高仿”饿了吗”外卖App

# 准备

## 项目描述

* 此项目为外卖App核心的**商家模块**的**SPA**
* 包括商品、评论、商家介绍、购物车等多个子模块
* 使用**Vue全家桶+ES6+Webpack等**前端最新最热的技术
* 采用**模块化**、**组件化**、**工程化**的模式开发

## 项目功能界面

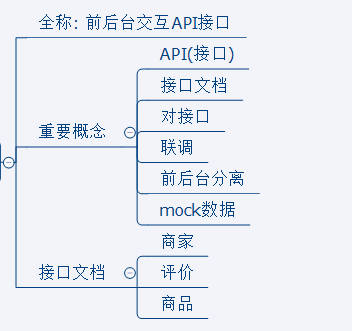
## 技术选型



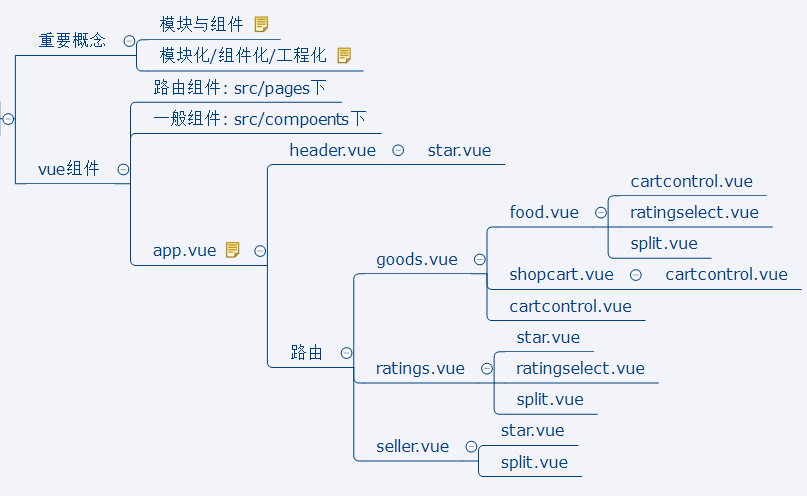
## 前端路由



## API接口



## vue组件



01. app.vue

\* 应用的根组件

02. header.vue

\* 页面头部显示商家基本信息的组件

03. goods.vue

\* 商品分路由组件: 商品列表/购物车相关功能

04. ratings.vue

\* 评价分路由组件: 商品评价列表

05. seller.vue

\* 商家详情分路由组件: 显示商家详情

06. star.vue

\* 星级评价组件

07. food.vue

\* 商品列表组件: 显示商品分类/商品列表, 购物项加减

08. shopcart.vue

\* 底部购物车组件: 显示购物项列表及相关操作

09. cartcontrol.vue

\* 购物项操作组件: 增加/减少某个购物项的数量

10. ratingselect.vue

\* 商品评论列表组件: 过滤查看商品评价列表

11. split.vue

\* 分隔线小组件

## 你能从此项目中学到什么?

* 流程及开发方法
  + 熟悉一个项目的开发流程
  + 学会**组件化、模块化、工程化**的开发模式
  + 掌握使用**vue-cli脚手架**初始化Vue.js项目
  + 学会**模拟json后端数据**，实现**前后端分离**开发
  + 学会**ES6+eslint**的开发方式
  + 掌握一些**项目优化**技巧
* Vue插件或第三方库
  + 学会使用**vue-router**开发单页应用
  + 学会使用**axios/vue-resource**与后端进行数据交互
  + 学会使用**vuex**管理应用组件状态
  + 学会使用**better-scroll/vue-scroller**实现页面滑动效果
  + 学会使用**mint-ui**组件库构建界面
  + 学会使用**vue-lazy**实现图片惰加载
  + 学会使用**mockjs**模拟后台数据接口
* 样式/布局/效果相关
  + 学会使用**stylus**编写模块化的CSS
  + 学会使用Vue.js的**过渡**编写酷炫的交互动画
  + 学会制作并使用**图标字体**
  + 学会解决移动端**1px边框**问题
  + 学会移动端经典的**css sticky footer**布局
  + 学会**flex**弹性布局

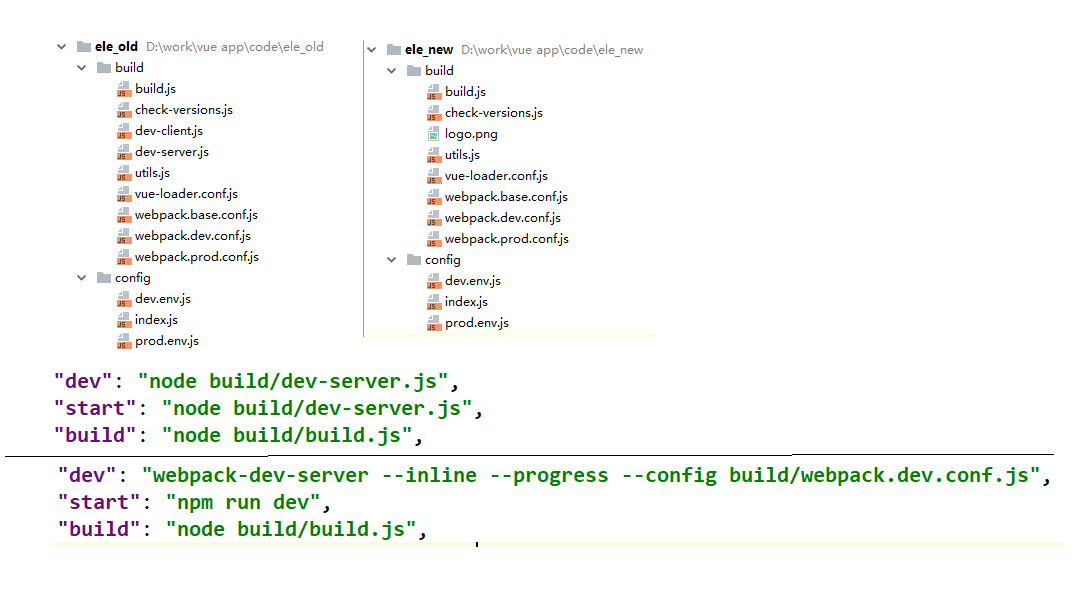
# 应用开发详解

## 开启项目开发

### 使用vue-cli(脚手架)搭建项目

* Vue-cli是vue官方提供的用于搭建基于vue+webpack+es6项目的脚手架工具
* 在线文档: <https://github.com/vuejs/vue-cli>
* 操作:
  + npm install -g vue-cli : 全局下载工具
  + vue init webpack ele :下载基于webpack模板项目
  + cd ele
  + npm install : 下载项目依赖的所有模块
  + npm run dev
  + 访问: localhost:8080

### 新老版本的脚手架项目对比



### 项目结构分析

eleApp

|-- build : webpack相关的配置文件夹(基本不需要修改)

|-- dev-server.js : 通过express启动后台服务器

|-- config: webpack相关的配置文件夹(基本不需要修改)

|-- index.js: 指定的后台服务的端口号和静态资源文件夹

|-- node\_modules

|-- src : 源码文件夹

|-- components: vue组件及其相关资源文件夹

|-- app.vue: 应用根主组件

|-- main.js: 应用入口js

|-- static: 静态资源文件夹

|-- .babelrc: babel的配置文件

|-- .editorconfig: 通过编辑器的编码/格式进行一定的配置

|-- .eslintignore: eslint检查忽略的配置

|-- .eslintrc.js: eslint检查的配置

|-- .gitignore: git版本管制忽略的配置

|-- index.html: 主页面文件

|-- package.json: 应用包配置文件

|-- README.md: 应用描述说明的readme文件

### 编码测试与打包发布项目

* 编码测试
  + npm run dev
  + 访问: <http://localhost:8080>
  + 编码, 自动编译打包(HMR), 查看效果
* 打包发布
  + npm run build
  + npm install -g serve
  + serve dist
  + 访问: [http://localhost:5000](http://localhost:9000)

## 功能需求分析

* 演示项目功能, 对功能模块进行分析说明

## 资源准备

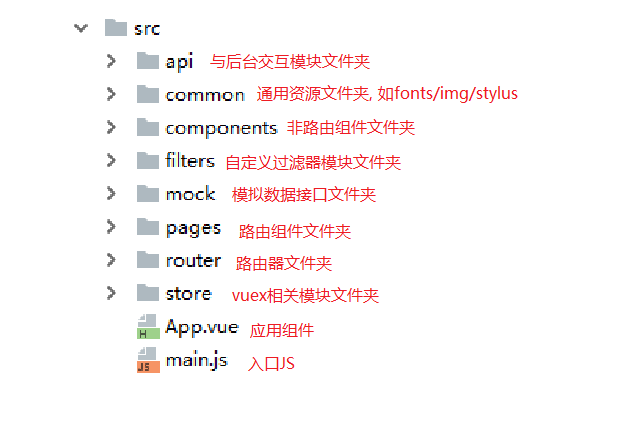
### 相关概念

* 标注图(设计稿): 对应用界面各个组成元素进行坐标/大小/颜色等进行标签的界面图
* 切图: 将应用界面的一些静态图形部分, 通过工具(如photoshop)剪裁生成的图片
* 图片Base64: 样式中引用的小图片, 在webpack打包会自动处理转换为样式内部的Base64编码字符串
* 2x与3x图: 不同手机的屏幕密度不一样, 一般都在2以上(如iphone6为2,iphone6s为3),  为了适配不同的手机, UI设计师为同一个图片制作了2x和3x的2套图片(图形一样, 但大小不一样)

### 图标字体制作

* 图标字体: 使用IcoMoon将SVG格式的图标转换生成图标字体及样式
* 进入icoMoon官网: <https://icomoon.io/>
* 点击右上角, 进入处理页面
* 点击左上角,　选择resource\SVG\\*.svg, 上传显示到页面
* 在页面选择所有svg, 点击右下角生成图标字体样式
* 点击左上角指定Font-Name为sell-icon,　点击右下角下载到本地
* 解压zip包, 访问demo.html测试
* 我们项目需要的是fonts和style.css

## 项目源码目录设计



## Stylus

### 当前主流的三种css预编译器([对比](http://www.w3cplus.com/css/sass-vs-less-vs-stylus-a-preprocessor-shootout.html))

* Less
* Sass
* Stylus

### 安装stylus依赖包

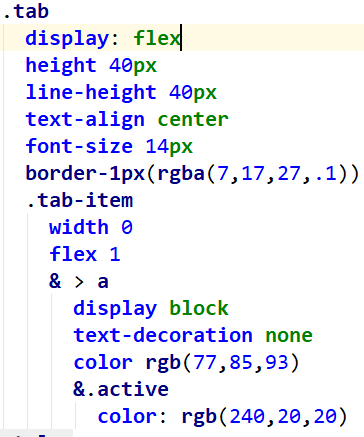
|  |
| --- |
| **npm install stylus stylus-loader --save-dev** |

### 编写样式

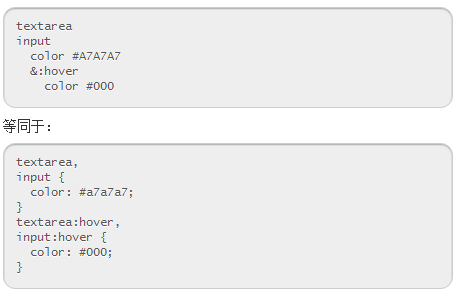
|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

### Stylus的基本语法

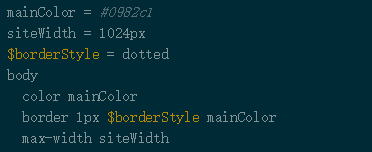
* + [stylus中文版参考文档](http://www.zhangxinxu.com/jq/stylus/)
  + **结构化:** 
    - 完全通过缩进控制, 不需要大括号和分号,　冒号是可选的



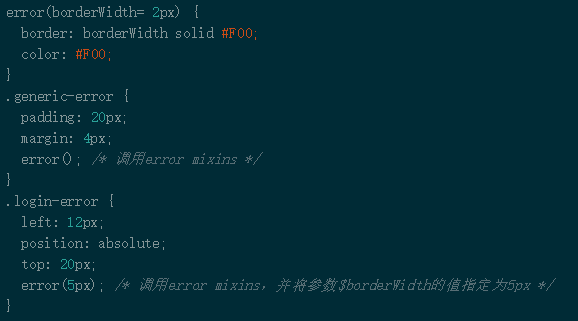
* + **父级引用:** 
    - 使用字符&指向父选择器



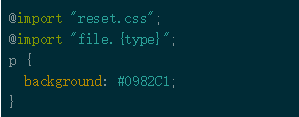
* + 变量:
    - 定义变量: name=value 如: mainColor = #0982c1
    - 引用变量: name 如: color mainColor
    - name可以没有前缀要求, 但最好以$开头(好识别)



* + **混合(Mixins)**
    - 也就是预处器中的函数
    - 函数参数可以指定默认值
    - 某段CSS样式要用到多个元素上, 只有其中的1,2个样式值有变化

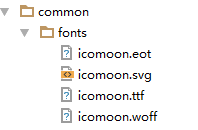


* + **导入(import)**
    - 通过@import引入其它样式文件



### 将font字体及样式加入

* 将icomoon\fonts\下所有文件拷贝到src\common\fonts目录下



* 将iconmoon\style.css拷贝到src\common\stylus\下, 重命名为fonts.styl
* 去除文件中的大括号和分号, 并修正url路径

|  |
| --- |
| **@font-face** {  **font-family**: **'icomoon'**;  **src**: **url**(**'../fonts/icomoon.eot?s2camu'**);  **src**: **url**(**'../fonts/icomoon.eot?s2camu#iefix'**) **format**(**'embedded-opentype'**),  **url**(**'../fonts/icomoon.ttf?s2camu'**) **format**(**'truetype'**),  **url**(**'../fonts/icomoon.woff?s2camu'**) **format**(**'woff'**),  **url**(**'../fonts/icomoon.svg?s2camu#icomoon'**) **format**(**'svg'**);  **font-weight**: **normal**;  **font-style**: **normal**; }  [**class**^=**"icon-"**], [**class**\*=**" icon-"**] {  */\* use !important to prevent issues with browser extensions that change fonts \*/* **font-family**: **'icomoon' !important**;  **speak**: **none**;  **font-style**: **normal**;  **font-weight**: **normal**;  **font-variant**: **normal**;  **text-transform**: **none**;  **line-height**: 1;   */\* Better Font Rendering =========== \*/* **-webkit-font-smoothing**: **antialiased**;  **-moz-osx-font-smoothing**: **grayscale**; }  .**icon-add\_circle**:**before** {  **content**: **"\e900"**; } .**icon-arrow\_lift**:**before** {  **content**: **"\e901"**; } .**icon-check\_circle**:**before** {  **content**: **"\e902"**; } .**icon-close**:**before** {  **content**: **"\e903"**; } .**icon-favorite**:**before** {  **content**: **"\e904"**; } .**icon-keyboard\_arrow\_right**:**before** {  **content**: **"\e905"**; } .**icon-remove\_circle\_outline**:**before** {  **content**: **"\e906"**; } .**icon-shopping\_cart**:**before** {  **content**: **"\e907"**; } .**icon-thumb\_down**:**before** {  **content**: **"\e908"**; } .**icon-thumb\_up**:**before** {  **content**: **"\e909"**; } |

### 其它stylus样式

* mixins.styl

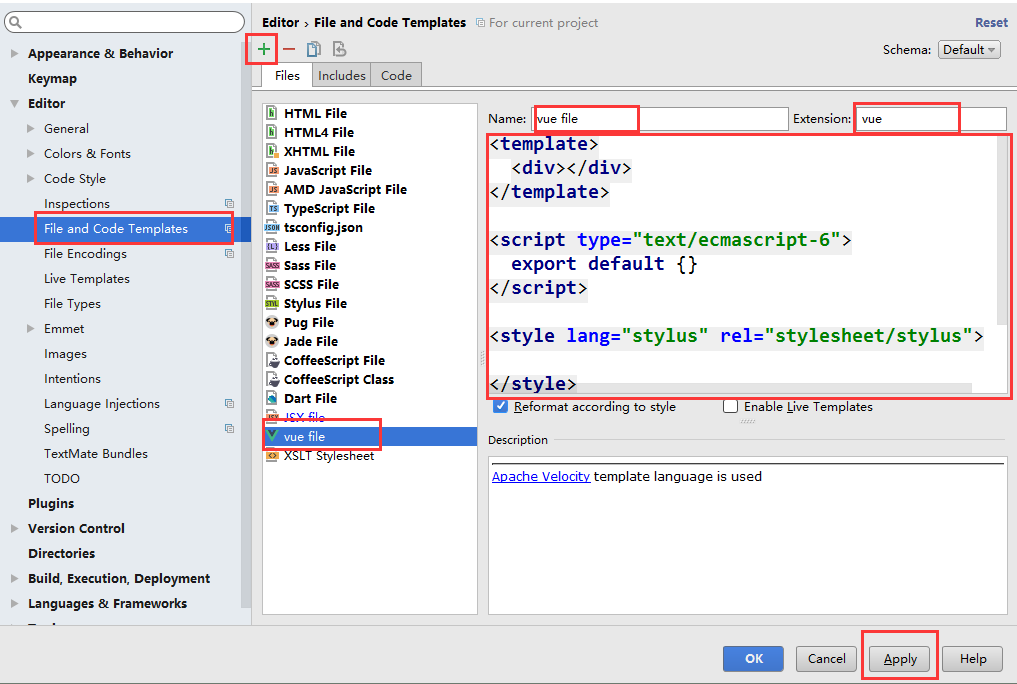
|  |
| --- |
| *// 混合：把混合底下的样式集原封不动的搬到调用处* **bg-image**(**$url**)  **background-image url**(**$url** +**"@2x.png"**)  **@media** (**min-device-pixel-ratio**:3),(**-webkit-min-device-pixel-ratio**:3)  **background-image url**( **$url** +**"@3x.png"**)   **bg-star**(**$url**)  **background-image url**(**"../star/"**+**$url** +**"@2x.png"**)  **@media** (**min-device-pixel-ratio**:3),(**-webkit-min-device-pixel-ratio**:3)  **background-image url**(**"../star/"**+**$url** +**"@3x.png"**)   **border-1px**(**$color**)  **position relative  border none** &:**after  position absolute  bottom** 0  **left** 0  **content ''  width** 100%  **height** 1**px  transform scaleY**(0.5)  **background $color** |

## ESLint

* + ESLint是一个代码规范检查工具
  + 基本已替代以前的JSHint
  + ESLint提供以下支持
    - ES6
    - AngularJS
    - JSX
    - Style检查
    - 自定义错误和提示
  + ESLint提供以下几种校验
    - 语法错误校验
    - 不重要或丢失的标点符号，如分号
    - 没法运行到的代码块（使用过WebStorm的童鞋应该了解）
    - 未被使用的参数提醒
    - 漏掉的结束符，如}
    - 确保样式的统一规则，如sass或者less
    - 检查变量的命名
  + 规则的错误等级有三种
    - 0：关闭规则。
    - 1：打开规则，并且作为一个警告（不影响exit code）。
    - 2：打开规则，并且作为一个错误（exit code将会是1）。
  + 相关配置文件
    - .eslintrc.js : 规则相关配置文件, 可以在此修改规则
    - .eslintignore: 指令检查忽略的文件,　可以在此添加想忽略的文件

## Vue组件化

### 配置vue组件文件模板



|  |
| --- |
| <**template**>  <**div**>   </**div**> </**template**> <**script**>  **export default** {   } </**script**> <**style lang=”stylus” rel=”stylesheet/stylus”**>  </**style**> |

### 分析应用的整体vue组件结构

src

|-- components------------非路由组件文件夹

|-- header---------------头部组件文件夹

|-- header.vue--------前部组件vue

|-- xxx.png-----------头部相关的n个图标

|-- pages-----------------路由组件文件夹

|-- seller---------------商家详情组件文件夹

|-- seller.vue--------商家详情组件vue

|-- goods----------------商品组件文件夹

|-- goods.vue---------商品组件vue

|-- ratings--------------评价组件文件夹

|-- ratings.vue-------评价组件vue

|-- app.vue---------------应用根组件vue

|-- main.js---------------应用入口js

### 各个vue组件的基本代码

* header.vue

|  |
| --- |
| <**template**>  <**div**>header vue template</**div**> </**template**>  <**script**>  **export default** {} </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

* goods.vue

|  |
| --- |
| <**template**>  <**div**>goods vue template</**div**> </**template**>  <**script**>  **export default** {} </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

* ratings.vue

|  |
| --- |
| <**template**>  <**div**>ratings vue template</**div**> </**template**>  <**script**>  **export default** {} </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

* seller.vue

|  |
| --- |
| <**template**>  <**div**>seller vue template</**div**> </**template**>  <**script**>  **export default** {} </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

* app.vue

|  |
| --- |
| <**template**>  <**div**>app vue template</**div**> </**template**>  <**script**>  **export default** {} </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  </**style**> |

* main.js

|  |
| --- |
| **import** Vue **from 'vue' import app** **from './app'** */\* eslint-disable no-new \*/* **new** Vue({  **el**: **'#app'**,  render: h => h(app) }) |

## 引入vue-router

### 下载vue-router

|  |
| --- |
| **npm install vue-router --save** |

### 编码

* router/index.js

|  |
| --- |
| **import** Vue **from 'vue' import** VueRouter **from 'vue-router' import** goods **from '../components/goods/goods.vue' import** ratings **from '../components/ratings/ratings.vue' import** seller **from '../components/seller/seller.vue'** Vue.use(VueRouter)  **export default new** VueRouter({  **linkActiveClass**: **'active'**,  **routes**: [  {  **path**: **'/'**,  **redirect**: **'/goods'** },  {  **path**: **'/goods'**,  **component**: goods  },  {  **path**: **'/ratings'**,  **component**: ratings  },  {  **path**: **'/seller'**,  **component**: seller  }  ] }) |

* index.js

|  |
| --- |
| **import** Vue **from 'vue' import** router **from './router' import** app **from './app'  import './common/stylus/fonts.styl'**  **new** Vue({  **el**: **'#app'**,  router,  **render**: h => h(app)}) |

* app.vue

|  |
| --- |
| <**template**>  <**div**>  <**ele-header**></**ele-header**>  <**div class="tab"**>  <**div class="tab-item"**>  <**router-link to="/goods"**>商品</**router-link**>  </**div**>  <**div class="tab-item"**>  <**router-link to="/ratings"**>评价</**router-link**>  </**div**>  <**div class="tab-item"**>  <**router-link to="/seller"**>商家</**router-link**>  </**div**>  </**div**>  <**router-view**></**router-view**>  </**div**> </**template**>  <**script**>  **import** header **from './components/header/header'   export default** {  *// 配置组件* **components**: {  **'ele-header'**: header  }  } </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  .**tab** .**tab-item  display inline-block** .**active  color**: **#ff0000** </**style**> |

* index.html

|  |
| --- |
| <**body**>  <**div id="app"**></**div**>  *<!-- built files will be auto injected -->* </**body**> |

### 引入reset样式

* static/css/reset.css

|  |
| --- |
| */\*\*  \* Eric Meyer's Reset CSS v2.0 (http://meyerweb.com/eric/tools/css/reset/)  \* http://cssreset.com  \*/* **html**, **body**, **div**, **span**, **applet**, **object**, **iframe**, **h1**, **h2**, **h3**, **h4**, **h5**, **h6**, **p**, **blockquote**, **pre**, **a**, **abbr**, **acronym**, **address**, **big**, **cite**, **code**, **del**, **dfn**, **em**, **img**, **ins**, **kbd**, **q**, **s**, **samp**, **small**, **strike**, **strong**, **sub**, **sup**, **tt**, **var**, **b**, **u**, **i**, **center**, **dl**, **dt**, **dd**, **ol**, **ul**, **li**, **fieldset**, **form**, **label**, **legend**, **table**, **caption**, **tbody**, **tfoot**, **thead**, **tr**, **th**, **td**, **article**, **aside**, **canvas**, **details**, **embed**, **figure**, **figcaption**, **footer**, **header**, **menu**, **nav**, **output**, **ruby**, **section**, **summary**, **time**, **mark**, **audio**, **video**, **input** {  **margin**: 0;  **padding**: 0;  **border**: 0;  **font-size**: 100%;  **font-weight**: **normal**;  **vertical-align**: **baseline**; }  */\* HTML5 display-role reset for older browsers \*/* **article**, **aside**, **details**, **figcaption**, **figure**, **footer**, **header**, **menu**, **nav**, **section** {  **display**: **block**; }  **body** {  **line-height**: 1; }  **blockquote**, **q** {  **quotes**: **none**; }  **blockquote**:**before**, **blockquote**:**after**, **q**:**before**, **q**:**after** {  **content**: **none**; }  **table** {  **border-collapse**: **collapse**;  **border-spacing**: 0; }  */\* custom \*/* **a** {  **color**: **#7e8c8d**;  **text-decoration**: **none**;  **-webkit-backface-visibility**: **hidden**; }  **li** {  **list-style**: **none**; }  ::**-webkit-scrollbar** {  **width**: 5**px**;  **height**: 5**px**; }  ::**-webkit-scrollbar-track-piece** {  **background-color**: **rgba**(0, 0, 0, 0.2);  **-webkit-border-radius**: 6**px**; }  ::**-webkit-scrollbar-thumb**:**vertical** {  **height**: 5**px**;  **background-color**: **rgba**(125, 125, 125, 0.7);  **-webkit-border-radius**: 6**px**; }  ::**-webkit-scrollbar-thumb**:**horizontal** {  **width**: 5**px**;  **background-color**: **rgba**(125, 125, 125, 0.7);  **-webkit-border-radius**: 6**px**; }  **html**, **body** {  **width**: 100%; }  **body** {  **-webkit-text-size-adjust**: **none**;  **-webkit-tap-highlight-color**: **rgba**(0, 0, 0, 0); } |

* index.html中引入

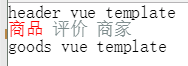
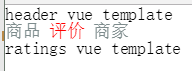
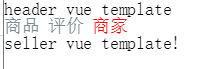
|  |
| --- |
| <**link rel="stylesheet" href="static/css/reset.css"**> |

### 添加viewport

* index.html

|  |
| --- |
| <**meta name="viewport"  content="width=device-width,initial-scale=1.0,maximum-scale=1.0,minimum-scale=1.0,user-scalable=no"**> |

### 运行效果

## 选项卡效果

### 说明

* 3个选项水平平均分配文字居中: flex布局
* 点击Item的整个区域都要能响应, 而不仅是文本区: display block
* 选项底部一个像素高度的横线: after伪类 + @media缩放

### 编码

* **app.vue**

|  |
| --- |
| <**div class="tab border-1px"**>  <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "./common/stylus/mixins.styl"** .**tab  display**: **flex  width**: 100%  **height**: 40**px  line-height**: 40**px****border-1px**(**rgba**(7, 17, 27, 0.1))  .**tab-item  flex**: 1  **text-align**: **center** & > **a  display**: **block  font-size**: 14**px  color**: **rgb**(77, 85, 93)  &.**active  color**: **rgb**(240, 20, 20) </**style**> |

## 模拟(mock)数据/接口

### Web应用前后端(台)分离:

* 后台向前台提供API接口, 只负责数据的提供和计算，而完全不处理展现
* 前台通过Http(Ajax)请求获取数据,　在浏览器端动态构建界面显示数据

### 设计JSON数据结构

* 理解JSON数据结构
* 编写模拟JSON数据: src/mock/data.json
* json数据
  + 结构: 名称, 数据类型
  + value
  + value可以变, 但结构不能变

### 利用Node+express提供模拟数据

1. **老版的vue脚手架项目**

* build/dev-server.js

|  |
| --- |
| *//加载json数据* **var** apiData = require(**'../src/mock/data.json'**) **var** seller = apiData.seller; **var** goods = apiData.goods; **var** ratings = apiData.ratings; *//得到路由器* **var** apiRouter = **express**.*Router*() *//注册路由* apiRouter.get(**'/seller'**, **function** (req, res) {  res.json({  **code**: 0, //0代表正确数据  **data**: seller  }) }) apiRouter.get(**'/goods'**, **function** (req, res) {  res.json({  **code**: 0,  **data**: goods  }) }) apiRouter.get(**'/ratings'**, **function** (req, res) {  res.json({  **code**: 0,  **data**: ratings  }) })  *//启用路由器* app.use(**'/api'**, apiRouter) |

**好处: 可以直接用浏览器/postman测试接口**

**缺点: 如果是打包发布就无法访问模拟数据, 只能是测试时使用**

### 利用mockjs提供模拟数据

* Mockjs: 用来拦截ajax请求, 生成随机数据返回
* 学习
  + <http://mockjs.com/>
  + <https://github.com/nuysoft/Mock>
* 下载:

|  |
| --- |
| **npm install mockjs --save** |

* 使用(mock/mockServer.js)

|  |
| --- |
| **import** Mock **from 'mockjs'**  **import** apiData **from './data.json'** Mock.mock(**'/api2/seller'**, {**code**:0, **data**:apiData.**seller**}) Mock.mock(**'/api2/goods'**, {**code**:0, **data**:apiData.**goods**}) Mock.mock(**'/api2/ratings'**, {**code**:0, **data**:apiData.**ratings**}) |

### 使用postman工具测试接口

* postman是用来测试API接口的chrome插件
* postman也是一个活接口文档

### ajax请求mock数据接口

* vue-resource是用于ajax请求的vue插件
* axios是浏览器/服务器端通用的ajax请求库
* 下载:

|  |
| --- |
| **npm install vue-resource axios --save** |

* 使用示例:

|  |
| --- |
| **import** *VueResource* **from 'vue-resource'** Vue.use(*VueResource*)  ***//组件对象多了$http属性, 使用它发送ajax请求***  **this**.**$http**.get(**'/api2/seller'**).then((response) => {  **console**.log(response.body) }, (response) => {  **console**.log(**'失败了'**) })  **import axios****from 'axios'**  axios.get(**'/api2/seller'**).then((response) => {  **console**.log(response.data) }, (response) => {  **console**.log(**'失败了'**) }) |

### 定义与后台ajax交互模块

* api/index.js

|  |
| --- |
| **import** axios **from 'axios' const** GOODS\_URL = **'/api2/goods' const** RATINGS\_URL = **'/api2/ratings' const** SELLER\_URL = **'/api2/seller'  export const** RESULT\_OK = 0  **export function** *reqGoods*() {  **return** axios.get(GOODS\_URL) }  **export function** *reqRatings* () {  **return** axios.get(RATINGS\_URL) }  **export function** *reqSeller* () {  **return** axios.get(SELLER\_URL) } |

## 使用vuex管理状态

### 下载vuex

|  |
| --- |
| **npm install --save vuex** |

### 定义state

* store/state.js

|  |
| --- |
| **export default** {  **goods**: [],  **ratings**: [],  **seller**: {} } |

### 定义types

* store/types.js

|  |
| --- |
| **export const** RECEIVE\_GOODS = **'receive\_goods' export const** RECEIVE\_SELLER = **'receive\_seller' export const** RECEIVE\_RATINGS = **'receive\_ratings'** |

### 定义mutations

* store/mutations.js

|  |
| --- |
| **import** {  RECEIVE\_GOODS,  RECEIVE\_RATINGS,  RECEIVE\_SELLER } **from './types'  export default** {  [RECEIVE\_GOODS] (state, {goods}) {  state.**goods** = goods  },  [RECEIVE\_SELLER] (state, {seller}) {  state.**seller** = seller  },  [RECEIVE\_RATINGS] (state, {ratings}) {  state.**ratings** = ratings  } } |

### 定义actions

* store/actions.js

|  |
| --- |
| **import** {  RECEIVE\_GOODS,  RECEIVE\_RATINGS,  RECEIVE\_SELLER } **from './types' import** {reqGoods, reqRatings, reqSeller, RESULT\_OK} **from '@/api'  export default** {  getSeller ({commit}, cb) {   reqSeller().then(response => {  **const** result = response.**data  if**(result.**code**==RESULT\_OK) {  **const** seller = result.**data** commit(RECEIVE\_SELLER, {seller})  cb && cb()  }  })  },   getRatings ({commit}, cb) {  reqRatings().then(response => {  **const** result = response.**data  if**(result.**code**==RESULT\_OK) {  **const** ratings = result.**data** commit(RECEIVE\_RATINGS, {ratings})  cb && cb()  }  })  },   getGoods ({commit}, cb) {  reqGoods().then(response => {  **const** result = response.**data  if**(result.**code**==RESULT\_OK) {  **const** goods = result.**data** commit(RECEIVE\_GOODS, {goods})  cb && cb()  }  })  } } |

### 定义入口

* store/index.js

|  |
| --- |
| **import** Vue **from 'vue' import** Vuex **from 'vuex' import** state **from './state' import** mutations **from './mutations' import** actions **from './actions' import** getters **from './getters'** Vue.*use*(Vuex)  **export default new** Vuex.Store({  state,  mutations,  actions,  getters }) |

## 开发header组件

### 在app.vue中通过vuex请求获取商家数据更新状态

|  |
| --- |
| mounted () {  **this**.$store.**dispatch**(**'getSeller'**) } |

### header组件样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**header  position**:**relative  color**:**rgb**(255,255,255)  **overflow**:**hidden  background-color**:**rgba**(7,17,27,0.5)  .**content-wrapper  position**:**relative  padding** 24**px** 12**px** 18**px** 24**px  font-size**:0  .**avatar  display**: **inline-block  vertical-align**: **top** .**content  display**: **inline-block  margin-left**: 16**px** .**title  margin**: 2**px** 0 8**px** 0  .**brand  margin-right**: 6**px  vertical-align**:**top  bg-image**(**brand**)  **display**: **inline-block  width**: 30**px  height**: 18**px  background-size**:30**px** 18**px  background-repeat**:**no-repeat** .**name  font-size**: 16**px  line-height**: 18**px  font-weight**: **bold** .**description  font-size**: 12**px  font-weight**: 200  **line-height**: 12**px** .**support  margin**: 10**px** 0 2**px** 0  .**icon  margin-right**:4**px  vertical-align**:**top  display**:**inline-block  width**: 12**px  height**: 12**px  background-size**: 12**px** 12**px  background-repeat**: **no-repeat**  &.**decrease  bg-image**(**decrease\_1**)&.**discount  bg-image**(**discount\_1**)  &.**guarantee  bg-image**(**guarantee\_1**)  &.**invoice  bg-image**(**invoice\_1**)  &.**special  bg-image**(**special\_1**)  .**text  font-size**: 10**px  font-weight**: 200  **line-height**: 12**px** .**supports\_count  position**:**absolute  right**: 12**px  bottom**: 15**px  height**: 24**px  padding**: 7**px** 8**px  border-radius**: 8**px  background-color**: **rgba**(0,0,0,0.2)  **box-sizing**:**border-box** .**count  vertical-align**:**top  font-size**: 10**px  line-height**: 12**px  font-weight**: 200  **margin-right**:2**px** .**icon-keyboard\_arrow\_right  font-size**: 10**px  line-height**: 12**px  font-weight**: 200  .**bulletin-wrapper  background-color**:**rgba**(7,17,27,0.2)  **position**:**relative  padding**: 0 20**px** 0 12**px  overflow**:**hidden  text-overflow**:**ellipsis  white-space**:**nowrap  display**:**block** .**bulletin-title  vertical-align**:**middle  display**:**inline-block  width**: 22**px  height**: 12**px  bg-image**(**bulletin**)  **background-size**:100% 100%  **background-repeat**:**no-repeat** .**bulletin-text  font-size**: 10**px  line-height**: 28**px** .**icon-keyboard\_arrow\_right  position**:**absolute  right**:12**px  bottom**:9**px  font-size**: 10**px** .**background  position**:**absolute  left**:0  **top**:0  **width**:100%  **height**:100%  **z-index**: -1  **img  filter**:**blur**(10**px**)  .**detail  position**:**fixed  left**:0  **top**:0  **z-index**:100  **width**:100%  **height**:100%  **background**:**rgba**(7,17,27,0.8)  **overflow**:**auto  transition**: .5**s  backdrop-filter**: **blur**(10**px**)  &.**fade-transition  opacity**:1  &.**fade-enter**,&.**fade-leave  opacity**:0  .**detail-wrapper  display**:**inline-block  width**:100%  **min-height**:100%  .**detail-main  margin-top**:64**px  padding-bottom**:64**px** .**name  text-align**:**center  font-size**: 16**px  line-height**: 16**px  font-weight**:700  .**star-wrapper  margin-top**:16**px  text-align**:**center** .**title  display**:**flex  width**:80%  **margin**: 28**px auto** 24**px** .**text  text-align**:**center  padding**: 0 12**px** .**line  flex**:1  **position**:**relative  top**:-6**px  border-bottom**:2**px solid rgba**(255, 192, 203,0.3)    .**supports  width**:80%  **padding**:0 12**px  margin**:0 **auto** .**support  margin-bottom**:12**px** &:**last-child  margin-bottom**:0  .**icon  display**:**inline-block  width**: 16**px  height**: 16**px  background-size**: 16**px** 16**px  background-repeat**:**no-repeat  vertical-align**:**top  margin-right**:6**px** &.**discount  bg-image**(**discount\_2**)  &.**decrease  bg-image**(**decrease\_2**)  &.**guarantee  bg-image**(**guarantee\_2**)  &.**invoice  bg-image**(**invoice\_2**)  &.**special  bg-image**(**special\_2**)   .**text  font-size**:12**px  line-height**: 12**px  font-weight**:200   .**content  width**:80%  **padding**:0 12**px  margin**: 0 **auto  font-size**: 12**px  line-height**: 24**px  font-weight**: 200    .**detail-close  margin-top**:-64**px  clear**:**both  text-align**:**center** .**icon-close  display**:**inline-block  width**: 32**px  height**: 32**px  font-size**:32**px** </**style**> |

### 图片资源

|  |
| --- |
| brand@3x.png  bulletin@2x.png  bulletin@3x.png  decrease\_1@2x.png  decrease\_1@3x.png  decrease\_2@2x.png  decrease\_2@3x.png  discount\_1@2x.png  discount\_1@3x.png  discount\_2@2x.png  discount\_2@3x.png  guarantee\_1@2x.png  guarantee\_1@3x.png  guarantee\_2@2x.png  guarantee\_2@3x.png  invoice\_1@2x.png  invoice\_1@3x.png  invoice\_2@2x.png  invoice\_2@3x.png  special\_1@2x.png  special\_1@3x.png  special\_2@2x.png  special\_2@3x.png |

### 显示头部基本信息

|  |
| --- |
| <**template**>  <**div class="header"**>  <**div class="content-wrapper"**>  <**div class="avatar"**>  <**img width="64" height="64" :src="seller.avatar"**>  </**div**>  <**div class="content"**>  <**div class="title"**>  <**span class="brand"**></**span**>  <**span class="name"**>{{seller.name}}</**span**>  </**div**>  <**div class="description"**>  {{seller.description}}/{{seller.deliveryTime}}分钟到达  </**div**>  <**div class="support" v-if="seller.supports"**>  <**span class="icon" :class="classMap[seller.supports[0].type]"**></**span**>  <**span class="text"**>{{seller.supports[0].description}}</**span**>  </**div**>  </**div**>  <**div class="supports\_count" v-if="seller.supports"**>  <**span class="count"**>{{seller.supports.length}}个</**span**>  <**span class="icon-keyboard\_arrow\_right"**></**span**>  </**div**>  </**div**>  <**div class="bulletin-wrapper"**>  <**span class="bulletin-title"**></**span**>  <**span class="bulletin-text"**>{{seller.bulletin}}</**span**>  <**i class="icon-keyboard\_arrow\_right"**></**i**>  </**div**>  <**div class="background"**>  <**img width="100%" height="100%" :src="seller.avatar"**>  </**div**>  <**div class="detail" v-show="false"**></**div**>  </**div**> </**template**>  <**script**>  **import** {mapState} **from 'vuex'  export default** {  data () {  **return** {  **supportClasses**: [**'decrease'**, **'discount'**, **'guarantee'**, **'invoice'**, **'special'**]  }  },  **computed**: **mapState(['seller'])**,  } </**script**> |

### 显示/隐藏优惠和公告详情

|  |
| --- |
| <**template**>  <**div class="header"**>  <**div class="content-wrapper"**>  <**div class="avatar"**>  <**img width="64" height="64" :src="seller.avatar"**>  </**div**>  <**div class="content"**>  <**div class="title"**>  <**span class="brand"**></**span**>  <**span class="name"**>{{seller.name}}</**span**>  </**div**>  <**div class="description"**>  {{seller.description}}/{{seller.deliveryTime}}分钟到达  </**div**>  <**div class="support" v-if="seller.supports"**>  <**span class="icon" :class="classMap[seller.supports[0].type]"**></**span**>  <**span class="text"**>{{seller.supports[0].description}}</**span**>  </**div**>  </**div**>  <**div class="supports\_count" v-if="seller.supports" @click="showDetail(true)"**>  <**span class="count"**>{{seller.supports.length}}个</**span**>  <**span class="icon-keyboard\_arrow\_right"**></**span**>  </**div**>  </**div**>  <**div class="bulletin-wrapper" @click="showDetail(true)"**>  <**span class="bulletin-title"**></**span**>  <**span class="bulletin-text"**>{{seller.bulletin}}</**span**>  <**i class="icon-keyboard\_arrow\_right"**></**i**>  </**div**>  <**div class="background"**>  <**img width="100%" height="100%" :src="seller.avatar"**>  </**div**>   **<div class="detail" v-show="detailShow">**  <**div class="detail-wrapper"**>  <**div class="detail-main clearfix"**>  <**div class="name"**>{{seller.name}}</**div**>  <**div class="star-wrapper"**>star组件内容</**div**>  <**div class="title"**>  <**div class="line"**></**div**>  <**div class="text"**>优惠信息</**div**>  <**div class="line"**></**div**>  </**div**>  <**ul class="supports" v-if="seller.supports"**>  <**li class="support" v-for="item in seller.supports"**>  <**span class="icon" :class="classMap[item.type]"**></**span**>  <**span class="text"**>{{item.description}}</**span**>  </**li**>  </**ul**>  <**div class="title"**>  <**div class="line"**></**div**>  <**div class="text"**>商家公告</**div**>  <**div class="line"**></**div**>  </**div**>  <**div class="content"**>  <**p**>{{seller.bulletin}}</**p**>  </**div**>  </**div**>  </**div**>  <**div class="detail-close" @click="showDetail(false)"**>  <**span class="icon-close"**></**span**>  </**div**>   **</div>**  </**div**> </**template**>  <**script**>  **import** {mapState} **from 'vuex'  export default** {  data () {  **return** {  **isShow**: **false**,  **supportClasses**: [**'decrease'**, **'discount'**, **'guarantee'**, **'invoice'**, **'special'**]  }  },   **methods**: {  **toggleShow ()** {  **this**.**isShow** = !**this**.**isShow** }  },   **computed**: mapState([**'seller'**]),  } </**script**> |

## 开发star组件

### 图片

|  |
| --- |
| star24\_half@2x.png  star24\_half@3x.png  star24\_off@2x.png  star24\_off@3x.png  star24\_on@2x.png  star24\_on@3x.png  star36\_half@2x.png  star36\_half@3x.png  star36\_off@2x.png  star36\_off@3x.png  star36\_on@2x.png  star36\_on@3x.png  star48\_half@2x.png  star48\_half@3x.png  star48\_off@2x.png  star48\_off@3x.png  star48\_on@2x.png  star48\_on@3x.png |

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**star  font-size**: 0  .**star-item  display**: **inline-block  background-repeat**: **no-repeat** &.**star-48** .**star-item  width**: 20**px  height**: 20**px  margin-right**: 22**px  background-size**: 20**px** 20**px** &:**last-child  margin-right**: 0  &.**on  bg-image**(**'star48\_on'**)  &.**half  bg-image**(**'star48\_half'**)  &.**off  bg-image**(**'star48\_off'**)  &.**star-36** .**star-item  width**: 15**px  height**: 15**px  margin-right**: 6**px  background-size**: 15**px** 15**px** &:**last-child  margin-right**: 0  &.**on  bg-image**(**'star36\_on'**)  &.**half  bg-image**(**'star36\_half'**)  &.**off  bg-image**(**'star36\_off'**)  &.**star-24** .**star-item  width**: 10**px  height**: 10**px  margin-right**: 3**px  background-size**: 10**px** 10**px** &:**last-child  margin-right**: 0  &.**on  bg-image**(**'star24\_on'**)  &.**half  bg-image**(**'star24\_half'**)  &.**off  bg-image**(**'star24\_off'**) </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div class="star" :class="'star-'+size"**>  <**span class="star-item" v-for="(item, index) in starClasses"  :class="item" :key="index"**></**span**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script type="text/ecmascript-6"**>  **const *LENGTH*** = 5  **const *CLASS\_ON*** = **'on'  const *CLASS\_HALF*** = **'half'  const *CLASS\_OFF*** = **'off'   export default**{  **props**: {  **score**: {  **type**: Number  },  **size**: {  **type**: Number  }  },   **computed**: {  starClasses () {  **const** scs = []  **const** score = **this**.score  *//去除小数部分求整* **const** scoreInteger = Math.floor(score)  **for** (let i = 0; i < scoreInteger; i++) {  scs.push(***CLASS\_ON***)  }  **if**(score-scoreInteger>=0.5) {  scs.push(***CLASS\_HALF***)  }  **while**(scs.**length**<***LENGTH***) {  scs.push(***CLASS\_OFF***)  }   **return** scs  }  }  } </**script**> |

### 使用组件标签

|  |
| --- |
| //使用组件标签, 传递属性数据  <**star :score="seller.score" :size="48"**></**star**>  //引入依赖组件模块  **import** star **from '../star/star'**  //定义组件标签  **components**: {  star } |

## 开发goods组件

### 说明

* goods组件是一个较复杂的路由组件
* 内部使用了另外3个组件
  + Shopcart: 购物车组件
  + Cartcontrol: 购物车操作组件
  + food: 食品详情组件
* 使用第三方库better-scroll: UI滑动

### 图片资源

|  |
| --- |
| special\_3@2x.png  invoice\_3@3x.png  guarantee\_3@3x.png  discount\_3@2x.png  invoice\_3@2x.png  guarantee\_3@2x.png  decrease\_3@3x.png  decrease\_3@2x.png  discount\_3@3x.png  special\_3@3x.png |

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**goods  display**: **flex  position**: **absolute  top**: 174**px  bottom**: 46**px  width**: 100%  **overflow**: **hidden** .**menu-wrapper  flex**: 0 0 80**px  width**: 80**px  background**: **#f3f5f7** .**menu-item  display**: **table  height**: 54**px  width**: 56**px  padding**: 0 12**px  line-height**: 14**px** &.**current  position**: **relative  z-index**: 10  **margin-top**: -1**px  background**: **#fff  font-weight**: 700  .**text  border-none**()  .**icon  display**: **inline-block  vertical-align**: **top  width**: 12**px  height**: 12**px  margin-right**: 2**px  background-size**: 12**px** 12**px  background-repeat**: **no-repeat** &.**decrease  bg-image**(**'decrease\_3'**)  &.**discount  bg-image**(**'discount\_3'**)  &.**guarantee  bg-image**(**'guarantee\_3'**)  &.**invoice  bg-image**(**'invoice\_3'**)  &.**special  bg-image**(**'special\_3'**)  .**text  display**: **table-cell  width**: 56**px  vertical-align**: **middle  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 12**px** .**foods-wrapper  flex**: 1  .**title  padding-left**: 14**px  height**: 26**px  line-height**: 26**px  border-left**: 2**px solid #d9dde1  font-size**: 12**px  color**: **rgb**(147, 153, 159)  **background**: **#f3f5f7** .**food-item  display**: **flex  margin**: 18**px  padding-bottom**: 18**px  border-1px**(**rgba**(7, 17, 27, 0.1))  &:**last-child  border-none**()  **margin-bottom**: 0  .**icon  flex**: 0 0 57**px  margin-right**: 10**px** .**content  flex**: 1  .**name  margin**: 2**px** 0 8**px** 0  **height**: 14**px  line-height**: 14**px  font-size**: 14**px  color**: **rgb**(7, 17, 27)  .**desc**, .**extra  line-height**: 10**px  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**desc  line-height**: 12**px  margin-bottom**: 8**px** .**extra** .**count  margin-right**: 12**px** .**price  font-weight**: 700  **line-height**: 24**px** .**now  margin-right**: 8**px  font-size**: 14**px  color**: **rgb**(240, 20, 20)  .**old  text-decoration**: **line-through  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**cartcontrol-wrapper  position**: **absolute  right**: 0  **bottom**: 12**px** </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <div>  <**div class="goods"**>  <**div class="menu-wrapper" ref="menuWrapper"**>  <**ul**>  <**li class="menu-item" v-for="(item, index) in goods"  :class="{current:currentIndex===index}"  @click="clickMenuItem(index,$event)"**>  <**span class="text border-1px"**>  <**span class="icon" v-if="item.type>0" :class="classMap[item.type]"**></**span**>{{item.name}}  </**span**>  </**li**>  </**ul**>  </**div**>  <**div class="foods-wrapper" ref="foodsWrapper"**>  <**ul**>  <**li class="food-list food-list-hook" v-for="item in goods"**>  <**h1 class="title"**>{{item.name}}</**h1**>  <**ul**>  <**li class="food-item border-1px" v-for="food in item.foods"**>  <**div class="icon"**>  <**img width="57" height="57"  :src="food.icon"**>  </**div**>  <**div class="content"**>  <**h2 class="name"**>{{food.name}}</**h2**>  <**p class="desc"**>{{food.description}}</**p**>  <**div class="extra"**>  <**span class="count"**>月售{{food.sellCount}}份</**span**>  <**span**>好评率{{food.rating}}%</**span**>  </**div**>  <**div class="price"**>  <**span class="now"**>￥{{food.price}}</**span**>  <**span class="old" v-show="food.oldPrice"**>￥{{food.oldPrice}}</**span**>  </**div**>  <**div class="cartcontrol-wrapper"**>  cartcontrol  </**div**>  </**div**>  </**li**>  </**ul**>  </**li**>  </**ul**>  </**div**>  </**div**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script type="text/ecmascript-6"**>  **import** {mapState} **from 'vuex'**  **import** BScroll **from 'better-scroll'   export default**{  data () {  **return** {  **classMap:** [**'decrease'**, **'discount'**, **'special'**, **'invoice'**, **'guarantee'**]  **tops**: [],  **scrollY**: 0,  }  },  mounted () {  *// 使用axios请求mockjs提供的接口* **this**.$store.**dispatch**(**'getGoods'**, () => {  */\*setTimeout(() => {  this.\_initScroll()  }, 1)\*/  // 将回调延迟到下次 DOM 更新循环之后执行* **this**.$nextTick(() => {  **this**.\_initScroll()  **this**.\_initTops()  })  })  },  **methods**: {  \_initScroll () {  **this**.**menuScroll** = **new** BScroll(**this**.$refs.menuWrapper, {  **click**: **true** })  **this**.**foodsScroll** = **new** BScroll(**this**.$refs.foodsWrapper, {  **click**: **true**,  **probeType**: 3  })  *//监视foods的滚动* **this**.**foodsScroll**.on(**'scroll'**, (pos) => {  *// console.log(pos.x + '~' + pos.y)* **this**.**scrollY** = Math.abs(Math.round(pos.**y**))  })  },  \_initTops () {  **const** foodList = **this**.$refs.foodsWrapper.getElementsByClassName(**'food-list-hook'**)  let top = 0  **this**.tops.push(top)  **for**(let i=0;i<foodList.**length**;i++) {  let item = foodList[i]  top += item.**clientHeight  this**.tops.push(top)  }  console.log(**this**.tops)  },   clickMenuItem (index, event) {  console.log(event.**\_constructed**, -**this**.tops[index])  **if**(!event.**\_constructed**) {  **return** }  **const** foodList = **this**.$refs.foodsWrapper.getElementsByClassName(**'food-list-hook'**)  **const** el = foodList[index]  **this**.**foodsScroll**.scrollToElement(el, 300)  }  },   **computed**: {  ...mapState([**'seller'**, **'goods'**]),  currentIndex () {  **return this**.tops.*findIndex*((top, index)=> {  **return this**.scrollY>=top && **this**.scrollY<**this**.tops[index+1]  })  }  }  } </**script**> |

## 开发cartcontrol组件

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  .**cartcontrol  font-size**: 0  .**cart-decrease  display**: **inline-block  padding**: 6**px  line-height**: 24**px  font-size**: 24**px  color**: **rgb**(0, 160, 220)  &.**move-enter-active**, &.**move-leave-active  transition**: **all** .4**s linear** &.**move-enter**, &.**move-leave-active  opacity**: 0  **transform**: **translate3d**(24**px**, 0, 0) **rotate**(180**deg**)  .**cart-count  display**: **inline-block  vertical-align**: **top  width**: 12**px  padding-top**: 6**px  line-height**: 24**px  text-align**: **center  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**cart-add  display**: **inline-block  padding**: 6**px  line-height**: 24**px  font-size**: 24**px  color**: **rgb**(0, 160, 220) </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div class="cartcontrol"**>  <**transition name="move"**>  <**div class="cart-decrease icon-remove\_circle\_outline" v-show="food.count"  @click.stop="updateFoodCount(food, false, $event)"**>  </**div**>  </**transition**>  <**div class="cart-count" v-show="food.count"**>{{food.count}}</**div**>  <**div class="cart-add icon-add\_circle"  @click.stop="updateFoodCount(food, true, $event)"**></**div**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **export default** {  **props**: {  **food**: Object,  **updateFoodCount**: Function  }  } </**script**> |

### goods.vue

|  |
| --- |
| <**div class="cartcontrol-wrapper"**>  <**cartcontrol :food="food" :update-food-count="updateFoodCount"**></**cartcontrol**> </**div**>  updateFoodCount (food, isAdd, event) { *// 过滤系统的点击回调* **if**(!event.**\_constructed**) {  **return** } **if**(isAdd) { *// 加* **if**(!food.**count**) { *// 第一次  // food.count = 1 // 新增的属性没有监视* Vue.set(food, **'count'**, 1) *// 有监视* } **else** {  food.**count**++  } } **else** { *// 减* **if**(food.**count**) {  food.**count**--  } } } |

## 开发shopcart组件

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**shopcart  position**: **fixed  left**: 0  **bottom**: 0  **z-index**: 50  **width**: 100%  **height**: 48**px** .**content  display**: **flex  background**: **#141d27  font-size**: 0  **color**: **rgba**(255, 255, 255, 0.4)  .**content-left  flex**: 1  .**logo-wrapper  display**: **inline-block  vertical-align**: **top  position**: **relative  top**: -10**px  margin**: 0 12**px  padding**: 6**px  width**: 56**px  height**: 56**px  box-sizing**: **border-box  border-radius**: 50%  **background**: **#141d27** .**logo  width**: 100%  **height**: 100%  **border-radius**: 50%  **text-align**: **center  background**: **#2b343c** &.**highlight  background**: **rgb**(0, 160, 220)  .**icon-shopping\_cart  line-height**: 44**px  font-size**: 24**px  color**: **#80858a** &.**highlight  color**: **#fff** .**num  position**: **absolute  top**: 0  **right**: 0  **width**: 24**px  height**: 16**px  line-height**: 16**px  text-align**: **center  border-radius**: 16**px  font-size**: 9**px  font-weight**: 700  **color**: **#fff  background**: **rgb**(240, 20, 20)  **box-shadow**: 0 4**px** 8**px** 0 **rgba**(0, 0, 0, 0.4)  .**price  display**: **inline-block  vertical-align**: **top  margin-top**: 12**px  line-height**: 24**px  padding-right**: 12**px  box-sizing**: **border-box  border-right**: 1**px solid rgba**(255, 255, 255, 0.1)  **font-size**: 16**px  font-weight**: 700  &.**highlight  color**: **#fff** .**desc  display**: **inline-block  vertical-align**: **top  margin**: 12**px** 0 0 12**px  line-height**: 24**px  font-size**: 10**px** .**content-right  flex**: 0 0 105**px  width**: 105**px** .**pay  height**: 48**px  line-height**: 48**px  text-align**: **center  font-size**: 12**px  font-weight**: 700  &.**not-enough  background**: **#2b333b** &.**enough  background**: **#00b43c  color**: **#fff** .**ball-container** .**ball  position**: **fixed  left**: 32**px  bottom**: 22**px  z-index**: 200  **transition**: **all** 0.4**s cubic-bezier**(0.49, -0.29, 0.75, 0.41)  .**inner  width**: 16**px  height**: 16**px  border-radius**: 50%  **background**: **rgb**(0, 160, 220)  **transition**: **all** 0.4**s linear** .**shopcart-list  position**: **absolute  left**: 0  **top**: 0  **z-index**: -1  **width**: 100%  **transform**: **translate3d**(0, -100%, 0)  &.**fold-enter-active**, &.**fold-leave-active  transition**: **all** 0.5**s** &.**fold-enter**, &.**fold-leave-active  transform**: **translate3d**(0, 0, 0)  .**list-header  height**: 40**px  line-height**: 40**px  padding**: 0 18**px  background**: **#f3f5f7  border-bottom**: 1**px solid rgba**(7, 17, 27, 0.1)  .**title  float**: **left  font-size**: 14**px  color**: **rgb**(7, 17, 27)  .**empty  float**: **right  font-size**: 12**px  color**: **rgb**(0, 160, 220)   .**list-content  padding**: 0 18**px  max-height**: 217**px  overflow**: **hidden  background**: **#fff** .**food  position**: **relative  padding**: 12**px** 0  **box-sizing**: **border-box  border-1px**(**rgba**(7, 17, 27, 0.1))  .**name  line-height**: 24**px  font-size**: 14**px  color**: **rgb**(7, 17, 27)  .**price  position**: **absolute  right**: 90**px  bottom**: 12**px  line-height**: 24**px  font-size**: 14**px  font-weight**: 700  **color**: **rgb**(240, 20, 20)  .**cartcontrol-wrapper  position**: **absolute  right**: 0  **bottom**: 6**px** .**list-mask  position**: **fixed  top**: 0  **left**: 0  **width**: 100%  **height**: 100%  **z-index**: 40  **backdrop-filter**: **blur**(10**px**)  **opacity**: 1  **background**: **rgba**(7, 17, 27, 0.6)  &.**fade-enter-active**, &.**fade-leave-active  transition**: **all** 0.5**s** &.**fade-enter**, &.**fade-leave-to  opacity**: 0  **background**: **rgba**(7, 17, 27, 0) </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div**>  <**div class="shopcart"**>  <**div class="content"**>  <**div class="content-left" @click="toggleList"**>   <**div class="logo-wrapper"**>  <**div class="logo" :class="{highlight: foodList.length}"**>  <**i class="icon-shopping\_cart highlight"**></**i**>  </**div**>  <**div class="num" v-if="totalCount"**>{{totalCount}}</**div**>  </**div**>   <**div class="price"**>￥{{totalPrice}}</**div**>   <**div class="desc"**>另需配送费￥{{deliveryPrice}}元</**div**>  </**div**>   <**div class="content-right" @click="pay"**>  <**div class="pay" :class="payClass"**>  {{payText}}  </**div**>  </**div**>  </**div**>  <**div class="ball-container"**>   </**div**>  <**transition name="fold"**>  <**div class="shopcart-list" v-show="listShow"**>  <**div class="list-header"**>  <**h1 class="title"**>购物车</**h1**>  <**span class="empty" @click="clear"**>清空</**span**>  </**div**>  <**div class="list-content" ref="listContent"**>  <**ul**>  <**li class="food" v-for="food in foodList"**>  <**span class="name"**>{{food.name}}</**span**>  <**div class="price"**>  <**span**>￥{{food.price}}</**span**>  </**div**>  <**div class="cartcontrol-wrapper"**>  <**cartcontrol :food="food" :update-food-count="updateFoodCount"**></**cartcontrol**>  </**div**>  </**li**>  </**ul**>  </**div**>  </**div**>  </**transition**>   </**div**>   <**transition name="fade"**>  <**div class="list-mask" v-show="listShow" @click="toggleList"**></**div**>  </**transition**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **import** BScroll **from 'better-scroll'  import** cartcontrol **from '../cartcontrol/cartcontrol.vue'   export default** {  **props**: {  **foodList**: Array,  **minPrice**: Number,  **deliveryPrice**: Number,  **updateFoodCount**: Function  },   data () {  **return** {  **isShow**: **false**,  }  },   **computed**: {  totalCount () {  **var** count = 0  **this**.foodList.forEach(food=> {  count += food.**count** })  **return** count  },  totalPrice () {  **var** price = 0  **this**.foodList.forEach(food=> {  price += food.**count**\*food.price  })  **return** price  },  payText () {  **if**(**this**.totalCount===0) {  **return `￥**${**this**.minPrice}**元起送`** } **else if**(**this**.totalPrice<**this**.minPrice) {  **return `还差￥**${**this**.minPrice-**this**.totalPrice}**元起送`** } **else** {  **return '去结算'** }  },  payClass () {  **return this**.totalPrice<**this**.minPrice ? **'not-enough'** : **'enough'** },   listShow () {   **if**(**this**.foodList.**length**===0) {  **this**.**isShow** = **false  return false** }   **if**(**this**.**isShow**) {  **this**.$nextTick(() => {*// 延迟到界面更新完成后调用* **if**(!**this**.**scroll**) {  **this**.**scroll** = **new** BScroll(**this**.$refs.listContent, {  **click**: **true** })  } **else** {  **this**.**scroll**.refresh() *//刷新形成滚动* }  })  }   **return this**.**isShow** }  },   **methods**: {  toggleList () {  **if**(**this**.foodList.**length**) {  **this**.**isShow** = !**this**.**isShow** }  },  clear () {  **if**(confirm(**'确定清空购物车吗?'**)) {  *// 触发事件* **this**.$emit(**'clear'**, **this**.foodList)  }  },   pay () {  **if**(**this**.totalPrice-**this**.minPrice>=0) {  alert(**`支付**${**this**.totalPrice+**this**.deliveryPrice}**`**)  }  }  },   **components**: {  cartcontrol  }  } </**script**> |

### 使用组件标签

* goods.vue

|  |
| --- |
| <**shopcart :food-list="foodList" :min-price="seller.minPrice"  :delivery-price="seller.deliveryPrice"  :update-food-count="updateFoodCount"  @clear="clearCart"  ref="shopcart"**></**shopcart**>  **import** shopcart **from '../shopcart/shopcart'**  **export default**{  **props**: {  **seller**: {  **type**: Object  }  },  methods: {  clearCart () {  **this**.foodList.forEach(food => {  food.**count** = 0  })  }  },  **computed**: {  foodList () { *// 找出所有count>0的food* **const** foods = []  **this**.goods.forEach(good => {  good.foods.forEach(food => {  **if**(food.**count**) {  foods.push(food)  }  })  })  **return** foods  }  },   **components**: {  shopcart  } } |

* App.vue

|  |
| --- |
| <**router-view :seller="seller"**></**router-view**> |

## 开发ratingselect组件

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**ratingselect** .**rating-type  padding**: 18**px** 0  **margin**: 0 18**px  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 0  .**block  display**: **inline-block  padding**: 8**px** 12**px  margin-right**: 8**px  line-height**: 16**px  border-radius**: 1**px  font-size**: 12**px  color**: **rgb**(77, 85, 93)  &.**active  color**: **#fff** .**count  margin-left**: 2**px  font-size**: 8**px** &.**positive  background**: **rgba**(0, 160, 220, 0.2)  &.**active  background**: **rgb**(0, 160, 220)  &.**negative  background**: **rgba**(77, 85, 93, 0.2)  &.**active  background**: **rgb**(77, 85, 93)  .**switch  padding**: 12**px** 18**px  line-height**: 24**px  border-bottom**: 1**px solid rgba**(7, 17, 27, 0.1)  **color**: **rgb**(147, 153, 159)  **font-size**: 0  &.**on** .**icon-check\_circle  color**: **#00c850** .**icon-check\_circle  display**: **inline-block  vertical-align**: **top  margin-right**: 4**px  font-size**: 24**px** .**text  display**: **inline-block  vertical-align**: **top  font-size**: 12**px** </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div class="ratingselect"**>  <**div class="rating-type border-1px"**>  <**span class="block positive" :class="{active: selectType===2}" @click="setSelectType(2, $event)"**>  {{desc.all}}<**span class="count"**>{{ratings.length}}</**span**>  </**span**>  <**span class="block positive" :class="{active: selectType===0}" @click="setSelectType(0, $event)"**>  {{desc.positive}}<**span class="count"**>{{positiveLength}}</**span**>  </**span**>  <**span class="block negative" :class="{active: selectType===1}" @click="setSelectType(1, $event)"**>  {{desc.negative}}<**span class="count"**>{{ratings.length-positiveLength}}</**span**>  </**span**>  </**div**>   <**div class="switch" :class="{on: onlyContent}" @click="switchOnlyContent"**>  <**span class="icon-check\_circle"**></**span**>  <**span class="text"**>只看有内容的评价</**span**>  </**div**>   </**div**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **export default** {  **props**: {  **desc**: Object,  **onlyContent**: Boolean,  **ratings**: Array,  **selectType**: Number  },   **computed**: {  positiveLength () {  */\*  var total = 0  this.ratings.forEach(rating => {  total += rating.rateType===0 ? 1 : 0  })\*/    // 办法一: 过滤得到数组, 取出数组的length  // return this.ratings.filter(rating => rating.rateType===0).length  // 方法二: 使用reduce进行统计* **return this**.ratings.reduce((preTotal, rating) => {  **return** preTotal + (rating.rateType===0 ? 1 : 0)  }, 0)  }  },   **methods**: {  setSelectType (type, event) {  **if**(!event.**\_constructed**) {  **return** }  *// 触发事件* **this**.$emit(**'setselecttype'**, type)  },   switchOnlyContent (event) {  **if**(!event.**\_constructed**) {  **return** }  *// 触发事件* **this**.$emit(**'switchonlycontent'**)  }  }  } </**script**> |

## 开发food组件

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixins.styl"** .**food  position**: **fixed  left**: 0  **top**: 0  **bottom**: 48**px  z-index**: 30  **width**: 100%  **background**: **#fff** &.**move-enter-active**, &.**move-leave-active  transition**: **transform** .2**s linear** &.**move-enter**, &.**move-leave-active  transform**: **translate3d**(100%, 0, 0)   .**image-header  position**: **relative  width**: 100%  **height**: 0  **padding-top**: 100%  **img  position**: **absolute  top**: 0  **left**: 0  **width**: 100%  **height**: 100%  .**back  position**: **absolute  top**: 10**px  left**: 0  .**icon-arrow\_lift  display**: **block  padding**: 10**px  font-size**: 20**px  color**: **#fff** .**content  position**: **relative  padding**: 18**px** .**title  line-height**: 14**px  margin-bottom**: 8**px  font-size**: 14**px  font-weight**: 700  **color**: **rgb**(7, 17, 27)  .**detail  margin-bottom**: 18**px  line-height**: 10**px  height**: 10**px  font-size**: 0  .**sell-count**, .**rating  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**sell-count  margin-right**: 12**px** .**price  font-weight**: 700  **line-height**: 24**px** .**now  margin-right**: 8**px  font-size**: 14**px  color**: **rgb**(240, 20, 20)  .**old  text-decoration**: **line-through  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**cartcontrol-wrapper  position**: **absolute  right**: 12**px  bottom**: 12**px** .**buy  position**: **absolute  right**: 18**px  bottom**: 18**px  z-index**: 10  **height**: 24**px  line-height**: 24**px  padding**: 0 12**px  box-sizing**: **border-box  border-radius**: 12**px  font-size**: 10**px  color**: **#fff  background**: **rgb**(0, 160, 220)  &.**fade-transition  transition**: **all** 0.2**s  opacity**: 1  &.**fade-enter**, &.**fade-leave  opacity**: 0  .**info  padding**: 18**px** .**title  line-height**: 14**px  margin-bottom**: 6**px  font-size**: 14**px  color**: **rgb**(7, 17, 27)  .**text  line-height**: 24**px  padding**: 0 8**px  font-size**: 12**px  color**: **rgb**(77, 85, 93)  .**rating  padding-top**: 18**px** .**title  line-height**: 14**px  margin-left**: 18**px  font-size**: 14**px  color**: **rgb**(7, 17, 27)  .**rating-wrapper  padding**: 0 18**px** .**rating-item  position**: **relative  padding**: 16**px** 0  **border-1px**(**rgba**(7, 17, 27, 0.1))  .**user  position**: **absolute  right**: 0  **top**: 16**px  line-height**: 12**px  font-size**: 0  .**name  display**: **inline-block  margin-right**: 6**px  vertical-align**: **top  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**avatar  border-radius**: 50%  .**time  margin-bottom**: 6**px  line-height**: 12**px  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**text  line-height**: 16**px  font-size**: 12**px  color**: **rgb**(7, 17, 27)  .**icon-thumb\_up**, .**icon-thumb\_down  margin-right**: 4**px  line-height**: 16**px  font-size**: 12**px** .**icon-thumb\_up  color**: **rgb**(0, 160, 220)  .**icon-thumb\_down  color**: **rgb**(147, 153, 159)   .**no-rating  padding**: 16**px** 0  **font-size**: 12**px  color**: **rgb**(147, 153, 159) </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**transition name="move"**>  <**div class="food" v-show="isShow" ref="food"**>  <**div class="food-content"**>   <**div class="image-header"**>  <**img :src="food.image"**>  <**div class="back" @click="showOrHide(false)"**>  <**i class="icon-arrow\_lift"**></**i**>  </**div**>  </**div**>   <**div class="content"**>  <**h1 class="title"**>{{food.name}}</**h1**>  <**div class="detail"**>  <**span class="sell-count"**>月售{{food.sellCount}}份</**span**>  <**span class="rating"**>好评率{{food.rating}}%</**span**>  </**div**>  <**div class="price"**>  <**span class="now"**>￥{{food.price}}</**span**>  <**span class="old" v-show="food.oldPrice"**>￥{{food.oldPrice}}</**span**>  </**div**>  <**div class="cartcontrol-wrapper"**>  <**cartcontrol :food="food" :update-food-count="updateFoodCount"**></**cartcontrol**>  </**div**>  <**div class="buy" v-if="!food.count"  @click="updateFoodCount(food, true, $event)"**>加入购物车</**div**>  </**div**>   <**split**></**split**>   <**div class="info"**>  <**h1 class="title"**>商品信息</**h1**>  <**p class="text"**>{{food.info}}</**p**>  </**div**>   <**split**></**split**>   <**div class="rating"**>  <**h1 class="title"**>商品评价</**h1**>  <**ratingselect :desc="desc"  :only-content="onlyContent"  :ratings="food.ratings"  :select-type="selectType"  @switchonlycontent="switchOnlyContent"  @setselecttype="setSelectType"  v-if="food.ratings"**></**ratingselect**>   <**div class="rating-wrapper"**>  <**ul**>  <**li class="rating-item border-1px" v-for="rating in filterRatings"**>  <**div class="user"**>  <**span class="name"**>{{rating.username}}</**span**>  <**img class="avatar" width="12" height="12"  :src="rating.avatar"**>  </**div**>  <**div class="time"**>{{rating.rateTime}}</**div**>  <**p class="text"**>  <**span :class="{'icon-thumb\_up': rating.rateType==0, 'icon-thumb\_down': rating.rateType==1}"**></**span**>{{rating.text}}  </**p**>  </**li**>  </**ul**>  <**div class="no-rating" v-show="!food.ratings || food.ratings.length===0"**>暂无评价</**div**>  </**div**>  </**div**>  </**div**>  </**div**>  </**transition**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **import** BScroll **from 'better-scroll'  import** cartcontrol **from '../cartcontrol/cartcontrol.vue'  import** split **from '../split/split.vue'  import** ratingselect **from '../ratingselect/ratingselect.vue'   const *ALL*** = 2 *// 全部* **export default** {  **props**: {  **food**: Object,  **updateFoodCount**: Function  },   data () {  **return** {  **isShow**: **false**,  **onlyContent**: **false**,  **selectType**: ***ALL*** }  },   created () {  **this**.**desc** = {  **all**: **'全部'**,  **positive**: **'推荐'**,  **negative**: **'吐糟'** }  },   **methods**: {  showOrHide(isShow) { *// 显示/隐藏当前food* **this**.**isShow** = isShow   **if**(**this**.**isShow**) { *// 只有显示时才需要做* **this**.$nextTick(() => {  **if**(!**this**.**scroll**) {  **this**.**scroll** = **new** BScroll(**this**.$refs.**food**, {  **click**:**true** })  } **else** {  **this**.**scroll**.refresh()  }  })  }  },   *// 切换onlyContent* switchOnlyContent () {  **this**.**onlyContent** = !**this**.**onlyContent** *// 刷新列表* **this**.$nextTick(() => {  **this**.**scroll**.refresh()  })  },  *// 更新selectType* setSelectType (selectType) {  **this**.**selectType** = selectType  *// 刷新列表* **this**.$nextTick(() => {  **this**.**scroll**.refresh()  })  }  },   **computed**: {  filterRatings () {  *// 如果还没有数据, 结束* **if**(!**this**.food.ratings) {  **return** []  }   **const** ratings = **this**.food.ratings  **const** selectType = **this**.selectType  **const** onlyContent = **this**.onlyContent  **return** ratings.*filter*(rating => {  **var** {rateType, text} = rating *// 解构赋值* **if**(selectType===2) {  */\*if(!onlyContent) {  return true  } else {  return text.length>0  }\*/* **return** !onlyContent || text.**length**>0  } **else** {  **return** selectType===rateType && (!onlyContent || text.**length**>0)  }  })  }  },   **components**: {  cartcontrol,  split,  ratingselect  }  } </**script**> |

### 开发分隔块组件(简单): split.vue

|  |
| --- |
| <**template**>  <**div class="split"**></**div**> </**template**>  <**script type="text/ecmascript-6"**>  **export default** {   } </**script**>  <**style lang="stylus" rel="stylesheet/stylus"**>  .**split  width**: 100%  **height**: 16**px  border-top**: 1**px solid rgba**(7, 17, 27, 0.1)  **border-bottom**: 1**px solid rgba**(7, 17, 27, 0.1)  **background**: **#f3f5f7** </**style**> |

### 使用组件标签

|  |
| --- |
| <**li class="food-item border-1px" v-for="food in item.foods"  @click="selectFood(food, $event)"**>  <**food :food="selectedFood" :update-food-count="updateFoodCount" ref="food"**></**food**>  data () {  **return** {  **selectedFood**: {}  } },  methods: {  selectFood(food, event) {  **if** (!event.**\_constructed**) {  **return** }  **this**.**selectedFood** = food  **this**.$refs.**food**.**showOrHide**(true)  } }  **components**: {  food } |

## 开发ratings组件

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixin.styl"** .**ratings  position**: **absolute  top**: 174**px  bottom**: 0  **left**: 0  **width**: 100%  **overflow**: **hidden** .**overview  display**: **flex  padding**: 18**px** 0  .**overview-left  flex**: 0 0 137**px  padding**: 6**px** 0  **width**: 137**px  border-right**: 1**px solid rgba**(7, 17, 27, 0.1)  **text-align**: **center  @media only screen and** (**max-width**: 320**px**)  **flex**: 0 0 120**px  width**: 120**px** .**score  margin-bottom**: 6**px  line-height**: 28**px  font-size**: 24**px  color**: **rgb**(255, 153, 0)  .**title  margin-bottom**: 8**px  line-height**: 12**px  font-size**: 12**px  color**: **rgb**(7, 17, 27)  .**rank  line-height**: 10**px  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**overview-right  flex**: 1  **padding**: 6**px** 0 6**px** 24**px  @media only screen and** (**max-width**: 320**px**)  **padding-left**: 6**px** .**score-wrapper  margin-bottom**: 8**px  font-size**: 0  .**title  display**: **inline-block  line-height**: 18**px  vertical-align**: **top  font-size**: 12**px  color**: **rgb**(7, 17, 27)  .**stars  display**: **inline-block  margin**: 0 12**px  vertical-align**: **top** .**score  display**: **inline-block  line-height**: 18**px  vertical-align**: **top  font-size**: 12**px  color**: **rgb**(255, 153, 0)  .**delivery-wrapper  font-size**: 0  .**title  line-height**: 18**px  font-size**: 12**px  color**: **rgb**(7, 17, 27)  .**delivery  margin-left**: 12**px  font-size**: 12**px  color**: **rgb**(147, 153, 159)  .**rating-wrapper  padding**: 0 18**px** .**rating-item  display**: **flex  padding**: 18**px** 0  **border-1px**(**rgba**(7, 17, 27, 0.1))  .**avatar  flex**: 0 0 28**px  width**: 28**px  margin-right**: 12**px  img  border-radius**: 50%  .**content  position**: **relative  flex**: 1  .**name  margin-bottom**: 4**px  line-height**: 12**px  font-size**: 10**px  color**: **rgb**(7, 17, 27)  .**star-wrapper  margin-bottom**: 6**px  font-size**: 0  .**stars  display**: **inline-block  margin-right**: 6**px  vertical-align**: **top** .**delivery  display**: **inline-block  vertical-align**: **top  line-height**: 12**px  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**text  margin-bottom**: 8**px  line-height**: 18**px  color**: **rgb**(7, 17, 27)  **font-size**: 12**px** .**recommend  line-height**: 16**px  font-size**: 0  .**icon-thumb\_up**, .**icon-thumb\_down**, .**item  display**: **inline-block  margin**: 0 8**px** 4**px** 0  **font-size**: 9**px** .**icon-thumb\_up  color**: **rgb**(0, 160, 220)  .**icon-thumb\_down  color**: **rgb**(147, 153, 159)   .**item  padding**: 0 6**px  border**: 1**px solid rgba**(7, 17, 27, 0.1)  **border-radius**: 1**px  color**: **rgb**(147, 153, 159)  **background**: **#fff** .**time  position**: **absolute  top**: 0  **right**: 0  **line-height**: 12**px  font-size**: 10**px  color**: **rgb**(147, 153, 159) </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div class="ratings" ref="ratings"**>  <**div class="ratings-content"**>  <**scroller**>  <**div class="overview"**>  <**div class="overview-left"**>  <**h1 class="score"**>{{*seller*.**score**}}</**h1**>  <**div class="title"**>综合评分</**div**>  <**div class="rank"**>高于周边商家{{*seller*.**rankRate**}}%</**div**>  </**div**>  <**div class="overview-right"**>  <**div class="score-wrapper"**>  <**span class="title"**>服务态度</**span**>  <**star :score="***seller*.**serviceScore" :size="**36**"**></**star**>  <**span class="score"**>{{*seller*.**serviceScore**}}</**span**>  </**div**>  <**div class="score-wrapper"**>  <**span class="title"**>商品评分</**span**>  <**star :score="***seller*.**foodScore" :size="**36**"**></**star**>  <**span class="score"**>{{*seller*.**foodScore**}}</**span**>  </**div**>  <**div class="delivery-wrapper"**>  <**span class="title"**>送达时间</**span**>  <**span class="delivery"**>{{*seller*.**deliveryTime**}}分钟</**span**>  </**div**>  </**div**>  </**div**>   <**split**></**split**>   <**ratingselect  :ratings="*ratings*"  :selectType="selectType"  :onlyContent="onlyContent"  :desc="**{**all**: **'全部'**, **positive**:**'满意'**, **negative**: **'不满意'**}**"  :toggleOnlyContent="toggleOnlyContent"  :setSelectType="setSelectType"**></**ratingselect**>   <**div class="rating-wrapper"**>  <**ul**>  <**li class="rating-item" v-for="*rating*** in filterRatings**"**>  <**div class="avatar"**>  <**img width="28" height="28" :src="*rating***.**avatar"**>  </**div**>  <**div class="content"**>  <**h1 class="name"**>{{***rating***.**username**}}</**h1**>  <**div class="star-wrapper"**>  <**star :score="*rating***.**score" :size="**24**"**></**star**>  <**span class="delivery"**>{{***rating***.**deliveryTime**}}</**span**>  </**div**>  <**p class="text"**>{{***rating***.**text**}}</**p**>  <**div class="recommend"**>  <**span :class="*rating***.**rateType**===0?**'icon-thumb\_up'**:**'icon-thumb\_down'"**></**span**>  <**span class="item" v-for="*item*** in ***rating***.**recommend"**>{{***item***}}</**span**>  </**div**>  <**div class="time"**>  {{***rating***.**rateTime** | date-**string**}}  </**div**>  </**div**>  </**li**>  </**ul**>  </**div**>  </**scroller**>  </**div**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **import** {mapState} **from 'vuex'  import** BScroll **from 'better-scroll'  import** star **from 'components/star/star.vue'  import** split **from 'components/split/split.vue'  import** ratingselect **from 'components/ratingselect/ratingselect.vue'   export default** {   data () {  **return** {  **selectType**: 0,  **onlyContent**: **false** }  },   mounted () {  **this**.$store.**dispatch**(**'getRatings'**, **function** () {  */\*this.$nextTick(function () {  this.scroll = new BScroll(this.$refs.ratings, {  click: true  })  })\*/* })  },   **methods**: {  setSelectType (selectType) {  **this**.**selectType** = selectType  */\*this.$nextTick(() => {  this.scroll.refresh()  })\*/* },   toggleOnlyContent () {  **this**.**onlyContent** = !**this**.**onlyContent** */\*this.$nextTick(() => {  this.scroll.refresh()  })\*/* }  },   **computed**: {  ...mapState([**'seller'**, **'ratings'**]),  filterRatings () {  **const** ratings = **this**.**ratings  if**(!ratings) {  **return** []  }  **const** {selectType, onlyContent} = **this  return** ratings.filter(rating => {  **const** {rateType, text} = rating  */\*  selectType: 2, //全部 // rating.rateType(0/1)  onlyContent: true // 是否只看有内容的 //rating.text  \*/  /\*  selectType: 0/1/2 如果是0/1需要判断是否与rating.rateType相等, 如果是2就不需要  onlyContent: true/false 如果为true需要判断rating.text必须有值, 如果是false就不需要  \*/* **if**(selectType===2) {  **return** !onlyContent || rating.**text**.**length**>0  } **else** {  **return** selectType===rateType && (!onlyContent || rating.**text**.**length**>0)  }  })  }  },   **components**: {  star,  split,  ratingselect  }  } </**script**> |

## 开发seller组件

### 图片资源

|  |
| --- |
| decrease\_4@3x.png  invoice\_4@3x.png  guarantee\_4@2x.png  discount\_4@2x.png  guarantee\_4@3x.png  decrease\_4@2x.png  special\_4@3x.png  invoice\_4@2x.png  discount\_4@3x.png  special\_4@2x.png |

### stylus样式

|  |
| --- |
| <**style lang="stylus" rel="stylesheet/stylus"**>  **@import "../../common/stylus/mixin.styl"** .**seller  position**: **absolute  top**: 174**px  bottom**: 0  **left**: 0  **width**: 100%  **overflow**: **hidden** .**overview  position**: **relative  padding**: 18**px** .**title  margin-bottom**: 8**px  line-height**: 14**px  color**: **rgb**(7, 17, 27)  **font-size**: 14**px** .**desc  padding-bottom**: 18**px  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 0  .**star  display**: **inline-block  margin-right**: 8**px  vertical-align**: **top** .**text  display**: **inline-block  margin-right**: 12**px  line-height**: 18**px  vertical-align**: **top  font-size**: 10**px  color**: **rgb**(77, 85, 93)  .**remark  display**: **flex  padding-top**: 18**px** .**block  flex**: 1  **text-align**: **center  border-right**: 1**px solid rgba**(7, 17, 27, 0.1)  **&**:**last-child  border**: **none  h2  margin-bottom**: 4**px  line-height**: 10**px  font-size**: 10**px  color**: **rgb**(147, 153, 159)  .**content  line-height**: 24**px  font-size**: 10**px  color**: **rgb**(7, 17, 27)  .**stress  font-size**: 24**px** .**favorite  position**: **absolute  width**: 50**px  right**: 11**px  top**: 18**px  text-align**: **center** .**icon-favorite  display**: **block  margin-bottom**: 4**px  line-height**: 24**px  font-size**: 24**px  color**: **#d4d6d9  &**.**active  color**: **rgb**(240, 20, 20)  .**text  line-height**: 10**px  font-size**: 10**px  color**: **rgb**(77, 85, 93)  .**bulletin  padding**: 18**px** 18**px** 0 18**px** .**title  margin-bottom**: 8**px  line-height**: 14**px  color**: **rgb**(7, 17, 27)  **font-size**: 14**px** .**content-wrapper  padding**: 0 12**px** 16**px** 12**px  border-1px**(**rgba**(7, 17, 27, 0.1))  .**content  line-height**: 24**px  font-size**: 12**px  color**: **rgb**(240, 20, 20)  .**supports** .**support-item  padding**: 16**px** 12**px  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 0  **&**:**last-child  border-none**()  .**icon  display**: **inline-block  width**: 16**px  height**: 16**px  vertical-align**: **top  margin-right**: 6**px  background-size**: 16**px** 16**px  background-repeat**: **no-repeat  &**.**decrease  bg-image**(**'decrease\_4'**)  **&**.**discount  bg-image**(**'discount\_4'**)  **&**.**guarantee  bg-image**(**'guarantee\_4'**)  **&**.**invoice  bg-image**(**'invoice\_4'**)  **&**.**special  bg-image**(**'special\_4'**)  .**text  line-height**: 16**px  font-size**: 12**px  color**: **rgb**(7, 17, 27)  .**pics  padding**: 18**px** .**title  margin-bottom**: 12**px  line-height**: 14**px  color**: **rgb**(7, 17, 27)  **font-size**: 14**px** .**pic-wrapper  width**: 100%  **overflow**: **hidden  white-space**: **nowrap** .**pic-list  font-size**: 0  .**pic-item  display**: **inline-block  margin-right**: 6**px  width**: 120**px  height**: 90**px  &**:**last-child  margin**: 0  .**info  padding**: 18**px** 18**px** 0 18**px  color**: **rgb**(7, 17, 27)  .**title  padding-bottom**: 12**px  line-height**: 14**px  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 14**px** .**info-item  padding**: 16**px** 12**px  line-height**: 16**px  border-1px**(**rgba**(7, 17, 27, 0.1))  **font-size**: 12**px  &**:**last-child  border-none**() </**style**> |

### 页面模板

|  |
| --- |
| <**template**>  <**div class="seller" ref="seller"**>  <**div class="seller-content"**>  <**div class="overview"**>  <**h1 class="title"**>{{*seller*.name}}</**h1**>  <**div class="desc border-1px"**>  <**star :score="***seller*.**score" :size="**36**"**></**star**>  <**span class="text"**>({{*seller*.**sellCount**}})</**span**>  <**span class="text"**>月售单</**span**>  </**div**>  <**ul class="remark"**>  <**li class="block"**><**h2**>起送价</**h2**>  <**div class="content"**>  <**span class="stress"**>{{*seller*.**minPrice**}}</**span**>元  </**div**>  </**li**>  <**li class="block"**><**h2**>商家配送</**h2**>  <**div class="content"**>  <**span class="stress"**>{{*seller*.**deliveryPrice**}}</**span**>元  </**div**>  </**li**>  <**li class="block"**><**h2**>平均配送时间</**h2**>  <**div class="content"**>  <**span class="stress"**>{{*seller*.**deliveryTime**}}</**span**>分钟  </**div**>  </**li**>  </**ul**>  <**div class="favorite" @click="**toggleFavorite**"**>  <**span class="icon-favorite" :class="**{**active**: **favorite**}**"**></**span**>  <**span class="text"**>收藏</**span**>  </**div**>  </**div**>   <**split**></**split**>   <**div class="bulletin"**>  <**h1 class="title"**>公告与活动</**h1**>  <**div class="content-wrapper border-1px"**>  <**p class="content"**>{{*seller*.**bulletin**}}</**p**>  </**div**>  <**ul class="supports"**>  <**li class="support-item border-1px" v-for="**support in *seller*.supports**"**>  <**span class="icon" :class="supportClasses**[support.**type**]**"**></**span**>  <**span class="text"**>{{support.**description**}}</**span**>  </**li**>  </**ul**>  </**div**>   <**split**></**split**>   <**div class="pics"**>  <**h1 class="title"**>商家实景</**h1**>  <**div class="pic-wrapper" ref="picWrapper"**>  <**ul class="pic-list" ref="ul"**>  <**li class="pic-item" v-for="**pic in *seller*.**pics"**>  <**img width="120" height="90" :src="**pic**"**>  </**li**>  </**ul**>  </**div**>  </**div**>   <**split**></**split**>   <**div class="info"**>  <**h1 class="title border-1px"**>商家信息</**h1**>  <**ul**>  <**li class="info-item" v-for="*info*** in *seller*.**infos"**>{{***info***}}</**li**>  </**ul**>  </**div**>  </**div**>  </**div**> </**template**> |

### JS

|  |
| --- |
| <**script**>  **import** {mapState} **from 'vuex'  import** BScroll **from 'better-scroll'  import** star **from 'components/star/star.vue'  import** split **from 'components/split/split.vue'   export default** {  data () {  **return** {  **favorite**: ***localStorage***.getItem(**'favorite'**) === **'true'**,  **supportClasses**: [**'decrease'**, **'discount'**, **'special'**, **'invoice'**, **'guarantee'**]  }  },   mounted () {   *// 如果seller还没有数据不能显示seller* **if**(!**this**.**seller**.**pics**) {  **return** }   **this**.\_initScroll() *// 从其它路由切换来过* },   **methods**: {  toggleFavorite () {  **this**.**favorite** = !**this**.**favorite** *// 保存到local* ***localStorage***.setItem(**'favorite'**, **this**.**favorite**) *// 'true'/'false'* },   \_initScroll () {  **this**.$nextTick(() => {  **new** BScroll(**this**.**$refs**.seller, {  **click**: **true** })   *// 给ul设置样式宽度* **const** ul = **this**.**$refs**.ul  **const** liWidth = 120  **const** space = 6  **const** liSize = **this**.**seller**.**pics**.length  ul.**style**.**width** = (liWidth+space)\*liSize-space + **'px'   new** BScroll(**this**.**$refs**.picWrapper, {  **click**: **true**,  **scrollX**: **true** *// 水平滚动* })  })  }  },   **computed**: mapState([**'seller'**]),   **watch**: { *// 监视seller的变化* seller (newSeller, oldSeller) { *// 在当前路由刷新* **this**.\_initScroll()  }  },  **components**: {  star,  split  }  } </**script**> |

## 购物车小球动画

### 模板页面

|  |
| --- |
| <**div class="ball-container"**>  <**transition @before-enter="beforeDrop"  @enter="dropping"  @after-enter="afterDrop"  :css="false"  v-for="(ball,index) in balls"  :key="index"**>  <**div class="ball" v-show="ball.isShow"**>  <**div class="inner inner-hook"**></**div**>  </**div**>  </**transition**> </**div**> |

### JS

|  |
| --- |
| <**script**> **export default** {  data () {  **return** {  **balls**: [  {**isShow**: **false**},  {**isShow**: **false**},  {**isShow**: **false**},  {**isShow**: **false**},  {**isShow**: **false**}  ],  **droppingBalls**: [] *// 保存多个待执行动画的ball* }  },   **methods**: {  *//启动一个小球动画* drop(startEl) {  *// 找到一个隐藏的小球元素, 让它显示出来, 并动画  // 从balls中找出isshow为falSE的ball* **const** ball = **this**.balls.**find**(ball =>!ball.**isShow**)  *// 只有找到, 才做动画* **if**(ball) {  ball.**isShow** = **true** ball.**startEl** = startEl *// 保存对应的起始位置的元素* **this**.droppingBalls.push(ball) *// 把启动动画的ball保存起来* }  },   *// 动画开始之前: 将小球瞬间移动到动画的起始位置(点击位置)* beforeDrop(el) {  **console**.log(**'beforeDrop()'**)  **var** offsetX = 0  **var** offsetY = 0   *// 取出第一个待启动动画的ball* **const** ball = **this**.droppingBalls.shift()  *// 找到点击的startEl* **var** startEl = ball.**startEl** *// 得到起始位置的坐标* **const** {left, top} = startEl.getBoundingClientRect()  *// 得劲原始位置的坐标* **const** elLeft = 32  **const** elBottom = 22  *// 算出偏移量* offsetX = left-elLeft  offsetY = -(**window**.**innerHeight**-top-elBottom)   *// 瞬间移动动画起始的位置* el.**style**.**transform** = **`translate3d(0,** ${offsetY}**px, 0)`** el.**style**.**webkitTransform** = **`translate3d(0,** ${offsetY}**px, 0)`  const** innerEl = el.**children**[0]  innerEl.**style**.**transform** = **`translate3d(**${offsetX}**px, 0, 0)`** innerEl.**style**.**webkitTransform** = **`translate3d(**${offsetX}**px, 0, 0)`** *// 保存ball* el.**ball** = ball  },   *// 动画开始时: 指定动画结束的位置* dropping(el) {  **console**.log(**'dropping()'**)  *//强制页面重绘(否则动画瞬间完成, 没有效果)* **const** rf = el.**offsetHeight  this**.$nextTick(() => {  el.**style**.**transform** = **`translate3d(0, 0, 0)`** el.**style**.**webkitTransform** = **`translate3d(0, 0, 0)`  const** innerEl = el.**children**[0]  innerEl.**style**.**transform** = **`translate3d(0, 0, 0)`** innerEl.**style**.**webkitTransform** = **`translate3d(0, 0, 0)`** })  },   *// 动画完成之后: 做一些收尾的工作  // 问题: 在动画生命周期回调函数中更新状态, 页面不变化 2.0没有* afterDrop(el) {  **console**.log(**'afterDrop()'**)  *// 找到对应的ball* **const** ball = el.**ball** ball.**startEl** = **null** *//el不会主动消失, 通过原生dom让el消失  // el.style.display = 'none'* setTimeout(() => { *// 延迟到动画结束时才去隐藏ball* ball.**isShow** = **false** }, 400)  }  }  } </**script**> |

## 项目优化/扩展

### 缓存路由组件对象

|  |
| --- |
| <**keep-alive**>  <**router-view** /> </**keep-alive**> |

* 好处: 复用路由组件对象, 复用路由组件获取的后台数据
* 是什么：vue-router里面的，是全局组件，到处可写，当前路由组件在vue开发工具里是深色的
* 做什么的：多个路由组件切换显示，可以使用浏览器的内存缓存，不需要刷新页面，效率高，如果不加是从后台获取的新数据，效率低
* 什么时候用：当路由组件切换显示时，实时性要求低时可以使用
* 想要最新的数据怎么办：直接刷新或从别的组建切回来使用缓存，在当前组建点击当前组建时刷新
* 缓存：共三种 浏览器端（针对个人的，分两种 内存：刷新后消失、硬盘：一直存在的）服务器端(针对所有用户)

### 路由组件懒加载

* src/router/index.js

|  |
| --- |
| */\* import goods from 'pages/goods/goods.vue' import ratings from 'pages/ratings/ratings.vue' import seller from 'pages/seller/seller.vue' \*/*  //import函数运行时动态加载模块包**const** *goods* = () => **import**(**'pages/goods/goods.vue'**) **const** *ratings* = () => **import**(**'pages/ratings/ratings.vue'**) **const** *seller* = () => **import**(**'pages/seller/seller.vue'**) |

是什么：开发环境下路由组件的代码在服务器的内存中，浏览器没有获取，路由组件的包在需要时才从后台获取，前端延迟向服务器获取加载需要的代码数据

做什么的：定义函数，函数执行时运行组件，起到拆解打包app.js的作用，也就是说只加载一部分js

什么时候用：比如首屏的显示加载东西多，导致效率低时间慢，通过拆分提高运行效率，让用户体验更好 一次显示的js代码比整体打包的小很多时可以考虑使用

生产时与开发时js打包问题：开发环境下js打成一个包app.js，开发只是为了调试 生产环境下js打成3个包，app.js（我们写的代码） manifest.js(清单webpack定义的语法代码，固定模板代码2k左右) vender.js（引入第三方包打包生成的代码）还有3个相应的map文件

### 图片懒加载: vue-lazyload

* Github: <https://github.com/hilongjw/vue-lazyload>
* main.js
* 下载包

|  |
| --- |
| npm install --save vue-loader |

* 使用

|  |
| --- |
| **import** VueLazyload **from 'vue-lazyload'**  **// 先将组件放在指定文件夹**  **import** loading **from './common/imgs/loading.gif'**  **//内部会定义一个全局的指令v-lazy**  Vue.use(VueLazyload, {  loading })  // 在组件中使用  <**img v-lazy="food**.image**"**> |

是什么:图片从后台动态获取，图片没加载出来之前，搞一个图片显示

做什么的:懒加载图片

什么时候用:图片没加载出来之前，需要显示点什么时，提高用户体验

### 使用vue UI组件库: mint-ui

* 主页: <http://mint-ui.github.io/#!/zh-cn>
* 下载:

|  |
| --- |
| npm install --save mint-ui |

* 实现按需打包:减小项目体积

|  |
| --- |
| 1. 下载  npm install --save-dev [babel-plugin-component](https://github.com/QingWei-Li/babel-plugin-component)   1. 修改babel配置   **// transform-runtime是个第三方包，abel-plugin-transform-runtime**  **"plugins"**: [**"transform-runtime"**,[**"component"**, [  {  // 按需加载只针对mint-ui，自动加载对应的样式  **"libraryName"**: **"mint-ui"**,  **"style"**: **true** } ]]] |

* mint-ui组件分类

1. 标签组件
2. 非标签组件

* 使用mint-ui的组件

|  |
| --- |
| **import** {Button} **from 'mint-ui'**  *// 注册全局组件* // button.name为mt-button，在mint-ui/button/index.js中有设置  Vue.component(Button.**name**, Button)  <**mt-button type="primary" @click.native="clearCart"**>清空</**mt-button**>  // ui组件分两种：标签组件，函数组件，标签组件需要注册成标签，函数组件执行函数出效果  // MessageBox 就是函数组件，哪里需要在哪里引入，不用在main.Js  **import** { MessageBox, Toast} **from 'mint-ui'**;  // Toast是简短的消息提示框，支持自定义位置、持续时间和样式  Toast({  **message**: **'清除成功'**,  **position**: **'middle'**,  **duration**: 2000 });  clearCart () {  // 用then返回两个操作，一个确定，一个取消  MessageBox.confirm('确定清空购物车吗?').then(  action => {  this.$store.dispatch('clearCart', this.foods)  Toast({  **message: '清除成功啦!!!',**  **position: 'middle',**  **duration: 1000**  });  },  () => {  console.log('点击了取消')  }  );  } |

### 新版vue脚手架项目

新老版本浏览器两个差别：

1.老版本npm start能直接打开浏览器，新版本默认不能，需要做配置 config下的index里面修改autoOpenBrowser: true,默认为false

2. 老版本的build里面有dev-server这个文件，使用express启动服务，在配置中可以使用express模拟数据

新版本的build里面没有dev-server这个文件，配置中没有直接使用express启动服务, 而是利用webpack-dev-server启动的服务，不能使用express模拟数据，如果自己弄一个dev-server，还要监听一个新的端口，也不应该在当前应用弄

问题:

配置中没有直接使用express启动服务, 而是利用webpack-dev-server启动的服务

老版本在配置中可以使用express模拟数据，这种方式在新版就不可用了

解决:

单独创建一个node+express应用(相当于服务器), 在此提供提供数据接口

webpack创建node工程：file—new—project—Node.js Express App 起个名字server模板用ejs npm i下在依赖

前后台分离后台提供数据，数据放在服务器端，通过express的路由暴露出来，express默认端口号3000，express的页面写在public中的html中

Api文档：向后台提供参数两种方式get和post，get的参数类型有两类pramas（message/2）prama映射路由path怎么确定（message/:id id作用：占位和标识名称）和query（message？id=2）

Get请求可以用JsonP解决，post请求要改服务器端

开发时候经常会出现跨域问题，尤其后台接口已经写好了，前端在本地服务器，后台是真实的服务器，但在生产环境下一般同域

问题:

开发中ajax请求存在跨域请求的问题

解决:

Webpack中配置代理(代理后台转发请求)

假设前端请求8082/api/seller，代理服务器监听端口号也是8082，这样就不跨域了，再去转发3000的后台地址，并修改地址，返回数据代理服务器也不使用，返回给前端

在Config—index—dev里面，添加下面内容，修改配置之后要重启前端项目

|  |
| --- |
| **proxyTable**: {  **'/api'**: { *// 匹配所有以 '/api'开头的请求路径* **target**: **'http://localhost:3000'**, *// 代理目标的基础路径* **changeOrigin**: **true**, *// 支持跨域* **pathRewrite**: {*// 重写路径: 去掉路径中开头的'/api'* **'^/api'**: **''** }  } } |

### Vue-scroller

* Github:[https://github.com/wangdahoo/vue-scroller](https://github.com/wangdahoo/vue-scroller%20)
* main.js
* 下载包

|  |
| --- |
| npm i vue-scroller --save |

* 使用

|  |
| --- |
| **import** VueScroller **from 'vue-scroller'**  **//内部会定义一个全局的组件标签<scroller>**  Vue.use(VueScroller)  // 在组件中使用  **<scroller>**  **// 顶部刷新效果，下拉产生效果 后面是监听回掉函数**  **:on-refresh=”refresh”**  **// 底部刷新效果，上滑加载下一页内容产生效果 如果加这两个就有相应效果，不加就没有相应效果，// 很智能**  **:on-infinite=”infinite”**  **// 滑动的内容写在此处，是由scroller包裹多个div**  **</scroller>**  想使用**scroller提供的例子，需要去到git上面demo中查看，** |

是什么:一个vue插件库，有缺陷只能垂直滑动，不能水平滑动

做什么的:滑动

什么时候用:vue组件需要滑动的时候使用，需要刷新的loading效果或者单纯的垂直滑动都是可以的

### cube-ui

* Github:[https://github.com/didi/cube-ui/](https://github.com/didi/cube-ui/%20)
* Github:[https://didi.github.io/cube-ui/#/zh-CN](https://didi.github.io/cube-ui/#/zh-CN )
* main.js
* 滴滴出的基于Vue.js实现的精致移动端组件库
* 使用方式看示例学习

### iview

针对PC端的ui组件库，个人写的

### element-ui

针对PC端的ui组件库，饿了么做的，饿了么共做了两套，另一套用于移动端mint-ui

### ui组件库介绍

真正做项目上的话用ui组件就整体用，包括页面构造布局和功能效果都是由组件搞定，不是我们自己去写，减少样式代码量，写项目更快捷，做个基本修饰改造改造就可以，套路无非就是写标签，传一些属性

ui组件库包含两个方面，一个css提供效果，一个js提供功能

ui组件分两种：标签组件，函数组件，标签组件需要注册成标签，函数组件执行函数出效果，哪里需要在哪里引入，不用在main.Js