# 前端项目配置启动

### **前言**

由于前端开发是采用node来进行服务启动，所以我们电脑上就要对node来进行配置以及的其他的配置。当我们拉取前端项目之后如何快速区分识别并能正常启动。

关于node的版本安装一般可以采用以安装node的时间为基准，安装最新的正式版本 正常16版本以上就可以

### window node安装教程

多前端项目由于创建项目时期不同有可能会存在node版本不一致的问题，所以我们就要支持多node版本切换，我们将采用nvm来进行node版本管理。

当然如果你已经安装过node但并不是使用nvm来进行管理，也是可以不用删除node重新安装nvm来进行管理的，只有当多个前端项目明确规定了node版本时并且版本不一致的时候，才会使用到nvm来进行node的多版本管理。

#### nvm安装 [nvm-setup.exe](/download/attachments/721097/nvm-setup.exe?version=1&modificationDate=1690727542000&api=v2)

**安装时会有两个路径需要注意，一个是nvm安装路径和node安装路径。需要特别注意文件夹名不要出现**中文**和**空格

安装完成后输入nvm version命令来检查是否安装成功。成功会显示以下信息。

在安装完成之后就可以来进行node的安装，但在此之前需要配置一下nvm的下载镜像源

#### 修改下载镜像源

由于nvm默认下载node的镜像源很慢所以需要替换成淘宝镜像源

找到nvm安装路径 => 找到 setting.txt 文件 =>新增以下两行信息，配置下载源

node\_mirror: <https://npm.taobao.org/mirrors/node/>  
npm\_mirror: <https://npm.taobao.org/mirrors/npm/>

**node安装** 有可能会提示权限不足，使用超级管理员安装

nvm install 14.5.0 // 安装14.5.0版本node

nvm install latest // 安装最新版本node

到这里node就安装完成了。可以使用  node -v 和 npm -v来查看node是否安装成功

#### nvm的常用命令

nvm ls // 显示已安装的版本（同 nvm list installed）

nvm list installed // 显示已安装的版本

nvm list available // 显示所有可以下载的版本  
nvm use 14.5.0 // 使用14.5.0版本node 切换node版本

以上步骤执行完毕之后，node的安装就完成了,下面就要进行npm私服地址的配置了

### npm私服地址配置

为什么配置私服地址，我们使用的@hbasesoft/web-plugin组件是部署在公司服务器内的所以需要配置私服地址才能安装

##### nrm   npm镜像管理包

使用npm全局安装nrm

npm install nrm -g

添加私服镜像地址到本地

nrm add hbs <http://nexus.hbasesoft.com:8888/repository/npm-public/>

切换本地镜像致私服地址

nrm use hbs

查看镜像源地址

nrm ls

至此私服地址的配置已经OK了

#### npm yarn pnpm的区分

npm,yarn,pnpm都是前端包管理器，首先全局包里要存在以上三种包管理器才能使用对应的管理器

npm由于是node的默认包管理器，所以当node安装好以后全局就会存在npm包管理器了

#### 如何全局安装 yarn和pnpm

npm install yarn -g

npm install pnpm -g

### 如何区分项目应该使用何种管理器来进行安装前端依赖

当我们拉取前端文件后首先检查前端目录下是否存在以下几个文件

1. package-lock.json 使用的是npm来进行的项目依赖包管理
2. yarn.lock 使用的是yarn来进行的项目依赖包管理
3. pnpm-lock.yaml 使用的是pnpm来进行的项目依赖包管理

上面的三种文件都是项目所依赖包的描述文件，可以理解为前端项目所依赖的所有包的版本描述

如果存在以上文件中的任何一种请使用相对应的包管理器来安装项目依赖。项目依赖的安装的命令是 npm install或则yarn install或则pnpm install。请在package.json文件层级下进行安装

**当我们使用对应的包管理来安装项目依赖后，是不会修改以上三种文件的，如果修改了就说明安装项目依赖出现了问题，寻找前端同事来进行排查**

如果上诉文件都不存在时，请寻找前端同事来确定使用何种管理器来进行安装

**需要特别注意的是上述三种文件在不修改项目依赖的情况下是不允许修改的**

当确定管理器之后并且成功安装项目依赖后就可以正常启动前端项目了

### 关于私服中@hbasesoft/web-plugin组件的升级

由于@hbasesoft/web-plugin是我们自己维护的前端组件，我们将不定期更新@hbasesoft/web-plugin，当我们升级@hbasesoft/web-plugin之后需要针对项目的@hbasesoft/web-plugin来进行升级

1. 拉取最新项目，package.json文件中会有最新的@hbasesoft/web-plugin版本
2. 重新执行 npm install或则yarn install或则pnpm install

何时进行升级

当@hbasesoft/web-plugin升级之后，前端同事会进行信息同步并且修改项目中@hbasesoft/web-plugin的版本

### 关于@hbasesoft/web-plugin

@hbasesoft/web-plugin是为了更高效的开发PC端页面所开发的组件库，配合元数据模型可以快速的生成一个页面具体的使用方法请见：<http://docs.hbasesoft.com:8888/>。此文档会持续更新。拓展文档：[procomponents.ant.design](https://procomponents.ant.design/)，[ant.design/docs/react/introduce-cn/](https://ant.design/docs/react/introduce-cn/)

### 插件版本介绍

package.json中dependencies中各个插件版本都带了^标识还有一种是~标识。

^：匹配最近的大版本，比如^1.0.2，将匹配1.x.x，不会匹配2.x.x

~：匹配最近的小版本，比如~1.0.2，将匹配1.0.x，但不会匹配1.1.x

通俗一点讲安装的版本表现如果为^1.1.1，那么过段时间再去安装就有可能安装成1.2.0，如果插件更新了版本，会自动更新版本的。这也会产生一个问题如果没有版本锁定文件的话过段时间安装就会出现和项目初始化时安装的版本不一致的问题。那么多人协作开发时由于安装依赖的时间不一致就会导致安装下来的版本也不一定一致，最终打包出的文件也有出入。

### FAQ

#### 关于多人协作开发时，编译出的文件内容不一致。

由于项目没有版本锁定文件，那么每一个人安装依赖的版本都有可能不一致。那么最终打包时也会出现偏差。

解决方案：由一个人重新安装项目依赖，并且使用公司内部的私服代理来进行安装，会产生一个.lock文件，提交此文件后续人员再按照此文件进行安装。