

上海海洋大学2025项目实训

前后端分离开发

前端(app):

VSCode工具：前端代码编辑工具。 免费，官网下载。 【自己安装】

开发环境：

nvm工具：下载和管理**Node**软件。 **node**就是前端框架运行的环境。

nvm作用就是可以在电脑上安装不同版本**node**，随意切换。

技术：**Vue2**、**Vue3**框架。

vue的版本不同，所需要**node**版本也不同。

运行：

浏览器 **Chrome** 安装插件。

Edge 安装插件（微软在线商店）。

后端（服务器程序）：

Idea2023版本 【自己安装破解】

开发环境：

JDK8

maven3.8.8 管理项目和项目中依赖(jar)。

数据库：

MySQL5 或 **MySQL8** 【自己安装，**root**用户密码记住】

Navicat Premium 17 数据库可视化的工具（破解）。

测试：

ApiFox 【自己安装】

nvm工具使用

作用：**nvm**可以帮助我们安装和管理**node**软件。

安装

下载**nvm**工具。安装 **D:\tools\nvm** 目录。

安装完毕实际路径：**D:\tools\nvm\nvm**

配置

步骤1:

首先在**D:\tools** 路径创建一个**nodejs**文件夹。指定**nvm**工具安装**node**到该文件夹下面。

创建完毕目录：**D:\tools\nodejs**

步骤2:

查看**nvm**环境变量（正常情况安装**nvm**自动配置环境变量）

NVM_HOME: **D:\tools\nvm** **nvm**安装的根目录

NVM_SYMLINK: **D:\tools\nodejs**

步骤3:

修改**nvm**软件配置参数，打开**nvm**安装目录，打开**settings.txt**文件，配置以下内容：

root:**D:\tools\nvm**

path:**D:\tools\nodejs**

node_mirror:<https://npmmirror.com/mirrors/node/>

```
npm_mirror:https://npmmirror.com/mirrors/npm/
```

步骤4

测试配置是否成功

启动Cmd终端，视窗+S 搜索cmd，以管理员的身份打开：

输入命令测试：

```
nvm version    --查询nvm版本
1.1.10
```

nvm安装NodeJS

如果之前电脑安装过NodeJS，现在想使用nvm，就必须先卸载之前安装nodeJS

--查看nvm镜像服务器上有哪些NodeJS版本。

```
>nvm list available
```

--安装指定版本nodeJS

```
>nvm install 18.19.0
```

--查看电脑已经安装的版本

```
>nvm list
```

--切换版本命令

```
>nvm use 18.19.0
```

--卸载的命令

```
>nvm uninstall 18.19.0
```

node和npm配置

node前端程序运行环境，node安装成功自带一个npm工具。

npm前端项目构建、相关插件的依赖库的工具。

使用命令查看 node版本。

```
>node -v
v18.19.0
```

node工具安装完毕，自动在node内部安装一个npm工具。

--查看当前npm版本

```
>npm -v
10.2.3
```

--查看npm默认的配置信息

```
>npm config list
```

配置npm信息

配置信息 实质在上 C:\Users\byterain\.npmrc 文件中存储。

所以，如果配置错误，可以将该文件删除，重新配置。

步骤1:

设置npm文件

在D:\tools\nvm\v18.19.0目录下，创建2个文件夹：

npm_cache 安装插件时所需要的缓存文件夹

npm_modules 安装插件所存放的目录

步骤2:

启动cmd终端, 配置新建的2个文件夹到npm。

```
npm config set prefix "D:\tools\nvm\v18.19.0\npm_modules"
```

```
npm config set cache "D:\tools\nvm\v18.19.0\npm_cache"
```

步骤3:

设置npm下载的服务器地址为国内的镜像地址

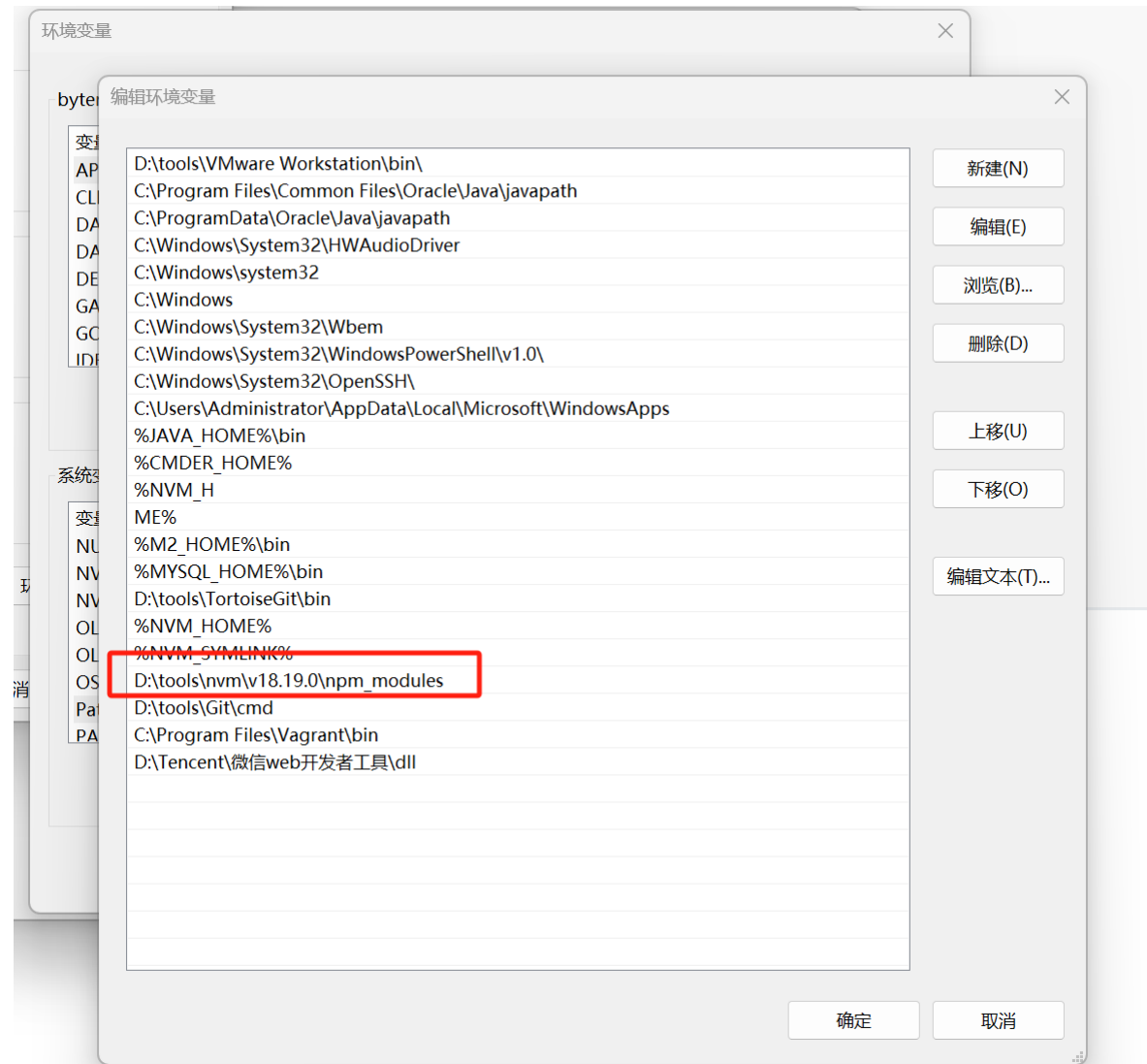
```
npm config set registry http://registry.npmmirror.com
```

步骤4:

```
npm config list
```

配置npm环境变量

添加 D:\tools\nvm\v18.19.0\npm_modules 到系统Path环境变量中。



使用npm创建vue项目

安装插件 脚手架(快速搭建Vue项目结构)。

-- npm安装插件命令

```
npm install 插件@版本
```

-- 安装vue脚手架插件

```
>npm install -g @vue/cli
```

```
--测试查看脚手架插件是否安装成功
>vue -V      --大写V
@vue/cli 5.0.8
```

创建Vue项目

切换到项目存储目录，使用命令创建vue项目

```
vue create 项目名（全部小写）
```

使用Vscode编辑器打开项目，在VScode中启动终端：Ctrl + `

```
-- 启动项目
D:\VueProjects\mall_vue_021> npm run serve
-- 注意：必须在项目文件夹路径下执行
```

终止终端服务运行

快捷键 Ctrl+c 输入y

Vue项目结构

```
mall_vue_021
|- node_modules 项目运行所需的插件库
|- public
|  |- favicon.ico 浏览器选项卡上图标
|  |- index.html  首页（单页面应用）
|- src  代码文件夹
|  |- assets  静态资源(图片)
|  |- components 公共组件文件夹
|  |- router  路由( 切换页面 )
|  |- views  所有视图界面( 视图组件)
|  |- App.vue  主组件
|  |- main.js  入口文件(项目代码从入口文件开始执行)
|- package.json  拆解库的配置文件,内部所有的配置数据都是以json形式配置
|- package-lock.json  定义各插件最高版本上限
|- vue.config.js  项目的配置文件，修改端口号，跨域。
```

Vue项目中 所有的组件文件后缀名都是.vue文件。

json对象

Javascript代码定义一个对象（数据的集合体 -- 变量和方法）。

```
let obj = { 属性:值,属性:值,属性:值 };
对象的属性值 可以是JS中任意类型： 数字、字符串、逻辑类型、方法(函数)、对象、数组...

<script>
  let stu = {
    name:"byterain",
    age:22,
    phone:'13388990099',
    getNameLength:function(){
      return this.name.length;
    }
  }
</script>
```

```
    },  
    address: {  
      code: '030006',  
      info: '上海市静安区南京西路1号'  
    },  
    language: ["Java", "C", "C++"]  
  };  
};
```

```
console.log(stu.name);  
console.log(stu.getNameLength());  
console.log(stu.address.info);  
console.log(stu.language);
```

```
</script>
```