# 一．搭建vuejs环境

1.从node.js官网下载并安装node，安装过程很简单，一路“下一步”就可以了（傻瓜式安装）。

安装完成之后，打开命令行工具，输入 node -v，如下图，如果出现相应的版本号，则说明安装成功。

假如安装在d:/nodejs 用cmd命令行进入nodejs下 输入node -v，出现版本号就证明安装成功

2.在当前nodejs命令行下，输入命令npm install -g cnpm --registry=http://registry.npm.taobao.org

3.安装vue-cli脚手架构建工具

在当前nodejs命令行下运行命令 npm install -g vue-cli ，然后等待安装完成。

通过以上三部，我们需要准备的环境和工具都准备好了，接下来就开始使用vue-cli来构建项目

4.用vue-cli构建项目

要创建项目，首先我们要选定目录，然后再命令行中把目录转到选定的目录。

在这里，我选择d:/vuejs来存放新建的项目，则我们需要先把目录cd到d:/vuejs，

在d:/vuejs命令行中运行命令 vue init webpack firstVue (注：firstVue是自定义名称)

此致环境的搭建基本完成

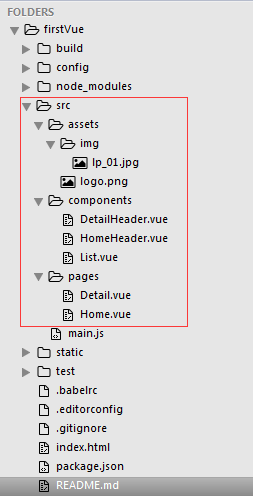
5.要安装依赖包，首先cd到项目文件夹（firstVue文件夹），然后在d:/vuejs/firstVue下运行命令 cnpm install

1. 在d:/vuejs/firstVue下运行项目 npm run dev

# 搭建vuejs商城项目

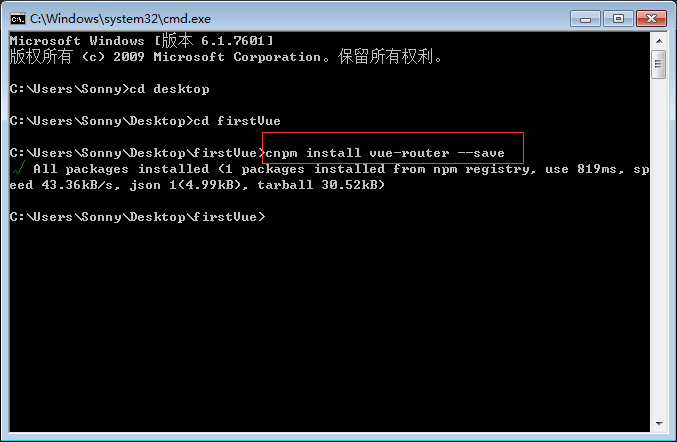
修改项目目录如下

两个页面，Home和detail和其中用到的三个组件。ps：assets目录下新建的img目录，里边放的一张图片是测试用的。



### 安装vue-router 2.0

打开命令行，cd到当前项目的目录，运行 cnpm install vue-router --save  ，f等待安装完成即可



修改根目录下的index.html如下

<!-- index.html -->

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, minimum-scale=1, user-scalable=no">

<title>first-vue</title>

<style type="text/css">

\* { margin: 0; padding: 0; }

html,body { background: #eee; }

ul,li { list-style: none; }

a { text-decoration: none; }

img { vertical-align: middle; }

/\* 跳转页面动画 \*/

.slide-enter,

.slide\_back-enter {

position: absolute;

width: 100%;

}

.slide-leave,

.slide\_back-leave {

position: absolute;

width: 100%;

}

.slide-enter-active,

.slide\_back-enter-active {

transition: all 0.3s linear;

}

.slide-leave-active {

position: absolute;

transition: all 0.3s linear;

transform: translate(-100%);

}

.slide-enter{

transform: translateX(100%);

}

.slide\_back-leave-active {

position: absolute;

transition: all 0.3s linear;

transform: translate(100%);

}

.slide\_back-enter {

transform: translateX(-100%);

}

</style>

</head>

<body>

<div id="app">

<transition :name="transitionName">

<router-view></router-view>

</transition>

</div>

<script type="text/javascript">

// 计算html的font-size

(function(){

function resizeBaseFontSize(){

var rootHtml = document.documentElement,

deviceWidth = rootHtml.clientWidth;

if(deviceWidth > 640){

deviceWidth = 640;

}

rootHtml.style.fontSize = deviceWidth / 7.5 + "px";

}

resizeBaseFontSize();

window.addEventListener("resize", resizeBaseFontSize, false);

window.addEventListener("orientationchange", resizeBaseFontSize, false);

})();

</script>

</body>

</html>

# Main.js

// main.js

// 导入Vue，这个是必需的，在使用Vue之前，必须先导入

import Vue from 'vue'

// 导入 vue-router，并使用

import VueRouter from 'vue-router'

Vue.use(VueRouter)

// 导入 pages 下的 Home.vue

import Home from './pages/Home'

import Detail from './pages/Detail'

// 定义路由配置

const routes = [

{

path: '/',

component: Home

},

{

path: '/detail',

component: Detail

}

]

// 创建路由实例

const router = new VueRouter({

routes

})

// 创建 Vue 实例

new Vue({

el: '#app',

data(){

return {

transitionName: 'slide'

}

},

router, // 在vue实例配置中，用router

watch: {

// 监视路由，参数为要目标路由和当前页面的路由

'$route' (to, from){

const toDepth = to.path.substring(0, to.path.length-2).split('/').length

// 官方给出的例子为 const toDepth = to.path.split('/').length 由于现在只有两个路由路径'/'和'/detail'

// 按照官方给的例子，这两个路由路径深度都为 2 ，所以，这里稍作调整，不知道有什么不妥

// 但目前在这个demo中能正常运行，如果知道更好的方法，欢迎留言赐教

const fromDepth = from.path.substring(0, from.path.length-2).split('/').length

this.transitionName = toDepth < fromDepth ? 'slide\_back' : 'slide'

// 根据路由深度，来判断是该从右侧进入还是该从左侧进入

}

}

})

# Detail.vue

<!-- Detail.vue -->

<template>

<div class="detail">

<detail-header></detail-header>

<img src="../assets/img/lp\_01.jpg" alt="">

<p>崇贤馆始置唐代太宗朝。1999年，李克先生及志同道合者复兴其宗旨。以积累、传播中华优秀传统文化，提供全新国学体验馆为宏愿。</p>

<p>其间，在季羡林、冯其庸等国学大师及著名文史学家傅璇琮、毛佩琦先生指导下，耕注先贤原典，以宣纸线装精品形式呈奉世人。作为一家国学传播机构，崇贤馆始终致力于中华传统文化的传承和推广，以古籍线装宣纸书的形式，对浩繁的史海巨著进行经典复刻。不仅如此，崇贤馆还延请了傅璇琮、毛佩奇等诸位在国学界内享有盛誉的专家和学者担纲学术顾问，以精益求精的治学态度面对每一部崇贤馆的作品，使之成为学术史中无尚的精品。</p>

</div>

</template>

<style>

.detail {

padding: 0.24rem;

font-size: 12px;

}

img {

display: block;

width: 80%;

margin: 0 auto 0.2rem;

}

p {

font-size: 14px;

line-height: 0.5rem;

text-align: justify;

padding-bottom: 0.24rem;

}

</style>

<script>

import DetailHeader from '../components/DetailHeader'

export default {

components: {

DetailHeader

}

}

</script>

# Home.vue

<!-- Home.vue -->

<template>

<div class="container">

<!-- 由于html不区分大小写，所以js中驼峰命名方式在html中要改成用短横线连接的形式 -->

<home-header></home-header>

<div class="content">

<ul class="cont\_ul">

<list

v-for="item in items"

:price="item.price"

:title="item.title">

</list>

</ul>

</div>

</div>

</template>

<style>

.container {

max-width: 640px;

margin: 0 auto;

overflow-x: hidden;

}

.cont\_ul {

padding-top: 0.05rem;

margin: 0 -0.12rem;

}

.cont\_ul:after {

content: "";

display: block;

width: 0;

height: 0;

clear: both;

}

</style>

<script>

// 导入要用到的子组件

import HomeHeader from '../components/HomeHeader'

import List from '../components/List'

export default {

data () {

return {

items: [

{ price: "129.00", title: "大学" },

{ price: "256.00", title: "中庸" },

{ price: "399.00", title: "论语" },

{ price: "998.00", title: "孟子" },

{ price: "99.00", title: "道德经" },

{ price: "89.00", title: "老子" },

{ price: "188.00", title: "金刚经" },

{ price: "209.00", title: "易筋经" },

]

}

},

// 在components字段中，包含导入的子组件

components: {

HomeHeader,

List

}

}

</script>

# DetailHeader.vue

<!-- DetailHeader.vue -->

<template>

<header class="header">

<div class="header\_inner flexWrap">

<div

id="header\_btn\_nav"

class="header\_btn header\_btn\_back"

v-on:click="goBack"

>返回</div>

<div class="header\_cont flex">详情</div>

<div class="header\_btn header\_btn\_cart"></div>

</div>

</header>

</template>

<style>

.flexWrap {

display: -webkit-flex;

display: flex;

}

.flex {

flex: 1;

}

.header {

height: 0.88rem;

}

.header\_inner {

position: fixed;

top: 0;

left: 0;

right: 0;

z-index: 99;

max-width: 640px;

height: 0.88rem;

box-sizing: border-box;

margin: 0 auto;

padding: 0 0.24rem;

border-bottom: 0.02rem solid #80ccd6;

background-color: #fff;

}

.header\_btn {

width: 0.5rem;

height: 100%;

background-repeat: no-repeat;

}

.header\_btn\_back {

line-height: 0.86rem;

}

.header\_cont {

text-align: center;

padding: 0 0.4rem;

line-height: 0.86rem;

font-size: 15px;

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

.header\_btn:active {

opacity: 0.7;

}

</style>

<script>

export default {

methods: {

goBack(){

window.history.back();

}

}

}

</script>

# Homeheader.vue

<!-- HomeHeader.vue -->

<template>

<header class="header">

<div class="header\_inner">

<div class="header\_cont">主页</div>

</div>

</header>

</template>

<style>

.header {

height: 0.88rem;

}

.header\_inner {

position: fixed;

top: 0;

left: 0;

right: 0;

z-index: 99;

max-width: 640px;

height: 0.88rem;

box-sizing: border-box;

margin: 0 auto;

padding: 0 0.24rem;

border-bottom: 0.02rem solid #80ccd6;

background-color: #fff;

}

.header\_cont {

text-align: center;

padding: 0 0.4rem;

line-height: 0.86rem;

font-size: 15px;

overflow: hidden;

text-overflow: ellipsis;

white-space: nowrap;

}

</style>

# List.vue

<!-- List.vue -->

<template>

<li class="sec\_li">

<router-link to="/detail" class="lp\_li\_a">

<div class="lp\_li\_imgWrap">

<img src="../assets/img/lp\_01.jpg" alt="">

</div>

<p class="lp\_li\_name">{{ title }}</p>

<p class="lp\_li\_price">￥{{ price }}元</p>

</router-link>

</li>

</template>

<style scoped>

.sec\_li {

float: left;

width: 50%;

margin-bottom: 0.1rem;

}

.lp\_li\_a {

display: block;

padding: 0.3rem 0;

margin: 0 0.05rem;

text-align: center;

background: #fff;

}

.lp\_li\_imgWrap {

padding: 0.24rem 0;

}

.lp\_li\_imgWrap > img {

width: auto;

height: 2.3rem;

}

.lp\_li\_name {

height: 0.5rem;

line-height: 0.5rem;

font-size: 16px;

color: #333;

}

.lp\_li\_price {

height: 0.5rem;

line-height: 0.5rem;

font-size: 16px;

color: #fb3b3b;

}

</style>

<script>

export default {

props: ['price', 'title']

}

</script>