引言

Mvvm框架

Vue 对比angular react易上手 学习简单 轻量 只有20k 借鉴了angular 指令 和react组件化 而且还独有计算属性

数据改变 驱动视图自动更新通过getter setter

组件化 拓展html元素封装可重用代码

组件设计原则 页面上每一个可是可交互的区域都是一个组件

组件就近维护 模板 逻辑 样式 在一个vue文件

页面是组件的容器,页面是由不同的组件拼合成的

项目的开始

脚手架 vue-cli 他可以帮我们搞定目录结构 本地调试 代码部署 热加载 单元测试 简化开发初始化

通过webpack生成的应用的目录结构是webpack template下的目录结构是一样的

Cli安装

目录介绍

Babelrc 将es6转化成es5的一些配置 会调用module中的一些插件进行转化

代码运行

Index.js -> main.js -> app.vue -> components/hello 页面呈现hello内容

Webpack 简介

配置比较复杂 他可以通过编译将项目编译成几个文件

他会检测文件的后缀名,然后通过lodaer的方法实现对文件的编译 会将小图片base64处理打包到js文件中 所以项目中的图片最终上线的时候可能一个都没有

图标字体文件在线制作

登录icomoon.io 点击import icons按钮 上传你的svg图片 点击选中图标 点击生成字体

下载图标按钮

项目目录配置 添加svg字体 字体style 引入图片资源

数据mock 模拟数据

到真实环境下就是数据库

Css使用stylu 进行预处理 布局采用弹性盒模型

页面切换使用router 配置mode:’history’模式 实现 商品 评论 商家的路由逻辑

移动端1px像素边框问题

给该元素添加一个伪类after 定义成一条宽度为1px的线

&:after

display: block

position: absolute

left:0

bottom 0

width 100%

border-top: 1px solid $color

content: ' '

根据设备像素比(物理像素和设备实际像素的比例)对伪类div进行y轴缩放 设备像素比为1.5的缩放0.7,设备像素比为1.5的缩放0.5

@media(-webkit-min-device-pixel-ratio:1.5),(min-device-pixel-ratio:1.5)

.border-1px

$::after

-webkit-transform:scaleY(0.7)

transform:scaleY(0.7)

@media(-webkit-min-device-pixel-ratio:2),(min-device-pixel-ratio:2)

.border-1px

$::after

-webkit-transform:scaleY(0.5)

transform:scaleY(0.5)

图片和文字的间距问题

给父元素的字体大小设为0, 后面有需要修改字体的需求再设

2x 3x图自动适配的stylus组件

内容小于屏幕长度,关闭按钮固定定位到底部,内容大于屏幕长度,关闭按钮在滚动条的底部的布局方式

评分系统组件封装

-----------小标题-------- 分割线布局 使用弹性盒模型

使用弹性盒模型进行布局, 两边的线根据屏幕自适应, 面试题