

Vue第三天

今日内容介绍

- 图片分享区域功能
- 图片详情功能
- 图片预览组件的使用
- 商品列表
- 商品详情
- 子组件的提取和复用

今日内容学习目标

- 能够完成图片列表页面静态结构设计
- 能够利用AJAX实现图片列表数据动态化功能
- 能够利用MUI的9宫格样式实现图片详情页缩略图排布设计
- 能够实现图片详情页数据动态化
- 能够参照vue-preview文档实现图片预览功能
- 能够实现商品列表页面静态结构设计以及数据动态化
- 将图片轮播功能提取到子组件中复用
- 能够开发子组件实现购买数量的控制

详细内容

1.0 图片列表页面



分类数据从api:
http://webhm.top:8899/api/getimgcategory
中获取

图片使用mint-ui中的
懒加载组件:Lazy-load
实现

lazy-load使用代码如下:

使用了v-lazy指定图片的地址后, mint-ui就会自动实现延迟加载

图片数据从api:

http://webhm.top:8899/api/getimages/图片分类id
中获取

2.0 图片详情页面



点击图片发出
/photo/imageshow/3
的请求将3传入给
imgshow.vue组件



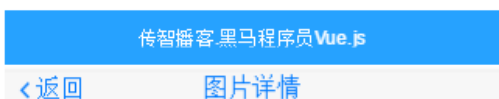
1、在imgshow.vue中通过 this.\$route.params.imgid
之所以能够点出imgid是因为在main.js中配置好了路由规则:
{path: '/photo/imageshow/:imgid', component: imgshow},

缩略图从api:
http://webhm.top:8899/api/getthumbimages/传入当前
图文数据id
中获取

从http://webhm.top:8899/api/get
imageInfo/图文数据id
中获取

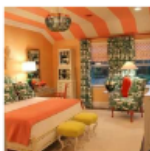
3.0 图片详情页面中图片预览

- 图片预览功能使用vue-preview 组件实现
- vue-preview使用帮助地址: <https://github.com/LS1231/vue-preview>
- 在图文详情页页面中的使用

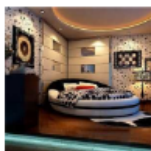


现代简约 明亮的外表卧室卧室背景墙、吊顶黄色

2015-04-18 0 次浏览 分类: 经济民生



点击图片后
预览效果



不要简朴不要素雅洋气卧室我做主，高低床榻榻米式靓丽卧室装扮，现代油画壁画帆布画卧室餐厅仟象映画，现代中式卧室装修效果图欣赏，温馨浪漫，而且很时尚的卧室设计，欧式卧室飘窗装修效果图。



首页



会员



购物车



搜索



每个图片是通过：

```
 实现的，其中两个加粗的地方是
实现效果的代码 index:表示图片的索引，thums: 表示存放图片数据的数组
thums各位严格要求为: [{
  src: 'https://placekitten.com/600/400',
  w: 600,
  h: 400
},
  {}, {}, ...]
```

vue-preview使用步骤总结：

1、安装 vue-preview ---> npm install vue-preview --save

2、在webpack.config.js的loaders中增加如下配置：

```
{
  test: /\.vue-preview.src.*?js$/,
  loader: 'babel'
}
```

3、在main.js中增加如下代码

```
import VuePreview from 'vue-preview'
Vue.use(VuePreview)
```

4、使用

```
<template>
  
</template>

<script>
export default {
  data () {
    return {
      list: [{
        src: 'https://placekitten.com/600/400',
        w: 600,
        h: 400
      }, {
        src: 'https://placekitten.com/1200/900',
        w: 1200,
        h: 900
      }
    ]
  }
}
```

```
        w: 1200,  
        h: 900  
      }]  
    }  
  }  
}</script>
```

4.0 图片详情页面中集成评论组件

- 关键代码如下

```
<!-- 4.0 评论组件 -->  
<comment :artid="imgid"></comment>
```

```
import comment from '../subcom/comment.vue'  
export default {  
  components: {comment},
```

5.0 商品列表页面

- 商品列表使用MUI的图文表格来实现: <http://www.dcloud.io/hellomui/examples/slider-table-default.html>



使用MUI的图文表格实现页面布局：

MUI图文表格案例地址：

<http://www.dcloud.io/hellomui/examples/slider-table-default.html>

数据来源于api：

<http://webhm.top:8899/api/getgoods?pageindex=1>

5.0 商品详情页面

商品详情页面分为如下区域：

- 1、轮播图

商品轮播图与首页轮播图的业务其实是一样的，因此这里可以抽出一个子组件来实现轮播功能的复用步骤：

- 1、在项目的 subcomm文件夹中创建一个名称为： silder.vue 的组件
- 2、在silder.vue中实现代码和在图文详情页面中使用：

silder.vue 组件中编写的代码:

```
<template>
  <div class="tpl">
    <mt-swipe :auto="2000">
      <mt-swipe-item v-for="item in swipeList">
        <a :href="item.url">
          
        </a>
      </mt-swipe-item>
    </mt-swipe>
  </div>
</template>
<script>
  export default{
    data(){
      return {
        //用于存放轮播图地址的数组，从父组件中传入
        props: ['swipeList']
      }
    }
  }
</script>
```

必须保持同名

调用的代码:

```
<!-- 1.0 轮播使用的mint-ui中的swipe组件 -->
<silder :swipeList="swipeList"></silder>
```

```
export default{
  data(){
    return {
      swipeList:[]
    }
  },
  components:{silder},
  methods:{
    // 1.0 获取轮播图数据
    getimglist(){
      this.$http.get(common.apiDomain+ '/api/getthumimages/商品id')
        .then(res=>{
          // 将数据赋值给swipeList属性，自动导致v-for重新执行
          this.swipeList = res.body.message;
        });
    }
  }
},
```

这个变量的值来自于
data中定义的数组变量

通过AJAX请求到数据后赋值给swipeList

- 2、购买区

- 购买数量子组件inputNumber.vue的实现

此页面使用inputNumber.vue的代码：

```
<template>
<div class="tpl">
  <!-- 1.0 轮播 -->
  <silder class="silder" :swipeList="swipeList"></silder>

  <!-- 2.0 购买操作区域 -->
  <div class="opt">
    <h4>{{info.title}}</h4>
    <p>市场价: <s>¥{{info.market_price}}</s>
    销售价: <span class="sellprice">¥{{info.sell_price}}</span></p>

    <div class="num">
      <div class="numleft">购买数量:</div>
      <inputnumber class="inputnumber" v-on:sendcount="getcount"></inputnumber>
    </div>
  </div>
</div>
</template>
```

结构

```
<template>
<div class="tpl">
  <div class="left" @click="substrict"></div>
  <div class="center">{{count}}</div>
  <div class="right" @click="add"></div>
</div>
</template>
<script>
export default{
  data(){
    return {
      count:1
    },
  },
  methods:{
    // 1.0 自减1
    substrict(){
      if(this.count>1){
        this.count--;
      }else{
        this.count = 1;
      }
      this.emit();
    },
    // 2.0 自增1
    add(){
      this.count ++;
      this.emit();
    },
    emit(){
      this.$emit('sendcount',this.count);
    }
  }
}
</script>
```

二者必须保持一致

```
// 2.0 获取inputNumber组件发送过来的count值
getcount(count){
  this.count = count;
},
```

将当前选择的数量传递给注册了 sendcount 事件的组件中

- 3、图文详情

- 图文详情的数据来源于: <http://webhm.top:8899/api/goods/getdesc/>图文id

- 4、商品评论

- 跳转到 subcomm/comment.vue子组件即可