**安装node.js**

网上很多教材，安装很简单，安装之后在CMD中输入 node -v 查看是否安装正确；

下载编码器：Visual Studio Code;

**Node.js：**

Npm: 包管理器通过nmp能够导入所需要的包

Node 的package.json (npm init后可以生成):

{

"name": "nodeproject",

"version": "1.0.0",

"main": "index.js",

"scripts": {

"test": "echo \"Error: no test specified\" && exit 1"，

"start": "node app.js" //npm run start 可以直接运行app.js 相当于设置本程序的入口

},

"author": "",

"license": "ISC",

"dependencies": { //npm导入包的默认位置：/npm install bootstrap@

"express": "^4.16.4"

},

"devDependencies": {//npm install --save-dev bootstrap@3

"bootstrap": "^3.4.1"

},

"description": ""

}

**创建vue-cli:**

1. vue init webpack my-projec
2. 进入项目 cd my-project
3. 运行 ：cnpm run dev

**Windows平台下MongoDB安装和环境搭建**

1. 下载安装包或者压缩包
2. 添加db存储和日志存储文件夹
3. 添加服务，配置和环境变量，启动Mongo

C:\Program Files\MongoDB\Server\4.0\bin

**从GitHUB上下载的项目使用方法：**

1. Cd 进入项目文件夹
2. cnpm install //安装依赖
3. Cnpm run dev //运行项目 注意：dev具体看项目，一般来说是dev
4. Cnpm run build //发布环境

**VUE启动时eslint disable错误:**

1. 在build文件夹下webpack.base.conf.js里面找到:
2. ...(config.dev.useEslint ? [createLintingRule()] : []);
3. 注释掉上面这句话就可以了；是校验语法的问题

**Package.json:**

Dependencies:{

} //项目依赖，这是项目上线之后依然要安装的依赖

Devdependencies:{

}//开发依赖，这是我们在开发过程中需要的依赖，项目上线后就不需要了

**vue-REsource:**

1. cnpm i vue-resource --save 下载vue-REsource架包
2. 在main.js里面导入并注册vue-resource
3. import VueResource from 'vue-resource' //导入
4. Vue.use(VueResource) //注册 这样就可以全局使用或者在某个vue实例中使用resource了

**Vue项目使用ElementUI框架：**

* 在main.js文件中导入：

import ElementUI from "element-ui"

import "element-ui/lib/theme-chalk/index.css"

* 通过Vue.use()使用：

Vue.use(ElementUI)

* 接着就可以直接通过官方API中使用ElementUI框架

**Vue项目使用axios:**

axios 简介

axios 是一个基于Promise 用于浏览器和 nodejs 的 HTTP 客户端，它本身具有以下特征：

* 从浏览器中创建 XMLHttpRequest
* 从 node.js 发出 http 请求
* 支持 Promise API
* 拦截请求和响应
* 转换请求和响应数据
* 取消请求
* 自动转换JSON数据
* 客户端支持防止 CSRF/XSRF

实例方法

可用的实例方法如下所示。 指定的配置将与实例配置合并。

axios＃request（config）  
axios＃get（url [，config]）  
axios＃delete（url [，config]）  
axios＃head（url [，config]）  
axios＃post（url [，data [，config]]）  
axios＃put（url [，data [，config]]）  
axios＃patch（url [，data [，config]]）

1. 导入依赖：cnpm i aixos --save
2. 在main.js入包：import axios from "axios"
3. import VueAxios from "vue-axios"
4. Vue.use(VueAxios,axios)

注意:不能和resource同时使用，会报错

Axios全局使用的方法网上说有三种：

1. Vue原生链绑定：Vue.prototype.$ajax = axios（不推荐，会污染原生链）
2. 和VueAxios一同使用，就可义全局使用了
3. 结合 Vuex的action

**vue-cli脚手架生成的Vue项目使用本地数据：**

1.在webpack.dev.conf.js中写入：

const express = require('express')

const app = express()

const myData = require('../src/db-data/dataJson.json')

const ratings = myData.ratings

const apiRouter = express.Router()

// apiRouter.get("/myjson",(req,res,next)=>{

// res.json(myData);

// })

app.use('/api',apiRouter)

1. 在devserver中增加：

before (app) {

app.get('/api/myjson', function (req, res) {

// 服务端收到请求后返回给客户端一个json数据

res.json({

// 当我们数据正常时，我们通过传递errno字符为0表示数据正常

errno: 0,

// 返回json中的卖家数据

data: myData

})

})

}

**Vue中样式使用技巧:**

1. 使用scss即css预处理器：

定义了一种新的专门的编程语言，编译后形成正常的css文件，为css增加一些编程特性，无需考虑浏览器的兼容性（完全兼容css3），让css更加简洁、适应性更强，可读性更佳，更易于代码的维护等诸多好处。CSS预处理语言有SCSS (SASS) 和LESS、POSTCSS

1. 导入包：

npm  install sass-loader --save-dev

npm install node-sass --sava-dev

1. 在build文件夹下的webpack.base.conf.js的rules里面添加配置：

{

test: /\.scss$/,

loaders: ['style', 'css', 'sass']

}

1.先用混合样式@mixin定义模板：

//宽高

@mixin wh($width, $height){

    width: $width; //$width 表示参数

    height: $height;

}

2.在使用@include调用：

@include wh(100%, 100%);

**Vue中使用bootstrap**

1. 因为bootstrap需要jquery，首先要导入配置Jquery：

·$ npm install jquery --save-dev

·在webpack.config.js 添加内容(蓝色字为需要添加的内容)

·+ const webpack = require("webpack");

module.exports = {

entry: './index.js',

output: {

path: path.join(\_\_dirname, './dist'),

publicPath: '/dist/',

filename: 'index.js'

},

+ plugins: [

new webpack.ProvidePlugin({

jQuery: 'jquery',

$: 'jquery'

})

]

}

·在入口文件index.js 里面添加内容：

import $ from 'jquery' ;

1. 安装Bootstrap

·$ npm install --save-dev bootstrap

·在入口文件index.js里引入相关：

import './node\_modules/bootstrap/dist/css/bootstrap.min.css';

import './node\_modules/bootstrap/dist/js/bootstrap.min.js';

之后就能够全局使用bootstrap样式了

Vue中使用Mint-ui:

//导入mint-ui 中的组件

import { Header } from 'mint-ui';

Vue.component(Header.name, Header);

import { Tabbar, TabItem } from 'mint-ui';

Vue.component(Tabbar.name, Tabbar);

Vue.component(TabItem.name, TabItem);

import Mint from "mint-ui"

Vue.use(Mint)

import 'mint-ui/lib/style.css'

Vue使用MUI：

Mui是和BootStrap类似，不是Vue的组件，而是提供样式表和代码块，是HTML片段；

首先，在GUIHUB上下载Mui包

下载地址：<https://github.com/dcloudio/mui>

在工程SRC目录下新建lib目录，

下载之后将F:\tool\mui-master\dist文件夹放入工程lib目录下

之后就可以在页面中直接使用MUI

//导入MUI的样式

import "./lib/mui/css/mui.min.css"

安装sass：

1 使用vscode自带的插件：sass live compiler

//使用vscode的插件 sass live compiler,编译好。scss文件后，按下shift+alt+p 输入sass，然后选择监听模式，就能够自动生成对应css文件

//sass 支持自定义变量

$w:16px;

$color:red;

$side:top;

.box {

width: $w

}

.btn{

font-size: $w

}

.sds{

height: $w

}

.h{

$w:12px; //sass 支持局部变量，与全局变量同名的情况下，使用的是局部变量

width:2 \* $w; //sass支持 变量运算

height: 0.12rem;

//sass 支持嵌套

.btn{

border-#{$side}: 15px; //sass 特殊变量：#{$xxx}

background: $color;

}

&:hover{

background: $color

}

}

//sass 支持for 循环：through 表示包含开头和结尾的参数，to表示不包含结尾

@for $i from 1 through 3{

.item-#{$i}{

@if $i >2 {

width: $i + px

}

@else{

width: $i \* 2 + px

}

}

}

@for $i from 1 to 3{

.item-#{$i}{

@if $i>2{

width: $i + px

}

@else{

width: ($i \* 2) +px

}

}

}

//sass支持 混合指令 mixin 可以定义一个css代码块，然后在别的地方使用 @include 引用

//不带参数

@mixin reset {

padding: 0;

margin: 0;

}

html,body,div{

@include reset()

}

//带参数

@mixin reset($p,$m) {

padding: $p;

margin: $m;

}

html,body,div{

@include reset(10px,2px)

}

##webpack

基础配置：webpack.base.js

开发阶段配置：webpack.dev.js

生产阶段配置：webpack.prod.js

++构建配置管理的可选方案：

通过多个配置文件管理不同环境的构建，webpack --config参数进行控制

将构建配置设计成一个库，比如：hjs-webpack,Neutrino,webpack-blocks

抽象成要给工具进行管理：比如：create-react-app,kyt,nwb

将所有的配置放在一个文件，通过--env参数控制分支选择

Vue项目中，使用空格报错解决办法：

报错：Expected indentation of 0 spaces but found 2

解决办法：在.eslintrc.js中的rules对象中，添加一条"indent": ["off", 2]就可以了；

  rules: {

    // allow async-await

    'generator-star-spacing': 'off',

    // allow debugger during development

    'no-debugger': process.env.NODE\_ENV === 'production' ? 'error' : 'off',

    "indent": ["off", 2]

  }

Vue项目中空格、注释报错，是使用脚手架配置项目的时候，启动了useEslint.js

只要在项目目录下config/index.js中找到useEslint：true，改为false就好了