到购物车时候的一些操作，而购买按钮和加入购物车按钮是在GoodsInfo.vue中，所以这里涉及到一个子组件传值给父组件的操作

a、在subcomponent文件夹中建立InputNumber.vue组件，并填充对应代码结构

<template>

<div>

输入数量

</div>

</template>

<script>

export default {

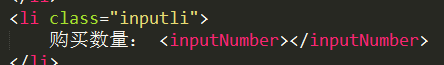
}

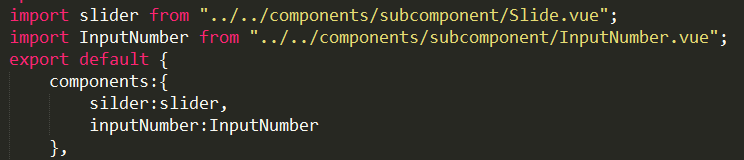
</script>

<style scoped>

</style>

b、在GoodsInfo.vue文件中使用InputNumber.vue这个组件





c、配置InputNumber.vue的静态界面效果

<template>

<div class="container">

<div class="inleft" v-on:click="substrict">-</div>

<div class="incenter" v-text="count">1</div>

<div class="inright" v-on:click="add">+</div>

</div>

</template>

<script>

export default {

data:function(){

return {

count:1

}

},

methods:{

substrict:function(){

this.count = this.count-1;

if(this.count <1){

this.count = 1;

}

},

add:function(){

this.count = this.count + 1;

}

}

}

</script>

<style scoped>

.container{

display: flex;

}

.inleft,.inright{

border: 1px solid #ccc;

width: 20px;

height: 20px;

margin-left: 10px;

text-align:center;

line-height: 20px;

}

.incenter{

width: 20px;

height: 20px;

text-align:center;

line-height: 20px;

margin-left: 10px;

}

</style>

d、将InputNumber中的值传给GoodInfo.vue，因为这个值会在用户点击立即购买和加入购物车按钮的时候使用到

在add方法和substrict中增加一个方法sendMessage，这个方法中将count的值传给父组件

methods:{

substrict:function(){

this.count = this.count-1;

if(this.count <1){

this.count = 1;

}

this.sendmessage();

},

add:function(){

this.count = this.count + 1;

this.sendmessage();

},

sendmessage:function(){

this.$emit('dataobj',this.count);

}

}

this.$emit(“dataobj”,this.count)的意思是会发出一个事件，这个事件的名字叫做dataobj，谁监听了这个事件谁就能响应这个事件。那谁能监听这个事件呢？当然是使用到了InputNumber这个组件的父组件了，因为我们往往是在一个标签中注册事件嘛

<li class="inputli">

购买数量： <inputNumber v-on:dataobj="getMessage"></inputNumber>

</li>

getMessage:function(count){

console.log(count);

}

e、最后修改GoodsInfo.vue组件中的标签以及样式

<li class="inputli">

<div class="count">

<span>购买数量：</span>

<inputNumber v-on:dataobj="getMessage"></inputNumber>

</div>

</li>

.count{

display: flex;

}