# 从零开发后台管理系统

之前端vue项目开发教程

前言

大家好，我是白米饭，咱们的课程叫《从零开发后台管理系统之前端vue项目开发》，咱们总的课程分为三套：前端vue开发部分、后端java的spring cloud服务接口开发部分和docker容器化部署部分，涵盖了前端和服务端以及部署全栈式开发内容。咱们这套课程是前端vue篇，所以在这套课程里，我会用vue，带着大家从零去搭建这样一个后台管理系统，大家看完这套课程，也能够独自去搭建和使用这个系统框架；咱们课程的特点是：从零开始，从安装环境开始实际操作，内容通俗易懂，大白话式讲解，真实的接口真实的项目，让大家易于理解和学习。

## 第一章 项目简介

### 项目描述

本系统是一个后台管理系统模版项目,主要基于vue和element-ui实现,代码上集成了常用的插件，功能上提供了常用且必需的功能，方便大家基于此项目，快速开发自己的项目。

### 功能描述

系统集成了登录认证、动态路由、动态菜单、整套权限功能、csdn同款富文本编辑器、文件上传、以及增删改查列表页，多语言切换等功能，基本实现了拿来就用。

### 网站展示

网址：101ycy.com:6001

讲解：登陆，顶部导航栏，左侧是菜单，右侧是菜单对应的页面，页底；基础模块中的综合表单页和使用文档页（操作增删改查）；系统管理中的用户管理，角色管理，资源管理，菜单管理，角色资源授权（操作添加新用户设置权限）（操作添加新菜单，并设置权限）；内容区tab标签，页底始终在底部（操作综合表单页查看）；多语言切换（操作切换）

### 前端技术栈

Vue:前端js框架

Element-ui:跟vue搭配的前端ui框架

Vue-cli:vue脚手架

Axios:调用后端接口的ajax工具

Vue-router:路由工具

Vuex:全局状态管理插件,维护全局响应式变量

Ckeditor:跟csdn同款文本编辑器,非常好用

Vue-i18n:多语言切换工具

Npm:包管理工具

ES6:更好用的js语法

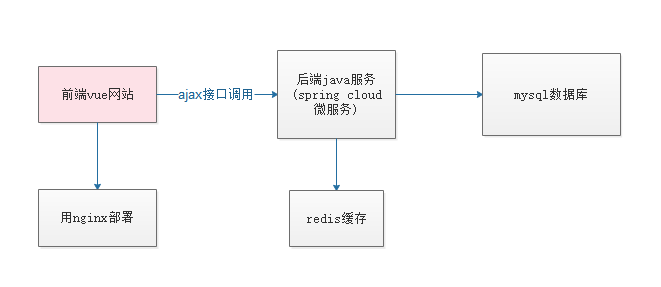
Sass:css预处理器，css扩展语言，更好用的css语法

Momentjs:JavaScript 日期处理类库

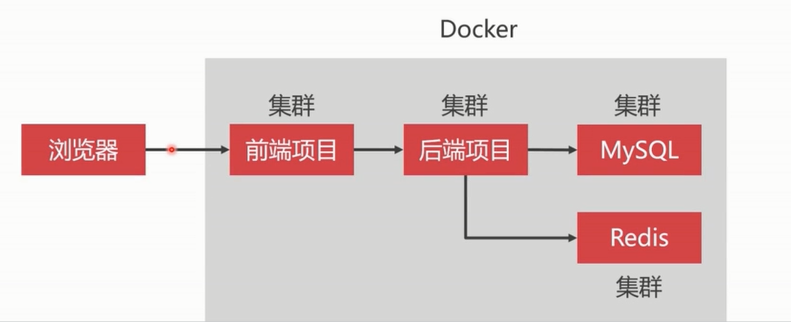
前置知识：html，js，css略懂

### 项目架构

简单模型



完整模型



### 章节介绍

第一章：项目简介，让大家了解咱们的项目是什么，通过网站让大家看到系统有哪些功能，以及用到的技术

第二章：环境搭建，会演示nodejs，vscode，vue-cli的安装和创建项目

第三章：代码目录和模块划分，会带着大家了解项目主要的代码目录和模块划分的思路

## 第二章 环境搭建

### 安装nodejs

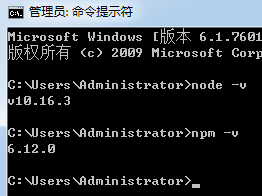
原因:nodejs中包含npm包管理器，这个用来管理系统的插件包，增加或删除包等

方式:

#### 安装nodejs

提示：百度搜索nodejs

打开nodejs官网右边设置中文（不要点击中文网，中文网是最新版本的，最新版的不支持），点击下载页下载长期支持版，然后像安装普通软件一样下载安装，安装完，打开电脑命令行工具，输入node -v和npm-v显示版本号则成功



#### 设置淘宝镜像

推荐方式

设置npm仓库地址为淘宝镜像地址

npm config set registry https:*//registry.npm.taobao.org*

npm config get registry

或者添加cnpm

提示：百度搜淘宝镜像

npm install -g cnpm --registry=https://registry.npm.taobao.org

cnpm config get registry

切换仓库

npm i -g nrm

nrm ls:查看可选仓库地址

nrm test：测试仓库的相应时间

nrm use edunpm:切换仓库

### 安装vscode开发工具

原因：微软出的免费且很好用的前端开发工具

#### 安装vscode

方式：

提示：百度搜vscode，点第一个网址，

进入vscode官网，点击下载按钮，下载安装

#### 安装插件

设置中文Chinese：插件中搜Chinese，安装第一个插件，根据插件中使用方法切换语言

建议安装的vscode插件

Vetur：语法高亮，格式化，智能提示

方式：插件搜索中搜索名称即可

### 安装vue-cli脚手架

原因：Vue 提供了一个官方的 CLI，为单页面应用 (SPA) 快速搭建繁杂的脚手架。它为现代前端工作流提供了 batteries-included（功能齐全） 的构建设置。只需要几分钟的时间就可以运行起来并带有热重载、保存时 lint 校验，以及生产环境可用的构建版本

通俗讲解：就是官方提供的已经搭建好的基础程序，拉下来就能跑，然后在这上边添加自己的代码，比咱们后台管理系统更基础的模版项目

方式：

#### 安装vue-cli

首先，安装vue-cli

提示：百度搜vue-cli，点击安装标签页

新建目录：D:\project\ycy

打开vscode，打开终端，进入目录

如果有vue-cli2，则卸载旧版本

npm uninstall vue-cli -g

安装vue-cli

npm install -g @vue/cli

检查版本是否正确

vue -V或vue --version

### 用vue-cli创建脚手架项目

有两种方式：命令行安装和可视化页面安装

##### 命令行安装

在父级目录下执行：vue create ycy-vue-web

选择手动配置：Manually select features

选择：bable，router，vuex，css pre-cessors

选项意思：

Babel，转译成浏览器可识别的语言，可以让你的项目支持更新的语法，如es6\es7等

TypeScript，支持ts，js的扩展语言

PWA，模拟原生app，渐进式网络应用程序

Router，路由工具

Vuex，状态管理工具

CSS Pre-processors，预处理语言

Linter/Formatter，代码规范

Unit Testing，组件单元测试

E2E Testing，端对端测试，模拟用户真实场景

Use history mode for router:是否启用历史模式，启用

Pre-processors:选择node-sass

Package.json:选择这个

往后一步一步安装完成

运行：npm run serve

编译：npm run build

##### 可视化安装

命令行中输入vue ui



在浏览器中根据提示创建一个项目，选项跟命令行方式一致

#### 添加vue.config.js

提示：vue-cli官网配置，devServer，port和proxy部分

devServer: {

        port: 9000, //端口

        proxy: {    //代理

            '/api': {

                target: 'http://101ycy.com:10001',

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

            }

        }

    }

重新运行一下

## 第四章 代码目录

1. 代码目录

* Dist 项目打包输出目录
* Node\_modules 依赖包文件夹
* Public 公共静态资源文件，绝对路径引用

1. Favicon.ico 网站图标
2. Index.html 页面入口

* Src 源代码主目录

1. Assets 静态资源，相对路径
2. Components 公共组件
3. Router 路由
4. Store vuex状态管理
5. Views 页面
6. App.vue 项目主组件，通过main.js挂载到index.html
7. Main.js 应用程序入口文件，初始化vue实例，引入插件，挂载app.vue主组件

* .browserslistrc 浏览器兼容性设置
* .gitignore git提交忽略设置
* Bable.config.js bable相关设置
* Package-lock.json 安装的包的具体版本号和地址
* Package.json 包引用大版本信息，启动和编译命令
* Postcss.config.js 是一个用 JavaScript 工具和插件转换 CSS 代码的工具，可以百度搜postcss了解

## 第三章 插件的安装

### 说明

npm命令行方式安装和vue ui可视化页面中安装

### Npm使用方式：

#### npm安装模块

【npm install xxx】利用 npm 安装xxx模块到当前命令行所在目录，如：./node\_modules；

【npm install -g xxx】利用npm安装全局模块xxx；

本地安装时将模块写入package.json中：

【npm install xxx】安装但不写入package.json；install简写i

【npm install xxx -save】 安装并写入package.json的”dependencies”中；-save简写-S

【npm install xxx -save-dev】安装并写入package.json的”devDependencies”中。-save-dev简写-D，这里的包只存在开发环境

#### npm 删除模块

【npm uninstall xxx】删除xxx模块；

【npm uninstall -g xxx】删除全局模块xxx；

### 可视化安装

在项目根目录执行vue ui命令，打开页面

点击依赖，安装依赖，搜索依赖选中进行安装

### 命令行方式安装常用插件

#### 安装element-ui

提示：百度搜索elementui

npm i element-ui -S

import ElementUI from 'element-ui';

import 'element-ui/lib/theme-chalk/index.css';

Vue.use(ElementUI);

#### 安装axios

提示：百度搜索axios

npm install --save axios vue-axios

import axios from 'axios'

import VueAxios from 'vue-axios'

Vue.use(VueAxios, axios)

#### 安装vuex-persist

提示：百度搜索vuex-persist

npm install --save vuex-persist

vue.config.js中添加兼容性处理配置

module.exports = {

/\* ... 让bable处理 ... \*/

transpileDependencies: ['vuex-persist']

}

在store/index.js中添加如下配置

import VuexPersistence from 'vuex-persist'

const vuexLocal = new VuexPersistence({

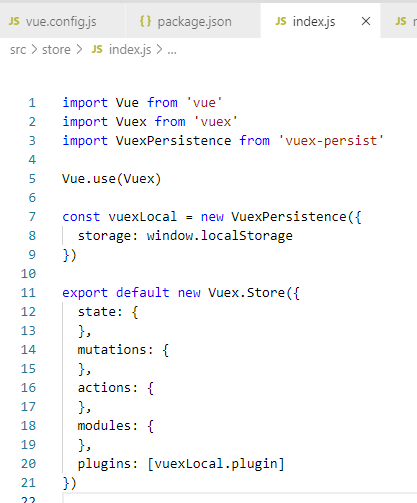
storage: window.localStorage

})

在store实例化中添加如下配置

plugins: [vuexLocal.plugin]

如图所示：



#### 安装vue-i18n

提示：百度搜索vue-i18n

npm install vue-i18n -S

import VueI18n from 'vue-i18n'

Vue.use(VueI18n)

实例化i18n配置到vue中（代码中演示，并演示模块化过程）

#### 安装moment

百度搜索：moment

npm install moment --save

window.moment = require('moment')

#### 安装ckeditor5

提示：百度搜索ckeditor5

npm install --save @ckeditor/ckeditor5-build-decoupled-document

npm i @ckeditor/ckeditor5-vue -S

引入提示：官网查看documentation，vue.js component

import CKEditor from '@ckeditor/ckeditor5-vue';

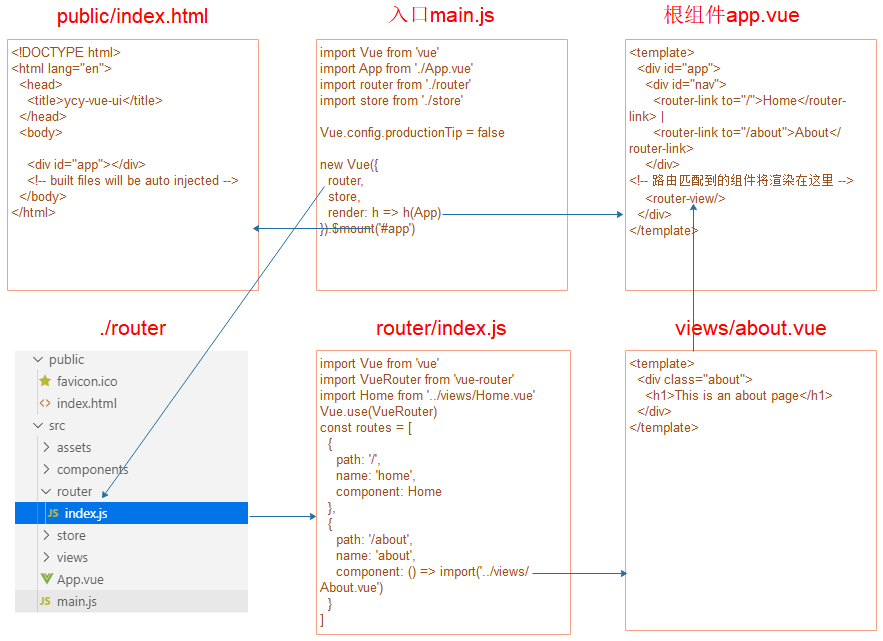
Vue.use( CKEditor );

## 第五章 页面加载渲染

### 说明

程序运行浏览器中将会打开页面，那么程序进行怎么样的过程加载渲染页面呢，这一章讲解程序加载中相关的文件和其调用关系及过程，让大家理解页面是如何渲染的。

### 页面加载图示



### 页面加载流程描述

入口文件main.js中引入路由配置index.js、根组件app.vue，通过new Vue实例化将app.vue挂载到宿主页面index.html中，路由index.js中配置了路由path对应的views文件夹下的页面组件，当浏览器地址为对应路由path的时候，将会把对应的页面组件渲染到根组件app.vue中的<router-view>处，从而渲染出整个页面。我们开发的页面其实都是在根组件app.vue的<route-view>中来渲染的，当然页面中还可以嵌套子<route-view>实现路由嵌套。

到此，大家应该懂得了main.js、index.html、app.vue、router/index.js、views/页面组件的关系了

### 添加一个新页面

Demo:views/5.4/New.vue

第一步，在views文件夹下添加new.vue组件

第二步，在route/index.js中添加new页面的路由配置

第三部，浏览器输入new页面path查看

## 第六章 插件的使用

### Vue的使用

怎么去创建页面，怎么去写HTML，CSS和JS，以及vue的基本用法

提示：百度vue看官网

Demo:views/6.1/New2.vue

1. 创建vue页面，讲解三部分
2. 数据绑定，表单绑定、差值表达式
3. 按钮事件，读取实体变量
4. 双向数据绑定：表单可以驱动数据改变，数据可以驱动dom元素改变（表单元素或页面元素）
5. 列表渲染：ul,li
6. table表格渲染
7. V-if和v-show
8. 子组件的使用和传参

讲解home.vue对helloword的使用和传参

1. 父子组件间的传值和相互调用

Demo:views/6.1/parent.vue和child.vue

新建parent.vue和child.vue

1. 演示引用：父组件引用子组件
2. 父传子参数：字符串传参，数据流向，单向父传子；对象传参，数据流向双向，但不建议修改子组件props中参数值
3. 子调用父组件方法this.$emit
4. 父组件调用子组件this.$refs
5. 父组件向子组件传递html：插槽slot

Demo:views/6.1/child2和child3

1. 子组件间调用：bus事件总线

Demo:views/6.1/lifecycle.vue

1. 组件生命周期

Create:在页面渲染之前调用

Mouted:在页面渲染之后调用，建议把页面初始化查询方法放在这里

### Element-ui的使用

主要讲基本控件的使用

Demo:views/6.2/basecontrol.vue

1. 文本框
2. 按钮
3. 单选
4. 多选
5. 下拉框
6. 日期
7. 表格

### axios的使用

提示：百度axios

Demo:views/6.3/axiosdemo.vue

1. POST传参

RequestBody：参数是实体对象，Content-Type: application/json;

RequestParam：简单参数，Content-Type: application/x-www-form-urlencoded;

RequestParam方式有两种传参方式：URLSearchParams和qs

1. GET传参

只有RequestParam方式

1. 调用例子

this.$http.post(url,param)

this.$http.get(url,{params:param})

1. 超时设置

全局设置，和单次调用设置

1. 拦截器

Demo:/router/index.js

跟路由守卫放在一起，都属于拦截类处理

请求拦截器：发起请求会拦截，可以设置请求头加token

响应拦截器：接口返回会拦截，可以处理全局异常

1. 接口封装

在apis文件夹下写接口文件，并汇总到apis.js中，页面中只需要引用apis.js，调用相关接口方法即可

Demo：views/6.3/AxiosApiDemo.vue

### vue-router的使用

百度：vue-router

Demo:views/6.4/routedemo.vue

1. 添加根路由

同步方式：全部加载

异步方式：请求的时候才单独加载，路由懒加载

1. 路由跳转

Url:网址中输入跳转

Route-link：渲染成a标签跳转

Redirect:路由重定向，一般把/重定向到/login

this.$router.push('/xxx')：跳转可以返回

this.$router.replace('/xxx')：跳转不可返回

this.$router.go(-1)：返回上一页

1. 路由传参

Params传参：必须路由name调用，刷新丢失

Query传参：路由name和path都可以，参数出现在浏览器地址栏，刷新不会丢失（推荐）

1. 读取路由信息

This.$route下

fullPath：路由完整地址

name：路由名称

path：路由path

meta:路由元信息

params：params参数信息

query：query参数信息

1. 路由嵌套

在父页中局部切换子页，例如左侧导航，右侧切换内容

路由中配置children，父页中配置<router-view/>

1. 导航守卫

全局前置守卫

## 第七章 模块划分

## 第八章 模块开发

## 第九章

## 第十章

## 第十一章