1. 配置环境
   1. 安装vue-cli、node、淘宝镜像、webpack、yarn，安装之前先查看是否存在
      1. <https://nodejs.org/en/> node-v
      2. npm install -g cnpm –registry=https://registry.npm.taobao.org cnpm-v 报错可能是已经有安装
      3. npm install webpack -g或者（npm install -g webpack)

webpack -v

* + 1. webpack 4.X 开始，需要安装 webpack-cli 依赖 ,所以使用这条命令 npm install webpack webpack-cli -g
    2. npm install @vue/cli -g(推荐) vue-V

npm install --global vue-cli 安装的为cli2版本，创建项目只能vue init webpack 项目名

安装高版本vue-cli，需要先卸载2版本，npm uninstall vue-cli -g

* + 1. npm install yarn -g

1. 创建项目
   1. 创建基于webpack的项目 cli2脚手架
      1. vue init webpack 项目名 创建的为vue2项目
   2. cli>2脚手架
      1. Vue create 项目名 可创建vue2或vue3
      2. Yarn serve
2. mac本地项目关联到github
   1. 本地终端执行

ls -al ~/.ssh (先执行这两句查看是否已经有ssh钥匙)

cat ~/.ssh/id\_rsa.pub

ssh-keygen （没有的话生成钥匙文件再查看）

报错是因为权限问题，可执行下方试试

sudo git config --global user.name username

sudo  git config --global user.email email

git config --list

sudo ssh-keygen -t rsa -C email（不能创建rsa文件，这句是否有用待确认）

* 1. github新建ssh key

将上面生成的密钥添加

* 1. Github 新建项目，不用初始化readme文件
  2. 关联本地仓库和github远程仓库
     1. 进入目标文件夹，输入git init, 使其成为可管理仓库
     2. 将文件缓存到本地仓库 git add . git commit -m “init”
     3. 关联本地仓库与git远程仓库

git remote add origin 项目到ssh地址

* + 1. 推送本地库内容到github

git push -u origin master

* 1. 报错
     1. Please make sure you have the correct access rights

and the repository exists. ssh问题，三.1

* + 1. The authenticity of host 'github.com (20.205.243.166)' can't be established.

ED25519 key fingerprint is SHA256:+DiY3wvvV6TuJJhbpZisF/zLDA0zPMSvHdkr4UvCOqU.

This key is not known by any other names

Are you sure you want to continue connecting (yes/no/[fingerprint])?

输入yes不要按回车，否则报错

1. 项目基础配置
   1. 安装vetur
   2. 浏览器安装Vue Devtools

git clone https://github.com/vuejs/vue-devtools.git

yarn

yarn build

修改packagse中shell-chrome下manifest.json中persistent为true

谷歌导入shell-chrome文件夹，重启vue刷新

* 1. 浏览器自动打开项目
     1. package.json文件 serve加上 --open 重启服务
  2. eslint校验关闭,新的脚手架仍会提示使用muti-name，多单词驼峰命名
     1. 在根目录下创建vue.config.js文件, vue/cli已经存在

lintOnSave: false

* 1. 项目使用sass,安装yarn add sass sass-loader
  2. 项目使用less，安装less less-loader
  3. index.html中引入默认清除样式reset.css
  4. 引入阿里的字体图标样式,非必需https://at.alicdn.com/t/font\_2901600\_cnggnrn1hxp.css
  5. src文件夹目录简写配置别名@, vue create创建的项目已经配置好

根目录下创建jsconfig.json文件

{

“compilerOptions”:{

“baseUrl”:’/’,

‘path’:{

“@/\*”:[“src/\*]

}

},

“exclude”:[“node\_modules”,’dist’] //此行vue createi自带的文件报错

}

* 1. 项目路由分析，搭建主要的路由
     1. 路由组件包含一直在变化的部分，比如页面分为上中下三部分，上下一直不变，那路由只包含中间部分相关的组件home search，

Login 、注册有头部没有底部，路由组件写在page文件夹

* + 1. 非路由组件包含header footer[在首页、footer页有，login、注册没有，后期通过控制判断]，非路由组件写在components文件夹
  1. 需要路由，安装vue-router，npm install vue-router，在router文件夹配置路由index.js,在根目录main.js文件中引入index.js并注册路由,app.vue组件中设置路由出口，<router-view>
     1. 默认安装vue-router 4.x 与vue3更搭配，否则报错警告很多，vue2安装vue-router3.x
     2. npm uninstall vue-router
     3. npm i [vue-router@3.5.2](mailto:vue-router@3.5.2)
  2. router 二次封装，因为同时push或replace跳转路由会导致抛出navigattionDuplicated的警告错误，所以需要重写路由push等方法router/index.js,routes.js存放路由配置
  3. postMan工具测试接口https://www.postman.com/downloads/ Intel版本 收费
  4. axios二次封装,进行请求拦截器，响应拦截器
     1. 请求拦截器，在发请求之前可以处理一些业务
     2. 响应拦截器，当服务器返回数据以后可以处理一些事情
     3. 安装axios yarn add axios
     4. src下新建文件夹api，存放axios请求
     5. axios二次封装request.js
     6. axios api统一管理 index.js

1. 跨域代理解决
   1. webpack.config.js/vue.config.js 重启服务

//*代理跨域*

devServer: {

proxy: {

'/api': {

target: 'http://39.98.123.211', //*服务器地址*

//*路径重写,增加/api是为了与本地请求混淆，但此处实际路径确实需要/api*

// *pathRewrite: { '^/api': '' },*

},

},

},

1. 项目开发
   1. 先写静态页面，再拆分组件，再加逻辑处理
   2. 路由router
      1. 搭建非路由组件放在components文件夹，新的脚手架，组件名需要多单词驼峰命名，引入到app.vue中
      2. 搭建路由组件，引入vue-rouer，组件放在pages/views文件夹中，router文件夹中配置路由, main中引入并注册路由
      3. 路由重定向：{path: ‘/’,redirect:’/home’}
      4. 路由跳转的方式：声明式导航、编程式导航

声明式导航能做的编程式导航都能做，而且还能进行额外的处理,通过$router push replace, $route读取路由

声明式导航只能进行路由跳转路由router-link to，相当于a标签 ,加上target=”\_blank”可以新开页面

* + 1. 配置路由meta信息，判断footer组件是否需要显示结合v-show,meta属性会添加在$route上

1. 知识点
   1. v-model 用于表单元素
   2. 编程式路由导航$router push replace
   3. 对于公共组件footer组件，login不显示home显示，
      1. 可以通过v-show判断路由解决,watch监测$route变化
      2. 根据路由meta元信息 结合v-show
   4. 路由传参：
      1. params参数：属于路由当中的一部分，需要注意，在配置路由的时候需要占位，

params参数如果是占位必传项不传的话跳转会连path都丢失

传空串也会出问题，通过undefined解决

如果非必传需要设置query属性后面加问号代表可传可不传

this**.**$router**.**push({

name: 'search',

params: {

// *keyword: '' || undefined, //传值为空传*

keyword: this**.**inputVal //*路由中配置的params属性*

},

query: {

search: this**.**inputVal

}

})

* + 1. query参数：不属于路径当中的一部分，类似与ajax中的query参数/home?k=1,不需要占位
    2. 字符串跳转

this**.**$router**.**push("/search/"+this**.**inputVal+"?search="+this**.**inputVal)

* + 1. 模板字符串跳转
    2. 对象跳转，需要在路由中给组件name命名

{

// *path: '/search/:keyword', //params参数需要占位*

path: '/search/:keyword?', //*params参数可传可不传*

component: Search,

meta: {

showFooter: true

},

name: 'search',

},

// *对象形式路由跳转，最常用s*

this**.**$router**.**push({

name: 'search',

params: {

// *keyword: '' || undefined, //传值为空传*

keyword: this**.**inputVal //*路由中配置的params属性*

},

query: {

search: this**.**inputVal

}

},

// *解决多次跳转同一个路径且参数不变navigattionDuplicated的警告错误*

()=>{},

()=>{}

)

* + 1. 路由组件传props属性，需要在路由中配置，可以通过props传递pramas参数，在具体组件的props中接收,可以直接使用

{

// *path: '/search/:keyword', //params参数需要占位*

path: '/search/:keyword?', //*params参数可传可不传*

component: Search,

meta: {

showFooter: true

},

name: 'search',

// *一般不通过props传递路由参数，直接使用$route获取*

// *props: true ,//只能传递props参数*

// *props: { //给路由组件传递额外的参数，在组件props属性中可接收*

// *a: 1,*

// *b: 2*

// *}*

props: ($route) => { //*最常用，pramas query参数都可传递，props中接收*

return {

keyword: $route**.**params**.**keyword,

search: $route**.**query**.**search

}

}

},

<template>

<div>search----{{keyword}}</div>

</template>

<script>

export default {

props: ['keyword'],

1. 商城类左右二级导航TypeNav
2. 静态页面拆分成组件
3. 轮播图swiper
4. 报错汇总
   1. ESLint: Component name "Login" should always be multi-word.(vue/multi-word-component-names)

组件名得用驼峰多个单词，解决与html标签冲突问题

1. 'import ... =' can only be used in TypeScript files.引入组件报错，删除自动引入内容，重新写可能正确
2. export 'default' (imported as 'VueRouter') was not found in 'vue-router' (possible exports: NavigationFailureType, RouterLink, RouterView, START\_LOCATION, createMemoryHistory, createRouter, createRouterMatcher, createWebHashHistory, createWebHistory, isNavigationFailure, matchedRouteKey, onBeforeRouteLeave, onBeforeRouteUpdate, parseQuery, routeLocationKey, routerKey, routerViewLocationKey, stringifyQuery, useLink, useRoute, useRouter, viewDepthKey)

import \* as VueRouter from 'vue-router'

但是仍不能解决报错，所以vue2+vue-router3

1. Refused to apply style from 'http://0.0.0.0:8080/iconfont.css' because its MIME type ('text/html') is not a supported stylesheet MIME type, and strict MIME checking is enabled.

css中有import引入

1. 编程式路由多次跳转至当前路由，参数不变，会抛出navigattionDuplicated的警告错误，以为路由的push返回的是promise对象，所以需要一个成功或失败的回调 router重写
2. vue-cli3配置proxy无效
3. 参考文件
   1. <https://gitee.com/jch1011/shangpinhui_0607> 源代码