

前端基础环境搭建

基础环境

Node.js

你需要先清楚的几个术语

- CURRENT：指代最新的 Node.js 版本系列（单数）。
- Active：指正在积极维护和升级的版本系列，包括向后移植非破坏性功能和改进，解决错误以及修补安全漏洞。
- Maintenance：这是一个维护的 LTS 版本系列，直到它的生命周期终止，只会在短时间内收到错误修复和安全补丁。
- LTS：是 Long-Term Support 的缩写，代表 Node.js 长期支持的版本（版本号为复数）。
- EOL：EOL 是 End of Life 的首字母缩写，进入到 EOL 时间线的版本，将不在维护。

我们用什么版本

- 一般来说,Node.js 每年 10 月发布奇数(9, 11...)版本, 次年 4 月发布偶数(10, 12...)版本。
- 中心开始使用 Vue 技术栈时间是在 2019 年中,此时对应的稳定大版本为 12,因此建议统一使用 12.16.0 版本。
- 截至2022年,node14版本已稳定运行,目前推荐14.18.0版本

如何安装

- [直接访问 Node.js 淘宝镜像地址进行下载](#),根据自己电脑选择.msi 安装包或者压缩文件。
- 通过 nvm 安装(**推荐**)

Nvm

- [Nvm 安装](#)
- Nvm 设置淘宝镜像

```
nvm node_mirror https://npm.taobao.org/mirrors/node/ #设置node镜像
nvm npm_mirror https://npm.taobao.org/mirrors/npm/ #设置npm镜像
```

- Nvm 使用

```
nvm -v          #查看当前nvm版本
nvm ls          #查看当前已安装的Node版本
nvm install latest #安装Node最新稳定版本
nvm install 10.16.0 #安装Node指定版本
nvm use 12.18.0   #切换至指定Node版本
nvm uninstall 12.18.0 #卸载指定Node版本
```

Npm

什么是 Npm

- NPM 是 Node.js 中的包管理器。允许我们为 Node.js 安装各种模块，这个包管理器为我们提供了安装、删除等其它命令来管理模块。这里有一点我们需要注意，我们必须要有个 package.json 文件或 node_modules 目录安装模块到本地。
- NPM 最好的一点是它会在本地存储我们所安装的依赖项，存在于 package.json 的 dependencies 对象里。例如，如果一个模块 X 使用了模块 A 版本为 1.0，模块 Y 使用了模块 A 版本为 1.5，那么模块 X 或 Y 都将在本地拥有自己对应的模块 A 的副本。

Npm 安装

NPM 不需要单独安装，在我们安装 Node.js 环境时，NPM 也就安装了

终端执行 `npm -v` 命令查看当前 npm 版本

```
npm -v
6.9.0
```

Npm 使用

```
npm init      // 初始化项目,将自动生成packag.json文件
npm install   // 读取package.json文件,安装全部依赖模块
npm install Vue // 安装单个 模块 npm5.0以后,默认自带-S
npm install Vue -D // 安装模块 仅开发环境,正式发布不会打包此类依赖
npm uninstall Vue // 卸载模块
npm update Vue // 升级模块 谨慎升级!可能出现版本不兼容
```

Nrm

什么是 Nrm

nrm 是一个 npm 源管理器, 允许你快速地在 npm源间切换。

什么时候需要Nrm

Nrm可以用来快速切换npm镜像源,中心自用的私服不太稳定,有时候安装容易出问题,但是xx-ui,xx-chat,xx-formmaking等重要组件都在中心私有npm服务器上,当然我们可以在install的时候直接添加--registry参数,但是通过Nrm来进行快速切换更为便捷

安装Nrm

```
#必须要完成node环境再执行nrm安装
npm i -g nrm
#等待执行完成即可
```

如何使用

```
#查看当前所有镜像源
nrm ls
```

```
$ nrm ls

* npm ---- https://registry.npmjs.org/
cnpm --- http://r.cnpmjs.org/
taobao - https://registry.npm.taobao.org/
nj ----- https://registry.nodejitsu.com/
rednpm - http://registry.mirror.cqupt.edu.cn/
npmMirror https://skimdb.npmjs.com/registry/
edunpm - http://registry.enpmjs.org/
```

目前基础研发部搭建了中心自用的npm仓库,镜像地址为<http://10.14.2.150:19000/repository/npm-public/>

如需添加这个镜像源,可直接执行

```
nrm add xxpm(本地注册的镜像名称) http://10.14.2.150:19000/repository/npm-public/
```

```
PS C:\Windows\system32> nrm add xxpm http://10.14.2.150:19000/repository/npm-public/
add registry xxpm success
PS C:\Windows\system32> nrm ls

npm ----- https://registry.npmjs.org/
yarn ----- https://registry.yarnpkg.com/
tencent ----- https://mirrors.cloud.tencent.com/npm/
cnpm ----- https://r.cnpmjs.org/
taobao ----- https://registry.npmmirror.com/
npmMirror --- https://skimdb.npmjs.com/registry/
xxpm ----- http://10.14.2.150:19000/repository/npm-public/
```

切换镜像源的时候直接使用nrm use 镜像名称即可

```
nrm use npm
```

Yarn

什么是 Yarn

Facebook 贡献的 Javascript 包管理器。可以看做是 Npm 的替代品,目前 Yarn 的与 Npm 共享所有依赖包,Yarn 由于其独特的缓存机制,所以再次使用时无需重复下载。同时利用并行下载以最大化资源利用率,因此安装速度更快。

安装 Yarn

[直接访问 Yarn 中文官方网站进行下载](#)

使用 Yarn

```
yarn init          // 初始化项目,将自动生成packag.json文件
yarn install        // 读取package.json文件,安装全部依赖模块 直接执行yarn也可以
yarn add Vue        // 安装模块 npm5.0以后,默认自带-S
yarn add Vue -D      // 安装模块 仅开发环境,正式发布不会打包此类依赖
yarn remove Vue     // 卸载模块
yarn upgrade Vue     // 升级模块 谨慎升级!可能出现版本不兼容
```

更换镜像源

在国内有时候受限于网络因素的影响,通常在安装一个包管理器之前可以切换为 淘宝源,是的速度可以更快,但是要注意如果是私有模块在 NPM 官方的,则必须切换为官方源,否则会出现 404 错误。

```
# npm设置为淘宝镜像
npm config set registry https://registry.npm.taobao.org/
# npm设置为官方镜像
npm config set registry https://registry.npmjs.org/
# yarn设置为淘宝镜像
yarn config set registry https://registry.npm.taobao.org/
# yarn设置为淘宝镜像
yarn config set registry https://registry.yarnpkg.com
```

(相对于切换为 淘宝 源,网上有更方便的解决方案,即直接安装 cnpm,但是据以往经验,cnpm 有时候会出现奇怪的路径错误,大家可以自行安装,但是建议手动设置为 淘宝源)

Git

- [Git 安装](#)
- 当前 Vue 技术栈,中心使用 Gitlab 进行源代码管理,因此 git 的使用就变得必然了!

```
git status          #查看状态
git add .           #将修改的东西,提交至修改树,.表示全部,可以用xxx.js进行单独提交
git commit -m "改动说明" #将工作区的代码 提交至 本地仓库
git pull            #从代码服务器更新至本地(可能会存在冲突)-解决冲突如果有冲突,在界面里面解决冲突之后,在git
merge
git push            #将代码更新至代码服务器
```

- 此处~~墙裂~~推荐VSCode的GitLens插件,方便你迅速找打让你头疼欲裂的罪魁祸首并迅速喷之!!
- 此外还推荐使用Github官方出的UI工具,[Github Desktop](#)

编辑器

VSCode 扩展插件

在 VSCode 中按下 Ctrl+Shift+X 进入扩展商店,直接搜索以下插件:

- **Chinese (Simplified) Language Pack for Visual Studio Code**
VSCode 汉化包
- **Vetur**
Vue 辅助插件,提供 emmit 模板,vue 常用语法自动补充,vue 代码高亮等功能;
- **vue-helper**
Vue 辅助插件,强力支撑ElementUI;
- **Prettier – Code formatter**
Prettier 代码格式化工具,有针对 vue,scss 等文件的格式化,保证代码可读性,美观性;
如果有安装其他代码格式化工具,需要将VSCode的格式化方案指定为Prettier
 - VSCode修改格式化方案:

1. 在VSCode中按下Ctrl+Shift+P;
2. 在弹出的命令行中输入format;
3. 在选项中选择prettier;

- 如果vue文件无法正确格式化,请检查Prettier对Vetur的支持

1. 点击VSCode左上角,文件 > 首选项 > 设置;
2. 在设置页面的搜索框中输入prettier;
3. 在搜索列表中找到vetur的大块,确认需要对应功能的方案为prettier;

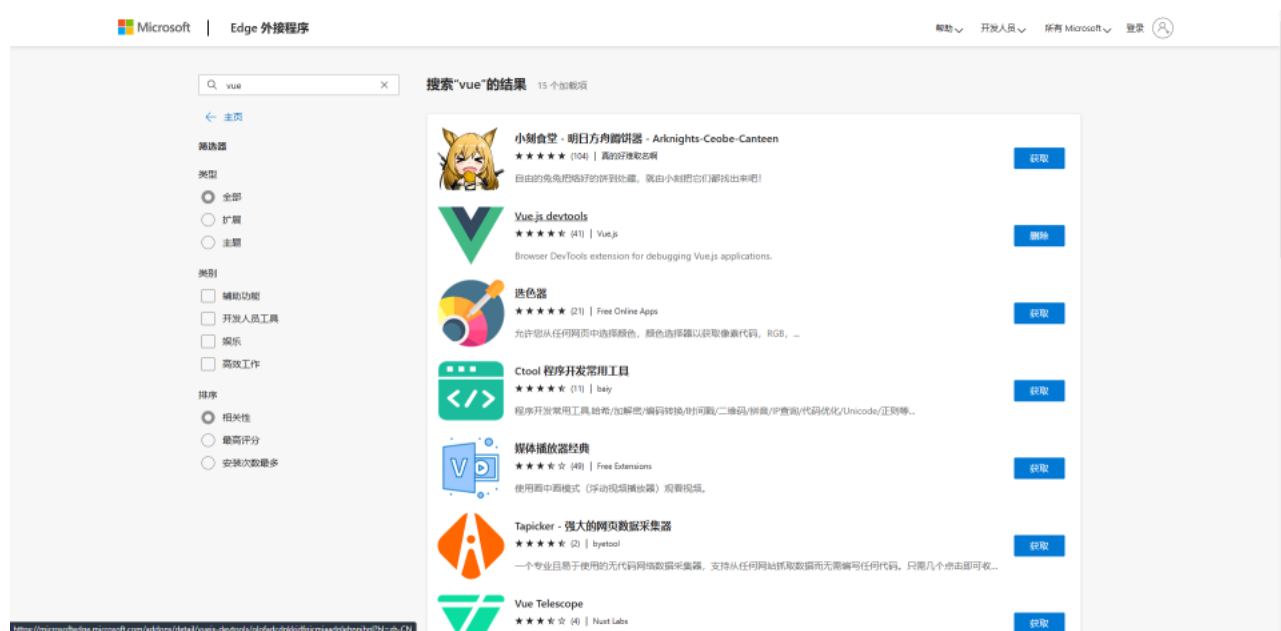
- **EditorConfig for VS Code**
editorconfig 编辑器风格统一工具,统一代码缩进方式,缩进量,换行风格;
- **ESLint**
ESLint 代码检查工具,对于部分声明错误及未定义变量检查有极好效果,配合 prettier 可统一团队代码;
- **GitLens**
GitLens — Git supercharged 提供git版本追溯能力
- **欢迎推荐及补充**

WebStorm 扩展插件

浏览器

Edge、Edge 扩展插件

Edge浏览器扩展



- **Vue 扩展 vue devtools**
Vue 扩展插件,方便快速定位 component,查看数据

- **分辨率调节 Browser Resizer**
快速切换浏览器分辨率,方便进行适配
- **JSON 格式化 JSON Formatter for Edge**
快速格式化 JSON 数据,用于浏览器直接访问数据接口

其他工具

PostMan 接口测试工具

前后端分离的开发模式,接口测试往往必不可少
[PostMan 下载](#)