高仿”饿了吗”外卖 App

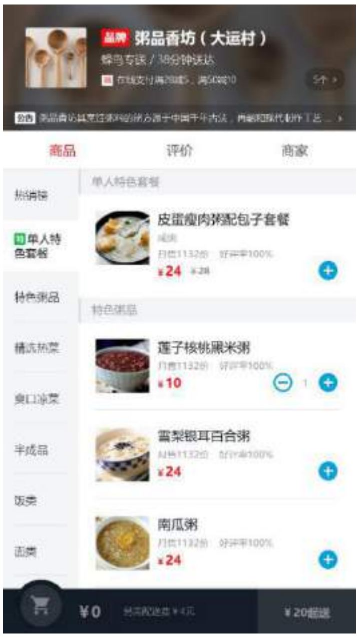
1. 准备

1.1. 项目描述

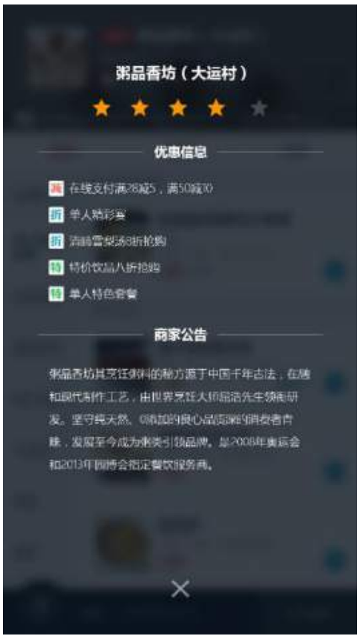
- 此项目为外卖 App 核心的**商家模块**的 **SPA**
- 包括商品、评论、商家介绍、购物车等多个子模块
- 使用 **Vuejs+ES6+Webpack 等**前端最新最热的技术
- 采用**模块化**、**组件化**、**工程化**的模式开发

1.2. 项目功能界面

01_商品页



02_商品页_公告及优惠信息



减 在线支付满28减5，满50减10

折 单人精彩套餐

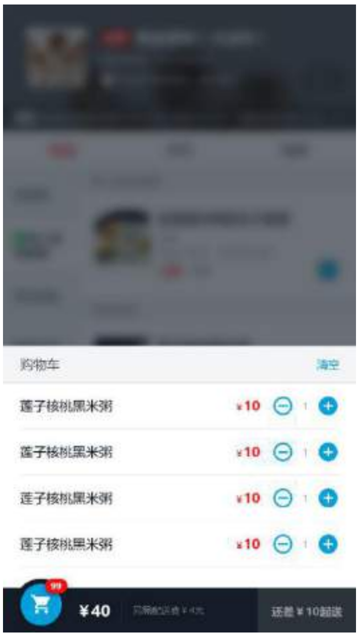
折 清甜雪梨汤8折抢购

特 特价饮品八折抢购

特 单人特色套餐

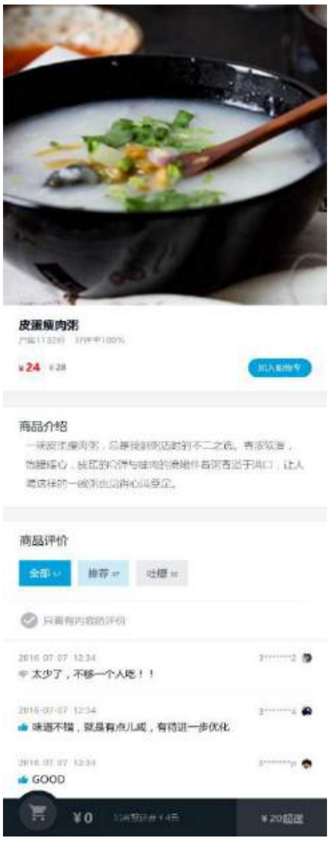
粥品香坊其烹饪粥料的秘方源于中国千年古法，在融和现代制作工艺，由世界烹饪大师屈浩先生领衔研发。坚守纯天然、0添加的良心品质深得消费者青睐，发展至今成为粥类引领品牌。是2008年奥运会和2013年园博会指定餐饮服务商。

03_商品页_购物车详情



莲子核桃黑米粥	¥10	1
莲子核桃黑米粥	¥10	1
莲子核桃黑米粥	¥10	1
莲子核桃黑米粥	¥10	1

04_商品页_商品详情

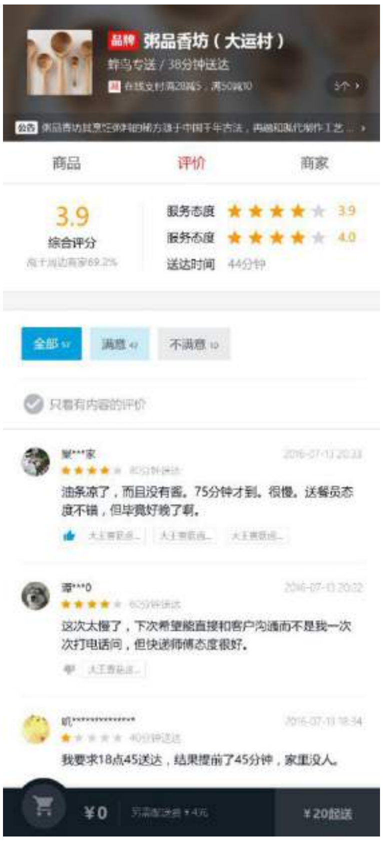


2016-07-07 12:34
太少了，不够一个人吃！！

2016-07-07 12:34
味道不错，就是有点儿咸，有待进一步优化

2016-07-07 12:34
GOOD

05_评论页

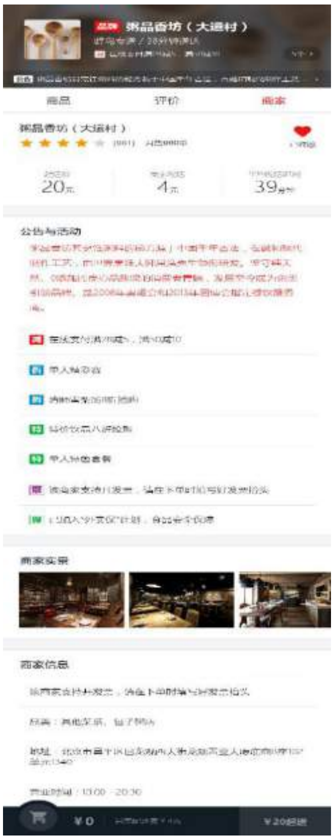


2016-07-13 20:33
油条凉了，而且没有酱。75分钟才到。很慢。送餐员态度不错，但毕竟好晚了啊。

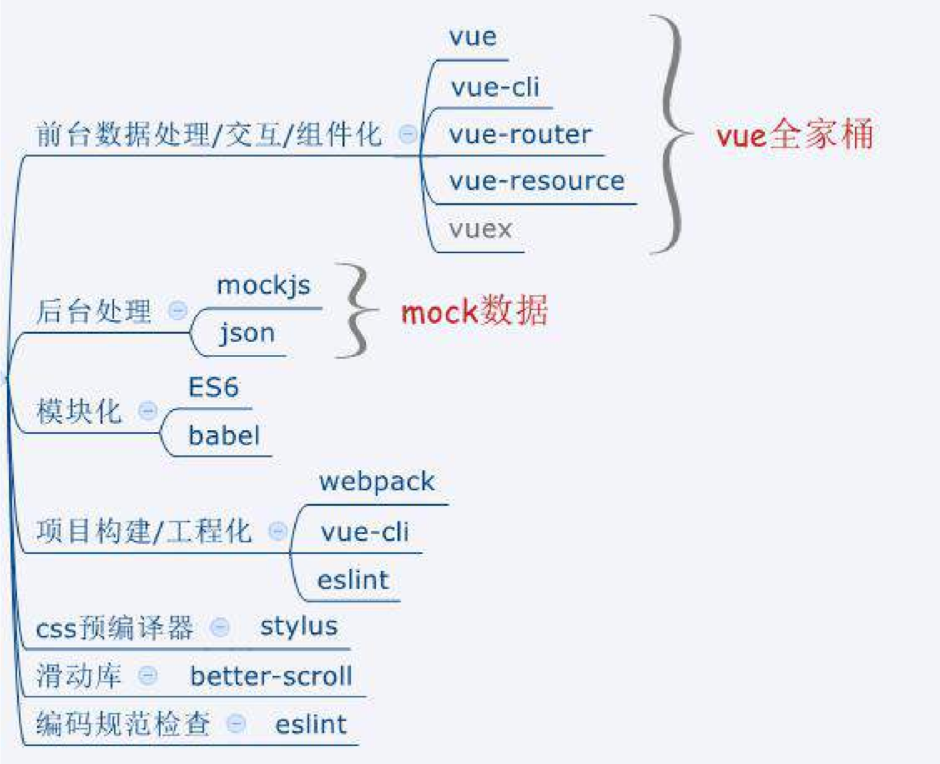
2016-07-13 20:22
这次太慢了，下次希望能直接和客户沟通而不是我一次次打电话问，但快递师傅态度很好。

2016-07-13 18:34
我要求18点45送达，结果提前了45分钟，家里没人。

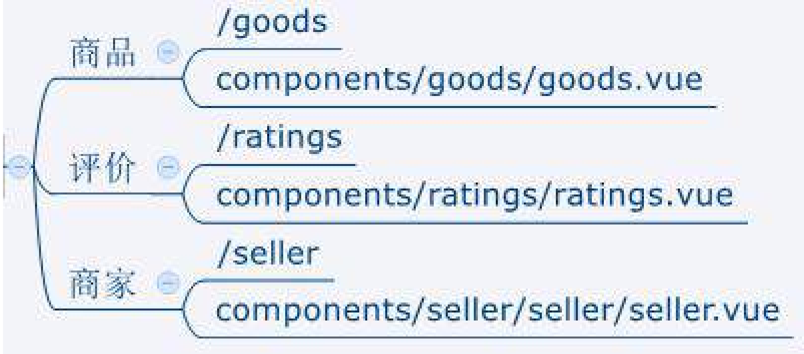
06_商家页



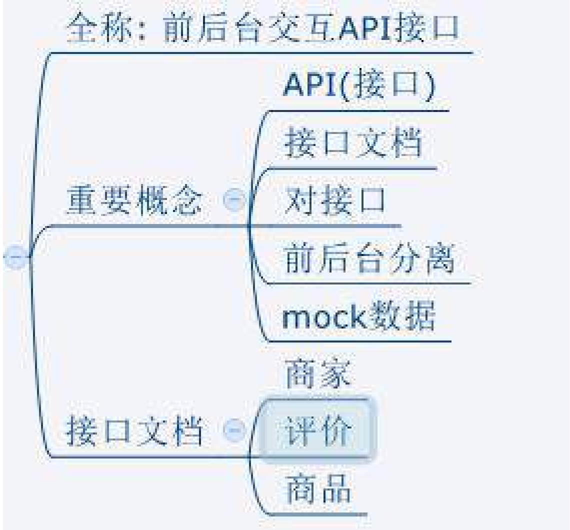
1.3. 技术选型



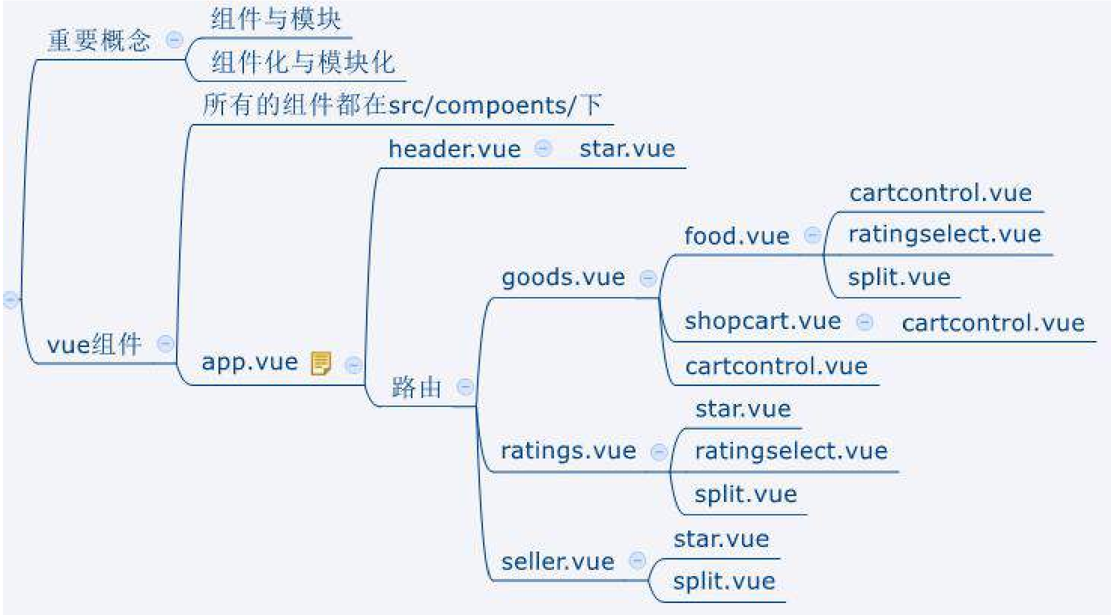
1.4. 前端路由


商品
/goods
components/goods/goods.vue
评价
/ratings
components/ratings/ratings.vue
商家
/seller
components/seller/seller/seller.vue

1.5. API 接口



1.6. 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

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

08. shopcart.vue

* 底部购物车组件: 显示购物项列表及相关操作

1.7. 你能从此项目中学到什么?

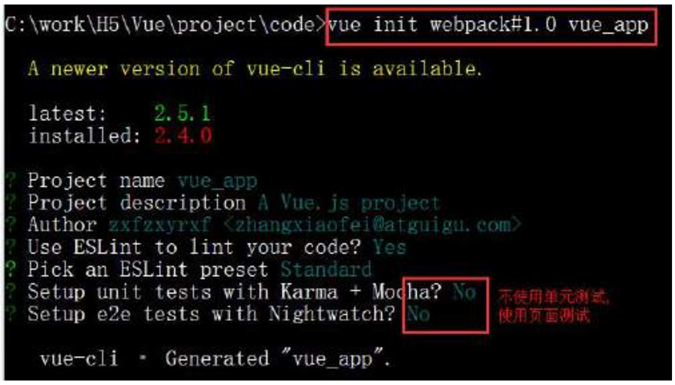
- 流程及开发方法
  - 了解一个项目的开发流程
  - 学会组件化、模块化、工程化的开发模式
  - 掌握使用 Vue-cli 脚手架初始化 Vue.js 项目
  - 学会模拟 json 后端数据，前后端分离开发
  - 学会 es6+eslint 的开发方式
- 第三方组件
  - 学会使用 stylus 编写模块化的 CSS
  - 学会使用 vue-router 开发单页应用
  - 学会使用 vue-resource 与后端数据交互
  - 学会如何在 Vue.js 框架里和第三方 JS 插件交互
- 设计思想与模式
  - 学会使用 Vue.js 的过渡编写酷炫的交互动画
  - 了解移动端设备像素比的概念
  - 学会制作并使用图标字
  - 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

2. 应用开发详解

2.1. 开启项目开发

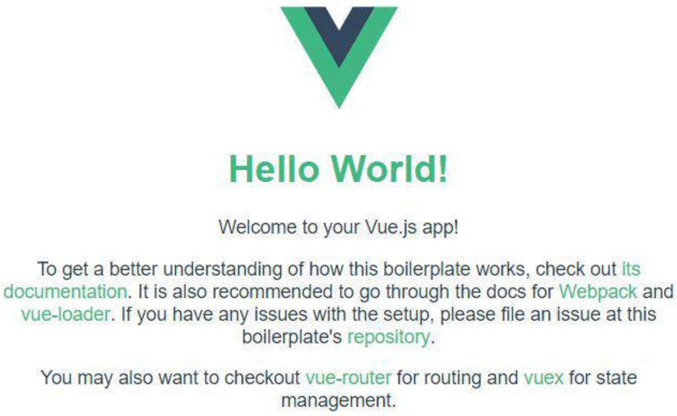
1) 使用 vue-cli(脚手架)搭建项目

- 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#1.0 vue_app :下载基于 webpack 模板项目
  - cd vue_app
  - cyarn / npm install : 下载项目依赖的所有模块
  - npm run dev
  - 访问: localhost:8080



```
C:\work\H5\Vue\project\code>vue init webpack#1.0 vue_app

  A newer version of vue-cli is available.

  latest:    2.5.1
  installed: 2.4.0

? Project name vue_app
? Project description A Vue.js project
? Author zxfzxyrxf <zhangxiaofei@atguigu.com>
? Use ESLint to lint your code? Yes
? Pick an ESLint preset Standard
? Setup unit tests with Karma + Mocha? No
? Setup e2e tests with Nightwatch? No

   vue-cli · Generated "vue_app".
```

不使用单元测试,
使用页面测试



Hello World!

Welcome to your Vue.js app!

To get a better understanding of how this boilerplate works, check out its documentation. It is also recommended to go through the docs for Webpack and vue-loader. If you have any issues with the setup, please file an issue at this boilerplate's repository.

You may also want to checkout vue-router for routing and vuex for state management.

2) 项目结构分析

```
vue_app
```

```
    |-- 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
```

3) 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 pushstate-server
  - pushstate-server dist
  - 访问: http://localhost:9000

2.2. 功能需求分析

- 演示项目功能, 对功能模块进行分析说明

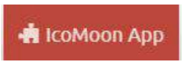
2.3. 资源准备

1) 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)

2) 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  IcoMoon App，进入处理页面
- 点击左上角 Import Icons， 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 Generate Font 生成图标字体样式
- 点击左上角 Preferences 指定 Font-Name 为 sell-icon， 点击右下角 Download 下载到本地 icomoon.zip
- 解压 zip 包，访问 demo.html 测试
- 我们项目需要的是 fonts 和 style.css

2.4. 项目源码目录设计

```
src
     |-- common----------------通用资源文件夹
          |-- fonts ---------------字体资源文件夹(图标字体文件)
          |-- js-------------------通用 js 文件夹
          |-- stylus---------------通用 stylus 文件夹
                 |-- font.styl--------图标字体样式
     |-- components------------所有 vue 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
          |-- 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
```

2.5. Stylus

1) 当前主流的三种 css 预编译器(对比)

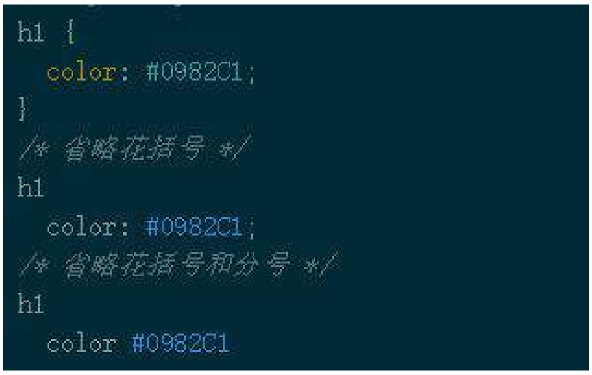
- Less
- Sass
- Stylus

2) 安装 stylus 依赖包

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

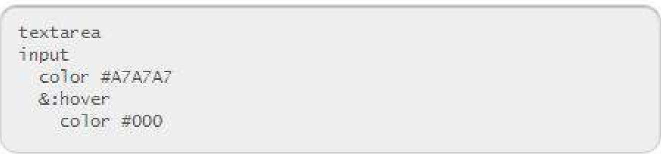
3) Stylus 的基本语法

- stylus 中文版参考文档
- 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



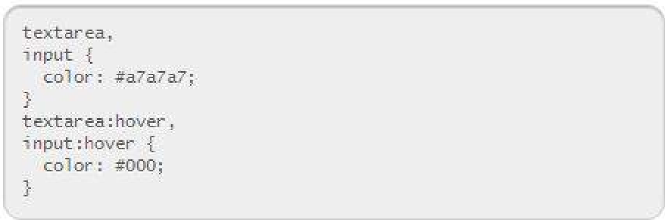
```
h1 {
  color: #0982C1;
}
/* 省略花括号 */
h1
  color: #0982C1;
/* 省略花括号和分号 */
h1
  color #0982C1
```

- 父级引用:
  - 使用字符&指向父选择器



```
textarea
input
  color #A7A7A7
  &:hover
    color #000
```

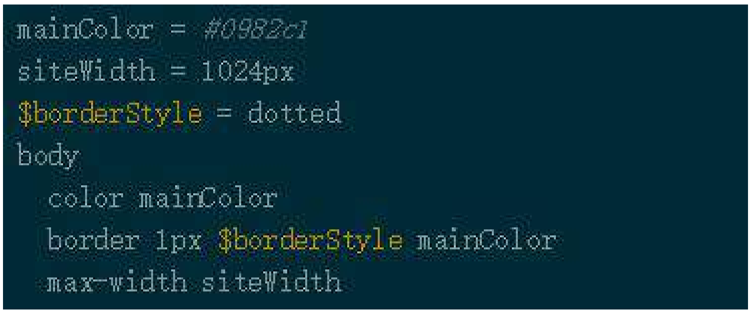
等同于：



```
textarea,
input {
  color: #a7a7a7;
}
textarea:hover,
input:hover {
  color: #000;
}
```

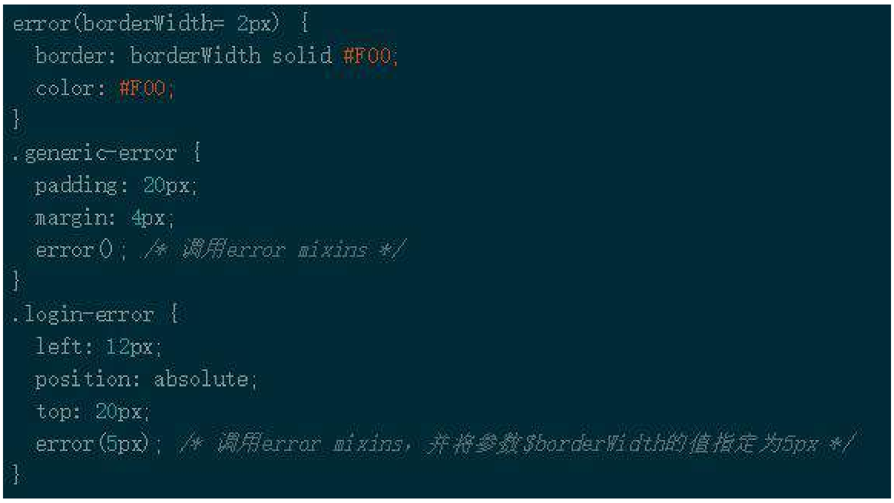
- 变量:
  - 定义变量: name=value　　如: mainColor = #0982c1

- 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)



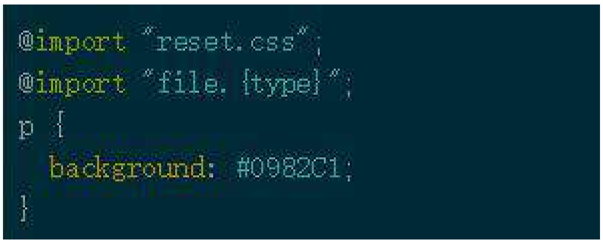
```
mainColor = #0982c1
siteWidth = 1024px
$borderStyle = dotted
body
  color mainColor
  border 1px $borderStyle mainColor
  max-width siteWidth
```

- **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



```
error(borderWidth= 2px) {
  border: borderWidth solid #F00;
  color: #F00;
}
.generic-error {
  padding: 20px;
  margin: 4px;
  error(); /* 调用error mixins */
}
.login-error {
  left: 12px;
  position: absolute;
  top: 20px;
  error(5px); /* 调用error mixins, 并将参数$borderWidth的值指定为5px */
}
```

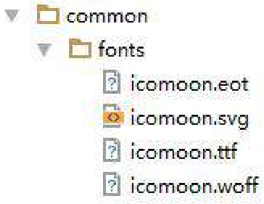
- **导入(import)**
  - 通过@import 引入其它样式文件



```
@import "reset.css";
@import "file.{type}";
p {
  background: #0982C1;
}
```

4) 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



```
▼ common
  ▼ fonts
      icomoon.eot
      icomoon.svg
      icomoon.ttf
      icomoon.woff
```

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

```
@font-face {
  font-family: 'sell-icon'
  src:  url('../fonts/sell-icon.eot?q98quz')
  src:  url('../fonts/sell-icon.eot?q98quz#iefix') format('embedded-opentype'),
    url('../fonts/sell-icon.ttf?q98quz') format('truetype'),
    url('../fonts/sell-icon.woff?q98quz') format('woff'),
    url('../fonts/sell-icon.svg?q98quz#sell-icon') 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: 'sell-icon' !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"
}
```

5) 其它基础 stylus 样式

- base.styl

```
html,body
  font-family: 'PingFang SC', 'STHeitiSC-Light', 'Helvetica-Light', arial, sans-serif

@media (min-device-pixel-ratio: 2),(-webkit-min-device-pixel-ratio: 2)
  .border-1px
    &::after
      transform:scaleY(0.5)
      -webkit-transform:scaleY(0.5)
@media (min-device-pixel-ratio: 3),(-webkit-min-device-pixel-ratio: 3)
  .border-1px
    &::after
      transform:scaleY(0.333333333333333333)
      -webkit-transform:scaleY(0.333333333333333333)

.clearfix
  zoom:1
  &:after
    content:''
    display:block
    clear:both
    height:0
    line-height:0
    visibility:hidden
```

- Mixin.styl

```
// 混合：把混合底下的样式集原封不动的搬到调用处
border-1px($color)
  position: relative
  &:after
    position: absolute
    left: 0
    bottom: 0
    content: ""
    display: block
    border-top: 1px solid $color
    width: 100%

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")
```

- index.styl

```
@import "style.styl"
@import "base.styl"
```

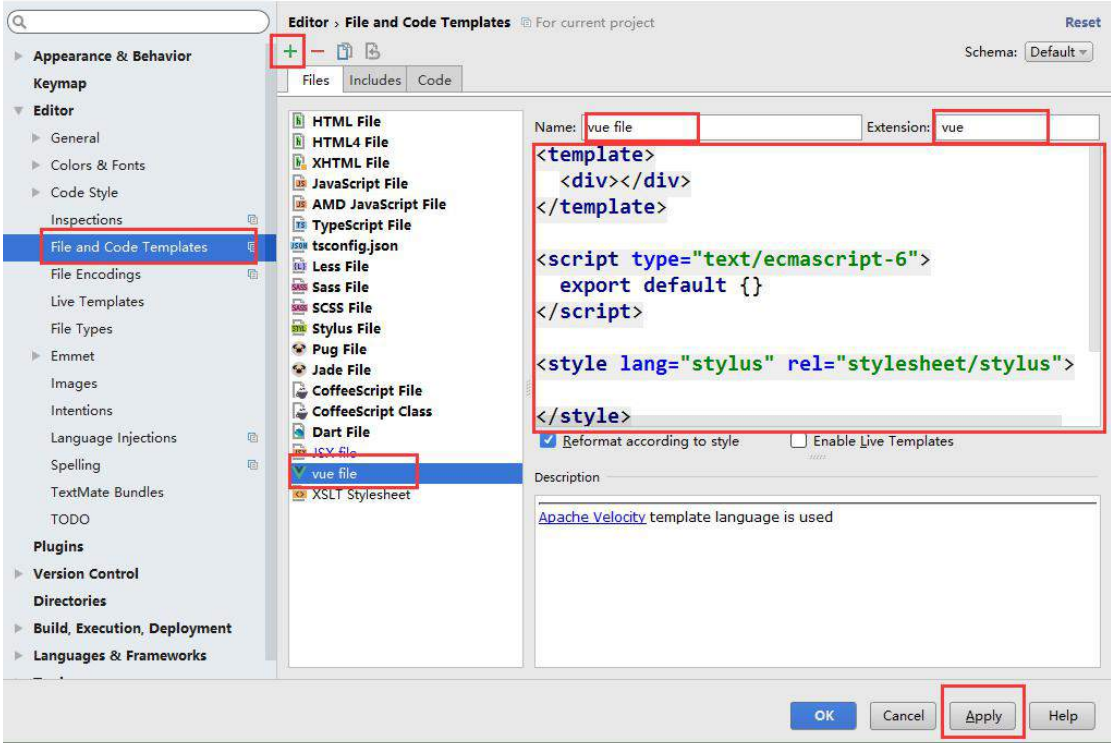
2.6. ESLint

- ESLint 是一个代码规范检查工具
- 基本已替代以前的 JSLint
- 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

2.7. Vue 组件化

1) 配置 vue 组件文件模板



Name: vue file

Extension: vue

```
<template>
  <div></div>
</template>

<script type="text/ecmascript-6">
  export default {}
</script>

<style lang="stylus" rel="stylesheet/stylus">

</style>
```

☑ Reformat according to style　□ Enable Live Templates

Description

Apache Velocity template language is used

2) 分析应用的整体 vue 组件结构

```
src
     |-- components------------所有 vue 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
          |-- 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
```

3) 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: 'body',
  components: { App }
})
```

2.8. 引入 vue-router

1) 下载 vue-router

- **npm install vue-router@0.7.13 --save**

2) 编码

- main.js

```
import Vue from 'vue'
import VueRouter from 'vue-router'

import app from './app'
import goods from './components/goods/goods'
import ratings from './components/ratings/ratings'
import seller from './components/seller/seller'

// 声明使用 vue 的扩展插件
Vue.use(VueRouter)
// 创建 router 对象
let router = new VueRouter()
// 映射路由
router.map({
  '/goods': {
    component: goods
  },
  '/ratings': {
    component: ratings
  },
  '/seller': {
    component: seller
  }
})
// 启动路由
router.start(app, '#app')
// 请求默认路由
router.go('/goods')
```

- app.vue

```
<template>
  <div>
    <!--头部组件标签-->
    <v-header></v-header>
    <!--包含 3 个路由 v-link-->
    <div class="tab">
      <div class="tab-item">
        <a v-link="{path: '/goods'}">商品</a>
      </div>
      <div class="tab-item">
        <a v-link="{path: '/ratings'}">评价</a>
      </div>
      <div class="tab-item">
        <a v-link="{path: '/seller'}">商家</a>
      </div>
    </div>
    <!--各路由 template 页面容器-->
    <router-view></router-view>
  </div>
</template>

<script>
  import header from './components/header/header'

  export default {
    // 配置组件
    components: {
      'v-header': header
    }
  }
</script>

<style lang="stylus" rel="stylesheet/stylus">
```

```
  .tab
    .tab-item
      display inline-block
      .v-link-active
        color: #ff0000
</style>
```

- index.html

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

3) 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: 5px;
    height: 5px;
}

::-webkit-scrollbar-track-piece {
    background-color: rgba(0, 0, 0, 0.2);
    -webkit-border-radius: 6px;
}

::-webkit-scrollbar-thumb:vertical {
    height: 5px;
    background-color: rgba(125, 125, 125, 0.7);
    -webkit-border-radius: 6px;
}

::-webkit-scrollbar-thumb:horizontal {
    width: 5px;
    background-color: rgba(125, 125, 125, 0.7);
    -webkit-border-radius: 6px;
}

html, body {
    width: 100%;
}

body {
    -webkit-text-size-adjust: none;
    -webkit-tap-highlight-color: rgba(0, 0, 0, 0);
}
```

- index.html 中引入

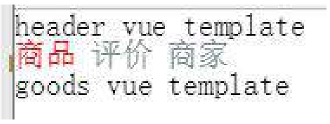
```
<link rel="stylesheet" href="static/css/reset.css">
```

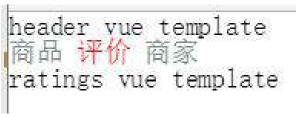
4) 添加 viewport

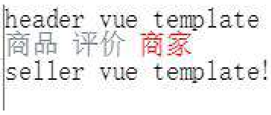
- index.html

```
<meta name="viewport"

content="width=device-width,initial-scale=1.0,maximum-scale=1.0,minimum-scale=1.0,u
ser-scalable=no">
```

5) 运行效果


header vue template
商品 评价 商家
goods vue template


header vue template
商品 评价 商家
ratings vue template


header vue template
商品 评价 商家
seller vue template!

2.9. 选项卡效果

1) 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

2) 编码

- **main.js**

```
import './common/stylus/index.styl'
```

- **app.vue**

```
<div class="tab border-1px">

<style lang="stylus" rel="stylesheet/stylus">
  @import "./common/stylus/mixin.styl"

  .tab
    display: flex
    width: 100%
    height: 40px
    line-height: 40px
    // border-bottom: 1px solid rgba(7, 17, 27, 0.1)
    border-1px(rgba(7, 17, 27, 0.1))
    .tab-item
      flex: 1
      text-align: center
      & > a
        display: block
        font-size: 14px
        color: rgb(77, 85, 93)
        &.active
          color: rgb(240, 20, 20)
</style>
```

2.10. 模拟(mock)数据

1) 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

2) 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

3) 利用 Node+express 提供模拟数据

- 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({
      errno: 0,   //0 代表正确数据
      data: seller
    })
})
apiRouter.get('/goods', function (req, res) {
  res.json({
    errno: 0,
    data: goods
  })
})
apiRouter.get('/ratings', function (req, res) {
  res.json({
    errno: 0,
    data: ratings
  })
})
```

```
//启用路由
app.use('/api', apiRouter)
```

问题：如果是打包发布就无法访问模拟数据，只能是测试时使用

4) 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

```
import apiData from './data.json'
import Mock from 'mockjs'  //npm install mockjs --save
Mock.mock('/api2/seller', {errno:0, data:apiData.seller})
Mock.mock('/api2/goods', {errno:0, data:apiData.goods})
Mock.mock('/api2/ratings', {errno:0, data:apiData.ratings})
```

好处：打包发布运行模拟接口也是可以用的，这样便于大家演示

5) 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

6) 引入 vue-resource 访问 mock 数据接口

- vue-resource 是 vue 项目中 ajax 请求库
- 官网: https://github.com/pagekit/vue-resource
- 下载: **npm install vue-resource --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('/api/goods').then((response) => {
  console.log(response.body)
}, (response) => {
  console.log('失败了')
})
```

2.11. 开发 header 组件

1) 在 app.vue 中请求获取商家数据，并传递给 header 组件

```
<v-header :seller="seller"></v-header>

data() {
  return {
    seller: {}
  }
},

created () {
  this.$http.get('/api/seller').then((response) => {
    const result = response.body
    console.log(result)
    if(result.errno === OK) {
      this.seller = result.data
    }
  })
}
```

2) header 组件样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .header
    position:relative
    color:rgb(255,255,255)
    overflow:hidden
    background-color:rgba(7,17,27,0.5)
    .content-wrapper
      position:relative
      padding 24px 12px 18px 24px
      font-size:0
      .avatar
        display: inline-block
        vertical-align: top
      .content
        display: inline-block
        margin-left: 16px
        .title
          margin: 2px 0 8px 0
          .brand
            margin-right: 6px
            vertical-align:top
            bg-image(brand)
            display: inline-block
            width: 30px
            height: 18px
            background-size:30px 18px
            background-repeat:no-repeat
          .name
            font-size: 16px
            line-height: 18px
            font-weight: bold
        .description
          font-size: 12px
          font-weight: 200
          line-height: 12px
        .support
          margin: 10px 0 2px 0
          .icon
            margin-right:4px
            vertical-align:top
            display:inline-block
            width: 12px
            height: 12px
            background-size: 12px 12px
            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: 10px
            font-weight: 200
            line-height: 12px
```

```
    .supports_count
      position:absolute
      right: 12px
      bottom: 15px
      height: 24px
      padding: 7px 8px
      border-radius: 8px
      background-color: rgba(0,0,0,0.2)
      box-sizing:border-box
      .count
        vertical-align:top
        font-size: 10px
        line-height: 12px
        font-weight: 200
        margin-right:2px
      .icon-keyboard_arrow_right
        font-size: 10px
        line-height: 12px
        font-weight: 200
  .bulletin-wrapper
    background-color:rgba(7,17,27,0.2)
    position:relative
    padding: 0 20px 0 12px
    overflow:hidden
    text-overflow:ellipsis
    white-space:nowrap
    display:block
    .bulletin-title
      vertical-align:middle
      display:inline-block
      width: 22px
      height: 12px
      bg-image(bulletin)
      background-size:100% 100%
      background-repeat:no-repeat
    .bulletin-text
      font-size: 10px
      line-height: 28px
    .icon-keyboard_arrow_right
      position:absolute
      right:12px
      bottom:9px
      font-size: 10px
```

```
.background
  position:absolute
  left:0
  top:0
  width:100%
  height:100%
  z-index: -1
  img
    filter:blur(10px)
.detail
  position:fixed
  left:0
  top:0
  z-index:100
  width:100%
  height:100%
  background:rgba(7,17,27,0.8)
  overflow:auto
  transition: .5s
  backdrop-filter: blur(10px)
  &.fade-transition
    opacity:1
  &.fade-enter,&.fade-leave
    opacity:0
  .detail-wrapper
    display:inline-block
    width:100%
    min-height:100%
    .detail-main
      margin-top:64px
      padding-bottom:64px
      .name
        text-align:center
        font-size: 16px
        line-height: 16px
        font-weight:700
      .star-wrapper
        margin-top:16px
        text-align:center
      .title
        display:flex
        width:80%
        margin: 28px auto 24px
        .text
    text-align:center
    padding: 0 12px
  .line
    flex:1
    position:relative
    top:-6px
    border-bottom:2px solid  rgba(255, 192, 203,0.3)


.supports
  width:80%
  padding:0 12px
  margin:0 auto
  .support
    margin-bottom:12px
    &:last-child
      margin-bottom:0
    .icon
      display:inline-block
      width: 16px
      height: 16px
      background-size: 16px 16px
      background-repeat:no-repeat
      vertical-align:top
      margin-right:6px
      &.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:12px
      line-height: 12px
      font-weight:200

.content
  width:80%
  padding:0 12px
```

```
           margin: 0 auto
           font-size: 12px
           line-height: 24px
           font-weight: 200


     .detail-close
       margin-top:-64px
       clear:both
       text-align:center
       .icon-close
         display:inline-block
         width: 32px
         height: 32px
         font-size:32px
</style>
```

3) 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
```

4) 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 type="text/ecmascript-6">
  export default{
    props: {
      seller: {
        type: Object
      }
    },

    created () {
      this.classMap = ["decrease", "discount", "guarantee", "invoice", "special"]
    }
  }
</script>
```

5) 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 type="text/ecmascript-6">
  export default{
    props: {
      seller: {
        type: Object
      }
    },

    data () {
      return {
        detailShow: false
      }
    },

    created () {
      this.classMap = ["decrease", "discount", "guarantee", "invoice", "special"]
    },

    methods: {
      showDetail (isShow) {
        this.detailShow = isShow
      }
    }
  }
</script>
```

2.12. 开发 star 组件

1) 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
```

2) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .star
    font-size: 0
    .star-item
      display: inline-block
      background-repeat: no-repeat
    &.star-48
      .star-item
        width: 20px
        height: 20px
        margin-right: 22px
        background-size: 20px 20px
        &: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: 15px
        height: 15px
        margin-right: 6px
        background-size: 15px 15px
        &: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: 10px
        height: 10px
        margin-right: 3px
        background-size: 10px 10px
        &:last-child
          margin-right: 0
        &.on
          bg-image('star24_on')
        &.half
          bg-image('star24_half')
        &.off
          bg-image('star24_off')
</style>
```

3) 页面模板

```
<template>
  <div class="star" :class="'star-'+size">
    <span class="star-item" v-for="item in starClasses"
          :class="item" track-by="$index"></span>
  </div>
</template>
```

4) 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>
```

5) 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
}
```

2.13. 开发 goods 组件

1) 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

2) 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
```

3) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .goods
    display: flex
    position: absolute
    top: 174px
    bottom: 46px
    width: 100%
    overflow: hidden
    .menu-wrapper
      flex: 0 0 80px
      width: 80px
      background: #f3f5f7
      .menu-item
        display: table
        height: 54px
        width: 56px
        padding: 0 12px
        line-height: 14px
        &.current
          position: relative
          z-index: 10
          margin-top: -1px
          background: #fff
          font-weight: 700
          .text
            border-none()
        .icon
          display: inline-block
          vertical-align: top
          width: 12px
          height: 12px
          margin-right: 2px
          background-size: 12px 12px
          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: 56px
          vertical-align: middle
          border-1px(rgba(7, 17, 27, 0.1))
          font-size: 12px
    .foods-wrapper
      flex: 1
      .title
        padding-left: 14px
        height: 26px
        line-height: 26px
        border-left: 2px solid #d9dde1
        font-size: 12px
        color: rgb(147, 153, 159)
        background: #f3f5f7
      .food-item
        display: flex
        margin: 18px
        padding-bottom: 18px
        border-1px(rgba(7, 17, 27, 0.1))
```

```
        &:last-child
          border-none()
          margin-bottom: 0
        .icon
          flex: 0 0 57px
          margin-right: 10px
        .content
          flex: 1
          .name
            margin: 2px 0 8px 0
            height: 14px
            line-height: 14px
            font-size: 14px
            color: rgb(7, 17, 27)
          .desc, .extra
            line-height: 10px
            font-size: 10px
            color: rgb(147, 153, 159)
          .desc
            line-height: 12px
            margin-bottom: 8px
          .extra
            .count
              margin-right: 12px
          .price
            font-weight: 700
            line-height: 24px
            .now
              margin-right: 8px
              font-size: 14px
              color: rgb(240, 20, 20)
            .old
              text-decoration: line-through
              font-size: 10px
              color: rgb(147, 153, 159)
          .cartcontrol-wrapper
            position: absolute
            right: 0
            bottom: 12px
</style>
```

4) 页面模板

```
<template>
  <div class="goods">
    <div class="menu-wrapper" v-el:menu-wrapper>
      <ul>
        <li class="menu-item" v-for="item 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" v-el:foods-wrapper>
      <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>
</template>
```

5) JS

```
<script type="text/ecmascript-6">
  import BScroll from 'better-scroll'

  export default{
    data () {
      return {
        goods: [],
        tops: [],
        scrollY: 0,
      }
    },

    created () {
      this.classMap = ['decrease', 'discount', 'special', 'invoice', 'guarantee']
      this.$http.get('/api/goods').then((response) => {
        const result = response.body
        console.log(result)
        if (result.errno === 0) {
          this.goods = result.data
          // 将回调延迟到下次 DOM 更新循环之后执行
          this.$nextTick(() => {
            //初始化滚动
            this._initScroll()
            //初始化列表项的top 值
            this._initTops()
          })
        }
      })
    },

    methods: {
      _initScroll () {
        this.menuScroll = new BScroll(this.$els.menuWrapper, {
          click: true
        })
        this.foodsScroll = new BScroll(this.$els.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.$els.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.$els.foodsWrapper.getElementsByClassName('food-list-hook')
        const el = foodList[index]
        this.foodsScroll.scrollToElement(el, 300)
      }
    },

    computed: {
      currentIndex () {
        return this.tops.findIndex((top, index)=> {
          return this.scrollY>=top && this.scrollY<this.tops[index+1]
        })
      }
    }
  }
```

```
</script>
```

2.14. 开发 cartcontrol 组件

1) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  .cartcontrol
    font-size: 0
    .cart-decrease
      display: inline-block
      padding: 6px
      transition: all 0.4s linear
      &.move-transition
        opacity: 1
        transform: translate3d(0, 0, 0)
        .inner
          display: inline-block
          line-height: 24px
          font-size: 24px
          color: rgb(0, 160, 220)
          transition: all 0.4s linear
          transform: rotate(0)
      &.move-enter, &.move-leave
        opacity: 0
        transform: translate3d(24px, 0, 0)
        .inner
          transform: rotate(180deg)
    .cart-count
      display: inline-block
      vertical-align: top
      width: 12px
      padding-top: 6px
      line-height: 24px
      text-align: center
      font-size: 10px
      color: rgb(147, 153, 159)
    .cart-add
      display: inline-block
      padding: 6px
      line-height: 24px
      font-size: 24px
      color: rgb(0, 160, 220)
</style>
```

2) 页面模板

```
<template>
  <div class="cartcontrol">
    <div class="cart-decrease" v-show="food.count>0" 
         @click="decrease" transition="move">
      <span class="inner icon-remove_circle_outline"></span>
    </div>
    <div class="cart-count" v-show="food.count>0">{{food.count}}</div>
    <div class="cart-add icon-add_circle" @click="increase"></div>
  </div>
</template>
```

3) JS

```
<script type="text/ecmascript-6">
  import Vue from 'vue'

  export default {
    props: {
      food: {
        type: Object
      }
    },
    methods: {
      increase(event) {
        if (!event._constructed) {
          return
        }
        if (!this.food.count) {
          //给对象添加属性, 更新界面
          Vue.set(this.food, 'count', 1)
        } else {
          this.food.count++
        }
      },
      decrease(event) {
        if (!event._constructed) {
          return
        }
        if (this.food.count) {
          this.food.count--
        }
      }
    }
  }
</script>
```

4) 使用组件标签

```
<div class="cartcontrol-wrapper">
  <cartcontrol :food="food"></cartcontrol>
</div>
```

2.15. 开发 shopcart 组件

1) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .shopcart
    position: fixed
    left: 0
    bottom: 0
    z-index: 50
    width: 100%
    height: 48px
    .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: -10px
      margin: 0 12px
      padding: 6px
      width: 56px
      height: 56px
      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: 44px
          font-size: 24px
          color: #80858a
          &.highlight
            color: #fff
      .num
        position: absolute
        top: 0
        right: 0
        width: 24px
        height: 16px
        line-height: 16px
        text-align: center
        border-radius: 16px
        font-size: 9px
        font-weight: 700
        color: #fff
        background: rgb(240, 20, 20)
        box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.4)
    .price
      display: inline-block
      vertical-align: top
      margin-top: 12px
      line-height: 24px
      padding-right: 12px
```

```
        box-sizing: border-box
        border-right: 1px solid rgba(255, 255, 255, 0.1)
        font-size: 16px
        font-weight: 700
        &.highlight
          color: #fff
      .desc
        display: inline-block
        vertical-align: top
        margin: 12px 0 0 12px
        line-height: 24px
        font-size: 10px
    .content-right
      flex: 0 0 105px
      width: 105px
      .pay
        height: 48px
        line-height: 48px
        text-align: center
        font-size: 12px
        font-weight: 700
        &.not-enough
          background: #2b333b
        &.enough
          background: #00b43c
          color: #fff
  .ball-container
    .ball
      position: fixed
      left: 32px
      bottom: 22px
      z-index: 200
      &.drop-transition
        transition: all 0.4s cubic-bezier(0.49, -0.29, 0.75, 0.41)
        .inner
          width: 16px
          height: 16px
          border-radius: 50%
          background: rgb(0, 160, 220)
          transition: all 0.4s linear
  .shopcart-list
    position: absolute
    left: 0
    top: 0
    z-index: -1
    width: 100%
    &.fold-transition
      transition: all 0.5s
      transform: translate3d(0, -100%, 0)
    &.fold-enter, &.fold-leave
      transform: translate3d(0, 0, 0)
    .list-header
      height: 40px
      line-height: 40px
      padding: 0 18px
      background: #f3f5f7
      border-bottom: 1px solid rgba(7, 17, 27, 0.1)
      .title
        float: left
        font-size: 14px
        color: rgb(7, 17, 27)
      .empty
        float: right
        font-size: 12px
        color: rgb(0, 160, 220)

    .list-content
      padding: 0 18px
      max-height: 217px
      overflow: hidden
      background: #fff
      .food
        position: relative
        padding: 12px 0
        box-sizing: border-box
        border-1px(rgba(7, 17, 27, 0.1))
        .name
          line-height: 24px
          font-size: 14px
          color: rgb(7, 17, 27)
        .price
          position: absolute
          right: 90px
          bottom: 12px
          line-height: 24px
          font-size: 14px
          font-weight: 700
          color: rgb(240, 20, 20)
          .cartcontrol-wrapper
            position: absolute
            right: 0
            bottom: 6px

  .list-mask
    position: fixed
    top: 0
    left: 0
    width: 100%
    height: 100%
    z-index: 40
    backdrop-filter: blur(10px)
    &.fade-transition
      transition: all 0.5s
      opacity: 1
      background: rgba(7, 17, 27, 0.6)
    &.fade-enter, &.fade-leave
      opacity: 0
      background: rgba(7, 17, 27, 0)
</style>
```

2) 页面模板

```
<template>
  <div class="shopcart">
    <div class="content" @click="toggleList">
      <div class="content-left">
        <div class="logo-wrapper">
          <div class="logo" :class="{'highlight':totalCount>0}">
            <i class="icon-shopping_cart" :class="{'highlight':totalCount>0}"></i>
          </div>
          <div class="num" v-show="totalCount>0">{{totalCount}}</div>
        </div>
        <div class="price" :class="{'highlight':totalPrice>0}">￥{{totalPrice}}</div>
        <div class="desc">另需配送费￥{{deliveryPrice}}元</div>
      </div>
      <div class="content-right" @click.stop.prevent="pay">
        <div class="pay" :class="payClass">
          {{payDesc}}
        </div>
      </div>
    </div>
    <div class="ball-container">
      <div transition="drop" v-for="ball in balls" v-show="ball.show" class="ball">
        <div class="inner inner-hook"></div>
      </div>
    </div>
    <div class="shopcart-list" v-show="listShow" transition="fold">
      <div class="list-header">
        <h1 class="title">购物车</h1>
        <span class="empty" @click="empty">清空</span>
      </div>
      <div class="list-content" v-el:list-content>
        <ul>
          <li class="food" v-for="food in selectFoods">
            <span class="name">{{food.name}}</span>
            <div class="price">
              <span>￥{{food.price*food.count}}</span>
            </div>
            <div class="cartcontrol-wrapper">
              <cartcontrol :food="food"></cartcontrol>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>
  <div class="list-mask" @click="hideList" v-show="listShow" transition="fade"></div>
</template>
```

3) JS

```
<script type="text/ecmascript-6">
  import BScroll from 'better-scroll';
  import cartcontrol from 'components/cartcontrol/cartcontrol';

  export default {
    props: {
      selectFoods: {
        type: Array,
        default() {
          return []
        }
      },
      deliveryPrice: {
        type: Number,
        default: 0
      },
      minPrice: {
        type: Number,
        default: 0
      }
    },

    data() {
      return {
        fold: true
      };
    },

    computed: {
      totalPrice() {
        let total = 0;
        this.selectFoods.forEach((food) => {
          total += food.price * food.count;
        });
        return total;
      },
      totalCount() {
        let count = 0;
        this.selectFoods.forEach((food) => {
          count += food.count;
        });
        return count;
      },
      payDesc() {
        if (this.totalPrice === 0) {
          return `￥${this.minPrice}元起送`;
        } else if (this.totalPrice < this.minPrice) {
          let diff = this.minPrice - this.totalPrice;
          return `还差￥${diff}元起送`;
        } else {
          return '去结算';
        }
      },
      payClass() {
        if (this.totalPrice < this.minPrice) {
          return 'not-enough';
        } else {
          return 'enough';
        }
      },
      listShow() {
        if (!this.totalCount) {
          this.fold = true;
          return false;
        }
        let show = !this.fold;
        if (show) {
          this.$nextTick(() => {
            if (!this.scroll) {
              this.scroll = new BScroll(this.$els.listContent, {
                click: true
              });
            } else {
              this.scroll.refresh();
            }
          });
        }
        return show;
      }
    },

    methods: {
      toggleList() {
        if (!this.totalCount) {
          return;
        }
        this.fold = !this.fold;
      },
      hideList() {
        this.fold = true;
      },
      empty() {
        this.selectFoods.forEach((food) => {
          food.count = 0;
        });
      },
      pay() {
        if (this.totalPrice < this.minPrice) {
          return;
        }
        window.alert(`支付${this.totalPrice}元`);
      }
    },

    components: {
      cartcontrol
    }
  };
</script>
```

4) 使用组件标签

- goods.vue

```
<shopcart v-ref:shopcart
    :select-foods="selectFoods"
    :delivery-price="seller.deliveryPrice"
    :min-price="seller.minPrice"></shopcart>

import shopcart from '../shopcart/shopcart'

export default{
  props: {
    seller: {
      type: Object
    }
  },
  computed: {
    selectFoods () {
      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>
```

2.16. 开发 ratingselect 组件

1) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .ratingselect
    .rating-type
      padding: 18px 0
      margin: 0 18px
      border-1px(rgba(7, 17, 27, 0.1))
      font-size: 0
      .block
        display: inline-block
        padding: 8px 12px
        margin-right: 8px
        line-height: 16px
        border-radius: 1px
        font-size: 12px
        color: rgb(77, 85, 93)
        &.active
          color: #fff
        .count
          margin-left: 2px
          font-size: 8px
        &.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: 12px 18px
      line-height: 24px
      border-bottom: 1px 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: 4px
        font-size: 24px
      .text
        display: inline-block
        vertical-align: top
        font-size: 12px
</style>
```

2) 页面模板

```
<template>
  <div class="ratingselect">
    <div class="rating-type border-1px">
      <span class="block positive" @click="select(2,$event)"
            :class="{'active':selectType===2}">
        {{desc.all}}<span class="count">{{ratings.length}}</span></span>
      <span class="block positive" @click="select(0,$event)"
            :class="{'active':selectType===0}">
        {{desc.positive}}<span class="count">{{positives.length}}</span></span>
      <span class="block negative" @click="select(1,$event)"
            :class="{'active':selectType===1}">
        {{desc.negative}}<span class="count">{{negatives.length}}</span>
      </span>
    </div>

    <div class="switch" @click="toggleContent" :class="{'on':onlyContent}">
      <span class="icon-check_circle"></span>
      <span class="text">只看有内容的评价</span>
    </div>
  </div>
```

```
</template>
```

3) JS

```
<script>
  const POSITIVE = 0
  const NEGATIVE = 1
  const ALL = 2

  export default {
    props: {
      ratings: {
        type: Array,
        default() {
          return []
        }
      },
      selectType: {
        type: Number
      },
      onlyContent: {
        type: Boolean
      },
      desc: {
        type: Object
      }
    },
    computed: {
      positives() {
        return this.ratings.filter((rating) => {
          return rating.rateType === POSITIVE
        })
      },
      negatives() {
        return this.ratings.filter((rating) => {
          return rating.rateType === NEGATIVE
        })
      }
    },
    methods: {
      select(type, event) {
        if (!event._constructed) {
          return
        }
        this.selectType = type
        this.$dispatch('ratingtype.select', type)
      },
      toggleContent(event) {
        if (!event._constructed) {
          return
        }
        this.onlyContent = !this.onlyContent
        this.$dispatch('content.toggle', this.onlyContent)
      }
    }
  }
</script>
```

2.17. 开发 food 组件

1) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .food
    position: fixed
    left: 0
    top: 0
    bottom: 48px
    z-index: 30
    width: 100%
    background: #fff
    &.move-transition
      transition: all 0.2s linear
      transform: translate3d(0, 0, 0)
    &.move-enter, &.move-leave
      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: 10px
        left: 0
        .icon-arrow_lift
          display: block
          padding: 10px
          font-size: 20px
          color: #fff

    .content
      position: relative
      padding: 18px
      .title
        line-height: 14px
        margin-bottom: 8px
        font-size: 14px
        font-weight: 700
        color: rgb(7, 17, 27)
      .detail
        margin-bottom: 18px
        line-height: 10px
        height: 10px
        font-size: 0
        .sell-count, .rating
          font-size: 10px
          color: rgb(147, 153, 159)
        .sell-count
          margin-right: 12px
      .price
        font-weight: 700
        line-height: 24px
        .now
          margin-right: 8px
          font-size: 14px
          color: rgb(240, 20, 20)
        .old
          text-decoration: line-through
          font-size: 10px
          color: rgb(147, 153, 159)
      .cartcontrol-wrapper
        position: absolute
        right: 12px
        bottom: 12px
      .buy
        position: absolute
        right: 18px
        bottom: 18px
        z-index: 10
        height: 24px
        line-height: 24px
        padding: 0 12px
        box-sizing: border-box
        border-radius: 12px
        font-size: 10px
        color: #fff
        background: rgb(0, 160, 220)
        &.fade-transition
          transition: all 0.2s
          opacity: 1
        &.fade-enter, &.fade-leave
          opacity: 0
    .info
      padding: 18px
      .title
        line-height: 14px
        margin-bottom: 6px
        font-size: 14px
        color: rgb(7, 17, 27)
      .text
        line-height: 24px
        padding: 0 8px
        font-size: 12px
        color: rgb(77, 85, 93)
    .rating
      padding-top: 18px
      .title
        line-height: 14px
        margin-left: 18px
        font-size: 14px
        color: rgb(7, 17, 27)
      .rating-wrapper
        padding: 0 18px
        .rating-item
          position: relative
          padding: 16px 0
          border-1px(rgba(7, 17, 27, 0.1))
          .user
            position: absolute
            right: 0
            top: 16px
            line-height: 12px
            font-size: 0
            .name
              display: inline-block
              margin-right: 6px
              vertical-align: top
              font-size: 10px
              color: rgb(147, 153, 159)
            .avatar
              border-radius: 50%
          .time
            margin-bottom: 6px
            line-height: 12px
            font-size: 10px
            color: rgb(147, 153, 159)
          .text
            line-height: 16px
            font-size: 12px
            color: rgb(7, 17, 27)
            .icon-thumb_up, .icon-thumb_down
              margin-right: 4px
              line-height: 16px
              font-size: 12px
            .icon-thumb_up
              color: rgb(0, 160, 220)
            .icon-thumb_down
              color: rgb(147, 153, 159)

        .no-rating
          padding: 16px 0
          font-size: 12px
          color: rgb(147, 153, 159)
</style>
```

2) 页面模板

```
<template>
  <div class="food" v-show="showFlag" v-el:food>
    <div class="food-content">
      <div class="image-header">
        <img :src="food.image">
        <div class="back" @click="hide">
          <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"></cartcontrol>
        </div>
        <div class="buy" @click.stop.prevent="addFirst"
             v-show="!food.count || food.count===0"
             transition="fade">加入购物车
        </div>
      </div>
      <split v-show="food.info"></split>
      <div class="info" v-show="food.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 :select-type="selectType"
                      :only-content="onlyContent"
                      :desc="desc"
                      :ratings="food.ratings"></ratingselect>
        <div class="rating-wrapper">
          <ul v-show="food.ratings && food.ratings.length">
```

```
            <li class="rating-item border-1px" 
v-show="needShow(rating.rateType,rating.text)" 
                v-for="rating in food.ratings">
              <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 | formatDate}}</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">暂无评
价</div>
        </div>
      </div>
    </div>
  </div>
</template>
```

3) JS

```
<script>
  import BScroll from 'better-scroll'
  import Vue from 'vue'
  import {formatDate} from '../../common/js/dateUtil'
  import cartcontrol from '../cartcontrol/cartcontrol'
  import ratingselect from '../ratingselect/ratingselect'
  import split from '../split/split'

  const ALL = 2

  export default {
    props: {
      food: {
        type: Object
      }
    },
    data() {
```

```
    return {
      showFlag: false,
      selectType: ALL,
      onlyContent: true,
      desc: {
        all: '全部',
        positive: '推荐',
        negative: '吐槽'
      }
    }
  },
  methods: {
    show() {
      this.showFlag = true
      this.selectType = ALL
      this.onlyContent = true
      this.$nextTick(() => {
        if (!this.scroll) {
          this.scroll = new BScroll(this.$els.food, {
            click: true
          })
        } else {
          this.scroll.refresh()
        }
      })
    },
    hide() {
      this.showFlag = false
    },
    addFirst(event) {
      if (!event._constructed) {
        return
      }
      this.$dispatch('cart.add', event.target)
      Vue.set(this.food, 'count', 1)
    },
    needShow(type, text) {
      if (this.onlyContent && !text) {
        return false
      }
      if (this.selectType === ALL) {
        return true
      } else {
        return type === this.selectType
        }
      }
    },
    events: {
      'ratingtype.select'(type) {
        this.selectType = type
        this.$nextTick(() => {
          this.scroll.refresh()
        })
      },
      'content.toggle'(onlyContent) {
        this.onlyContent = onlyContent
        this.$nextTick(() => {
          this.scroll.refresh()
        })
      }
    },
    filters: {
      formatDate(time) {
        let date = new Date(time)
        return formatDate(date, 'yyyy-MM-dd hh:mm')
      }
    },
    components: {
      cartcontrol,
      ratingselect,
      split
    }
  }
</script>
```

4) 定义日期格式化模块

src/common/js/dateUtil.js

```
export function formatDate(date, fmt) {
 if (/(y+)/.test(fmt)) {
  fmt = fmt.replace(RegExp.$1, (date.getFullYear() + '').substr(4 - RegExp.$1.length))
 }
 let o = {
  'M+': date.getMonth() + 1,
  'd+': date.getDate(),
  'h+': date.getHours(),
  'm+': date.getMinutes(),
  's+': date.getSeconds()
 }
 for (let k in o) {
  if (new RegExp(`(${k})`).test(fmt)) {
   let str = o[k] + ''
   fmt = fmt.replace(RegExp.$1, (RegExp.$1.length === 1) ? str : padLeftZero(str))
  }
 }
 return fmt
}

function padLeftZero(str) {
 return ('00' + str).substr(str.length)
}
```

5) 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: 16px
    border-top: 1px solid rgba(7, 17, 27, 0.1)
    border-bottom: 1px solid rgba(7, 17, 27, 0.1)
    background: #f3f5f7
</style>
```

6) 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" v-ref:food></food>

data () {
  return {
    selectedFood: {}
  }
},
methods: {
  selectFood(food, event) {
    if (!event._constructed) {
      return
    }
    this.selectedFood = food
    this.$refs.food.show()
  }
}

components: {
  food
}
```

2.18. 开发 ratings 组件

1) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .ratings
    position: absolute
    top: 174px
    bottom: 0
    left: 0
    width: 100%
    overflow: hidden
```

```
.overview
  display: flex
  padding: 18px 0
  .overview-left
    flex: 0 0 137px
    padding: 6px 0
    width: 137px
    border-right: 1px solid rgba(7, 17, 27, 0.1)
    text-align: center
    @media only screen and (max-width: 320px)
      flex: 0 0 120px
      width: 120px
    .score
      margin-bottom: 6px
      line-height: 28px
      font-size: 24px
      color: rgb(255, 153, 0)
    .title
      margin-bottom: 8px
      line-height: 12px
      font-size: 12px
      color: rgb(7, 17, 27)
    .rank
      line-height: 10px
      font-size: 10px
      color: rgb(147, 153, 159)
  .overview-right
    flex: 1
    padding: 6px 0 6px 24px
    @media only screen and (max-width: 320px)
      padding-left: 6px
    .score-wrapper
      margin-bottom: 8px
      font-size: 0
      .title
        display: inline-block
        line-height: 18px
        vertical-align: top
        font-size: 12px
        color: rgb(7, 17, 27)
      .star
        display: inline-block
        margin: 0 12px
        vertical-align: top
```

```
          .score
            display: inline-block
            line-height: 18px
            vertical-align: top
            font-size: 12px
            color: rgb(255, 153, 0)
        .delivery-wrapper
          font-size: 0
          .title
            line-height: 18px
            font-size: 12px
            color: rgb(7, 17, 27)
          .delivery
            margin-left: 12px
            font-size: 12px
            color: rgb(147, 153, 159)
    .rating-wrapper
      padding: 0 18px
      .rating-item
        display: flex
        padding: 18px 0
        border-1px(rgba(7, 17, 27, 0.1))
        .avatar
          flex: 0 0 28px
          width: 28px
          margin-right: 12px
          img
            border-radius: 50%
        .content
          position: relative
          flex: 1
          .name
            margin-bottom: 4px
            line-height: 12px
            font-size: 10px
            color: rgb(7, 17, 27)
          .star-wrapper
            margin-bottom: 6px
            font-size: 0
            .star
              display: inline-block
              margin-right: 6px
              vertical-align: top
            .delivery
```

```
              display: inline-block
              vertical-align: top
              line-height: 12px
              font-size: 10px
              color: rgb(147, 153, 159)
          .text
            margin-bottom: 8px
            line-height: 18px
            color: rgb(7, 17, 27)
            font-size: 12px
          .recommend
            line-height: 16px
            font-size: 0
            .icon-thumb_up, .item
              display: inline-block
              margin: 0 8px 4px 0
              font-size: 9px
            .icon-thumb_up
              color: rgb(0, 160, 220)
            .item
              padding: 0 6px
              border: 1px solid rgba(7, 17, 27, 0.1)
              border-radius: 1px
              color: rgb(147, 153, 159)
              background: #fff
          .time
            position: absolute
            top: 0
            right: 0
            line-height: 12px
            font-size: 10px
            color: rgb(147, 153, 159)
</style>
```

2) 页面模板

```
<template>
  <div class="ratings" v-el:ratings>
    <div class="ratings-content">
      <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 :size="36"  :score="seller.serviceScore"></star>
      <span class="score">{{seller.serviceScore}}</span>
     </div>
     <div class="score-wrapper">
      <span class="title">商品评分</span>
      <star :size="36"  :score="seller.foodScore"></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 :select-type="selectType"
              :only-content="onlyContent"
              :desc="desc"
              :ratings="ratings"></ratingselect>
   <div class="rating-wrapper">
    <ul>
     <li v-for="rating in ratings" v-show="needShow(rating.rateType, rating.text)"
class="rating-item">
      <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 :size="24"  :score="rating.score"></star>
        <span class="delivery"
v-show="rating.deliveryTime">{{rating.deliveryTime}}</span>
       </div>
       <p class="text">{{rating.text}}</p>
       <div class="recommend"  v-show="rating.recommend &&
rating.recommend.length">
        <span class="icon-thumb_up"></span>
        <span class="item"  v-for="item in rating.recommend">{{item}}</span>
```

```
              </div>
              <div class="time">
                {{rating.rateTime | formatDate}}
              </div>
            </div>
          </li>
        </ul>
      </div>
    </div>
  </div>
</template>
```

3) JS

```
<script type="text/ecmascript-6">
  import BScroll from 'better-scroll'
  import {formatDate} from '../../common/js/dateUtil'
  import star from '../star/star'
  import ratingselect from '../ratingselect/ratingselect'
  import split from '../split/split'

  const ALL = 2
  const ERR_OK = 0

  export default {
    props: {
      seller: {
        type: Object
      }
    },
    data() {
      return {
        ratings: [],
        selectType: ALL,
        onlyContent: true,
        desc: {
          all: '全部',
          positive: '推荐',
          negative: '吐槽'
        }
      }
    },
    created() {
      this.$http.get('/api/ratings').then((response) => {
        response = response.body
        if (response.errno === ERR_OK) {
          this.ratings = response.data
          this.$nextTick(() => {
            this.scroll = new BScroll(this.$els.ratings, {
              click: true
            })
          })
        }
      })
    },
    methods: {
      needShow(type, text) {
        if (this.onlyContent && !text) {
          return false
        }
        if (this.selectType === ALL) {
          return true
        } else {
          return type === this.selectType
        }
      }
    },
    events: {
      'ratingtype.select'(type) {
        this.selectType = type
        this.$nextTick(() => {
          this.scroll.refresh()
        })
      },
      'content.toggle'(onlyContent) {
        this.onlyContent = onlyContent
        this.$nextTick(() => {
          this.scroll.refresh()
        })
      }
    },
    filters: {
      formatDate(time) {
        let date = new Date(time)
        return formatDate(date, 'yyyy-MM-dd hh:mm')
      }
    },
    components: {
      star,
      split,
      ratingselect
    }
  }
</script>
```

2.19. 开发 seller 组件

1) 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
```

2) stylus 样式

```
<style lang="stylus" rel="stylesheet/stylus">
  @import "../../common/stylus/mixin.styl"

  .seller
    position: absolute
    top: 174px
    bottom: 0
    left: 0
    width: 100%
    overflow: hidden
    .overview
      position: relative
      padding: 18px
    .title
      margin-bottom: 8px
      line-height: 14px
      color: rgb(7, 17, 27)
      font-size: 14px
    .desc
      padding-bottom: 18px
      border-1px(rgba(7, 17, 27, 0.1))
      font-size: 0
      .star
        display: inline-block
        margin-right: 8px
        vertical-align: top
      .text
        display: inline-block
        margin-right: 12px
        line-height: 18px
        vertical-align: top
        font-size: 10px
        color: rgb(77, 85, 93)
    .remark
      display: flex
      padding-top: 18px
      .block
        flex: 1
        text-align: center
        border-right: 1px solid rgba(7, 17, 27, 0.1)
        &:last-child
          border: none
        h2
          margin-bottom: 4px
          line-height: 10px
          font-size: 10px
          color: rgb(147, 153, 159)
        .content
          line-height: 24px
          font-size: 10px
          color: rgb(7, 17, 27)
          .stress
            font-size: 24px
    .favorite
      position: absolute
      width: 50px
      right: 11px
      top: 18px
      text-align: center
      .icon-favorite
        display: block
        margin-bottom: 4px
        line-height: 24px
        font-size: 24px
        color: #d4d6d9
        &.active
          color: rgb(240, 20, 20)
      .text
        line-height: 10px
        font-size: 10px
        color: rgb(77, 85, 93)
  .bulletin
    padding: 18px 18px 0 18px
    .title
      margin-bottom: 8px
      line-height: 14px
      color: rgb(7, 17, 27)
      font-size: 14px
    .content-wrapper
      padding: 0 12px 16px 12px
      border-1px(rgba(7, 17, 27, 0.1))
      .content
        line-height: 24px
        font-size: 12px
        color: rgb(240, 20, 20)
    .supports
      .support-item
        padding: 16px 12px
        border-1px(rgba(7, 17, 27, 0.1))
        font-size: 0
        &:last-child
          border-none()
      .icon
        display: inline-block
        width: 16px
        height: 16px
        vertical-align: top
        margin-right: 6px
        background-size: 16px 16px
        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: 16px
        font-size: 12px
        color: rgb(7, 17, 27)
  .pics
    padding: 18px
    .title
      margin-bottom: 12px
      line-height: 14px
      color: rgb(7, 17, 27)
      font-size: 14px
    .pic-wrapper
      width: 100%
      overflow: hidden
      white-space: nowrap
      .pic-list
        font-size: 0
        .pic-item
          display: inline-block
          margin-right: 6px
          width: 120px
          height: 90px
          &:last-child
            margin: 0
  .info
    padding: 18px 18px 0 18px
    color: rgb(7, 17, 27)
    .title
      padding-bottom: 12px
      line-height: 14px
      border-1px(rgba(7, 17, 27, 0.1))
      font-size: 14px
    .info-item
      padding: 16px 12px
      line-height: 16px
```

```
        border-1px(rgba(7, 17, 27, 0.1))
        font-size: 12px
        &:last-child
          border-none()
</style>
```

3) 页面模板

```
<template>
  <div class="seller" v-el:seller>
    <div class="seller-content">
      <div class="overview">
        <h1 class="title">{{seller.name}}</h1>
        <div class="desc border-1px">
          <star :size="36" :score="seller.score"></star>
          <span class="text">({{seller.ratingCount}})</span>
          <span class="text">月售{{seller.sellCount}}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">{{favoriteText}}</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 v-if="seller.supports" class="supports">
          <li class="support-item border-1px" v-for="item in seller.supports">
            <span class="icon" :class="classMap[seller.supports[$index].type]"></span>
            <span class="text">{{seller.supports[$index].description}}</span>
          </li>
        </ul>
      </div>
      <split></split>
      <div class="pics">
        <h1 class="title">商家实景</h1>
        <div class="pic-wrapper" v-el:pic-wrapper>
          <ul class="pic-list" v-el:pic-list>
            <li class="pic-item" v-for="pic in seller.pics">
              <img :src="pic" width="120" height="90">
            </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>
```

4) JS

```
<script>
  import BScroll from 'better-scroll'
  import star from 'components/star/star'
  import split from 'components/split/split'
```

```
export default {
  props: {
    seller: {
      type: Object
    }
  },
  data() {
    return {
      favorite: !!localStorage.favorite
    }
  },
  computed: {
    favoriteText() {
      return this.favorite ? '已收藏' : '收藏'
    }
  },
  created() {
    this.classMap = ['decrease', 'discount', 'special', 'invoice', 'guarantee']
  },

  ready () {
    this._initScroll()
  },
  methods: {
    toggleFavorite(event) {
      if (!event._constructed) {
        return
      }
      this.favorite = !this.favorite
      localStorage.favorite = this.favorite
    },

    _initScroll() {
      console.log('_initScroll', this.seller)
      if(this.seller.pics) {

        this.$nextTick(() => {
          this.scroll = new BScroll(this.$els.seller, {
            click: true
          })
        })

        const picWidth = 120
```

```
          const margin = 6
          const width = (picWidth+margin)*this.seller.pics.length - margin + 'px'
          this.$els.picList.style.width = width
          this.$nextTick(() => {
            this.picScroll = new BScroll(this.$els.picWrapper, {
              scrollX: true,
              eventPassthrough: 'vertical'
            })
          })
        }
      },
    },
    components: {
      star,
      split
    }
  }
</script>
```

2.20. 购物车小球动画

1) 模板页面

```
<div class="ball-container">
  <div class="ball" v-for="ball in balls" transition="drop" v-show="ball.show">
    <div class="inner inner-hook"></div>
  </div>
</div>
```

2) JS

```
<script>
  import BScroll from 'better-scroll';
  import cartcontrol from 'components/cartcontrol/cartcontrol';

  export default {
    data() {
      return {
```

```
      balls: [
        {
          show: false
        },
        {
          show: false
        },
        {
          show: false
        },
        {
          show: false
        },
        {
          show: false
        }
      ],
      dropingBalls: []
    };
  },
  methods: {
    drop (el) {
      for(let i=0;i<this.balls.length;i++) {
        let ball = this.balls[i]
        if(!ball.show) {
          ball.show = true
          ball.el = el
          this.dropingBalls.push(ball)
          return
        }
      }
    }
  },
  transitions: {
    drop: {
      //指定动画开始时的样式状态
      beforeEnter (el) {
        const ball = this.dropingBalls.shift()
        const rect = ball.el.getBoundingClientRect()
        const x = rect.left - 32
        const y = -(window.innerHeight-rect.top-22)

        el.style.transform = `translate3d(0, ${y}px, 0)`
        el.style.webkitTransform = `translate3d(0, ${y}px, 0)`
```

```
          const innerEl = el.children[0]
          innerEl.style.transform = `translate3d(${x}px, 0, 0)`
          innerEl.style.webkitTransform = `translate3d(${x}px, 0, 0)`

          //将 ball 保存到 el 中，给 afterEnter 使用
          el.ball = ball
        },
        //指定动画结束时的样式状态
        enter (el) {
          //强制页面重绘(否则动画瞬间完成，没有效果)
          const rf = el.offsetHeight
          //异步指定动画结束时的样式
          this.$nextTick(function () {
            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)`
          })
        },
        //指定动画完成后调用
        afterEnter (el) {
          el.ball.show = false //重置为初始状态，为了反复使用(此操作不会隐藏当前小球)
          el.style.display = 'none' //手动隐藏
        }
      }
    }
  };
</script>
```

2.21. vue1.x 升级 vue2.x

1) 说明

- vue 从 1.x 升级到 2.x，几乎 90% 的 API 和核心概念都没有变
- vue 官方提供了专门用于升级的工具库(迁移工具)

2) API 变化

* vue-router 变化

* 初始化路由

* v-link ==> <router-link>

* keep-alive ===> <keep-alive>

* v-for 指令

* 移除$index

* v-for=”(item, index) in items”

* v-el 与 v-ref 指令

* ref 指令 ref=’xxx’

* this.$refs.xxx

* 组件的模板

* 只允许有一个根标签

* 组件间通信

* $dispatch()/$broadcast()被移除，只留下$emit

* events 选项被移除，通过 v-on 指令绑定监听

* 不能直接在子组件中修改父组件传入的 prop

* 过渡

* <transition>

* transitions 选项被移除

* 过滤 css

.fade-enter：进入的开始状态(不可见)，在此指定进入显示前不可见的样式

.fade-leave-active: 离开的结束状态(不可见)，在此指定离开后不可见的样式

.fade-enter-active: 进入的结束状态(可见)，在此指定显示的 transition

.fade-leave-active: 离开的结束状态(不可见),在此指定隐藏的 transition

* JS 动画

```
<transition name="drop"
        @before-enter="beforeDrop"
        @enter="dropping"
        @after-enter="afterDrop"
        v-bind:css="false">
```

* 虚拟 DOM/组件的生命周期

* 生命周期回调函数有增减

* 移除: ready()

* 增加: mounted() / beforeUpdate() / afterUpdate()

3) 操作

- 使用 vue-cli 下载基于 vue2 + vue-router2 的项目
- 下载其它依赖的包

```
npm install better-scroll mockjs moment vue-resource --save
npm install stylus stylus-loader --save-dev

npm install pubsub-js --save
```

- 将项目代码拷贝到新项目对应目录下
- 下载并使用 vue 的迁移工具进行升级

```
# install

npm install --global vue-migration-helper

# navigate to a Vue 1.x project directorycd path/to/my-vue-project

# scan all files in the current directory

vue-migration-helper# scan all files in specific sub-directories

vue-migration-helper src folder-a folder-b
```