Contents

[Nest JS 1](#_Toc35787834)

[React JS 3](#_Toc35787835)

[React Native 4](#_Toc35787836)

[私有仓库搭建和使用 4](#_Toc35787837)

[自定义React组件 4](#_Toc35787838)

# Nest JS

VSCode Debug Nest JS

[https://blog.csdn.net/weixin\_34129145/article/details/91442053](https://nam01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fblog.csdn.net%2Fweixin_34129145%2Farticle%2Fdetails%2F91442053&data=02%7C01%7Cxiaohuhu%40ebay.com%7C20119a8b07934faeba9308d7cce9a0b7%7C46326bff992841a0baca17c16c94ea99%7C0%7C0%7C637203176116343973&sdata=C5HXtYukAqHIutuPPaiLM2P56R3zH8MVAvAFwDu4TxM%3D&reserved=0)

NodeJS下载安装

<https://nodejs.org/en/download/>

下载安装Git

<https://git-scm.com/download/win>

国内下载慢，可以用以下链接

<https://pc.qq.com/detail/13/detail_22693.html>

国内下载慢，需要设置镜像

npm config set registry <https://registry.npm.taobao.org>

npm i -g @nestjs/cli

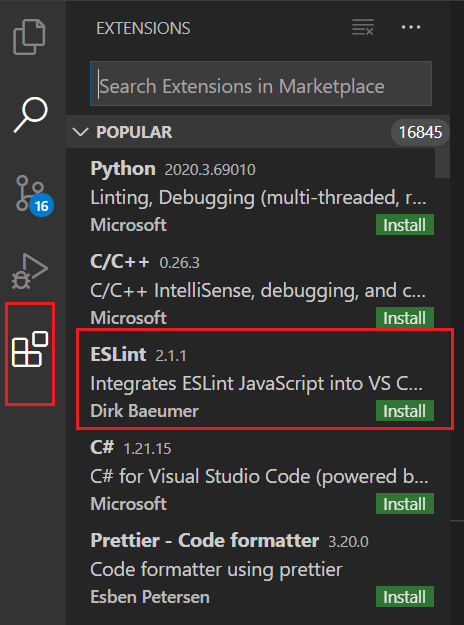
cd到指定路径

nest new project-name

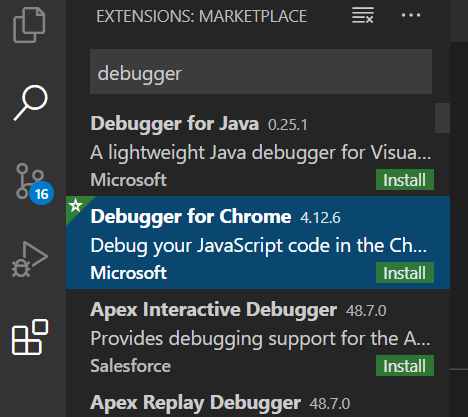
Cd project-name

Code .

在左侧找到Extension -> 安装ESLint



以及Debugger for Chrome



打开终端控制台，输入：

Npm install

//以下start命令可以直接运行

Npm run start

也可以使用VS进行Debug。 两种方法：

1. 添加launch.json

<https://www.jianshu.com/p/43668a35903a>

使用以下覆盖

{

    "version": "0.2.0",

    "configurations": [

        {

            "type": "node",

            "request": "launch",

            "name": "Debug Nest Framework",

            "args": ["${workspaceFolder}/src/main.ts"],

            "runtimeArgs": ["--nolazy", "-r", "ts-node/register"],

            "sourceMaps": true,

            "cwd": "${workspaceRoot}",

            "protocol": "inspector",

            "console": "integratedTerminal"

        }

    ]

}

然后F5，加断点，访问<http://localhost:3000/>

2）命令 npm run start:dev

<https://blog.csdn.net/weixin_34129145/article/details/91442053>

# React JS

npm install -g create-react-app

create-react-app reactdemo

cd reactdemo

npm start

# React Native

# 私有仓库搭建和使用

更新Node JS到最新版本

npm install -g npm

<https://www.jianshu.com/p/cf2e9f580e6d>

安装Python

<https://www.python.org/downloads/release/python-2715/>

环境变量设置

比如，安装路径：C:\Python27\

就在环境变量的Path中添加上述值

以下命令测试Python安装是否正确 （安装后需重启）

python -v

安装verdaccio （需先安装Python）

npm install –global verdaccio

安装后，输入命令：verdaccio 即可运行

# 自定义React组件

<https://www.jianshu.com/p/db6113c94dbc>

mkdir myReactComponent

cd myReactComponent

npm init -y

对package.json做如下修改：

"main": "dist/bundle.js",

"files": ["dist"],

"scripts": {

"start": "webpack-dev-server --config webpack.dev.config.js",

"dev": "webpack-dev-server --config webpack.dev.config.js",

"build": "webpack --config webpack.prod.config.js"

},

 main: 这里是我们组件的入口文件。开发者在 import 我们的组件的时候会引入这里 export 的内容

 files: 申明将要发布到 npm 的文件。如果省略掉这一项，所有文件包括源代码会被一起上传到 npm

 scripts: 申明命令行可用的各种指令。

安装依赖项

npm i react react-dom

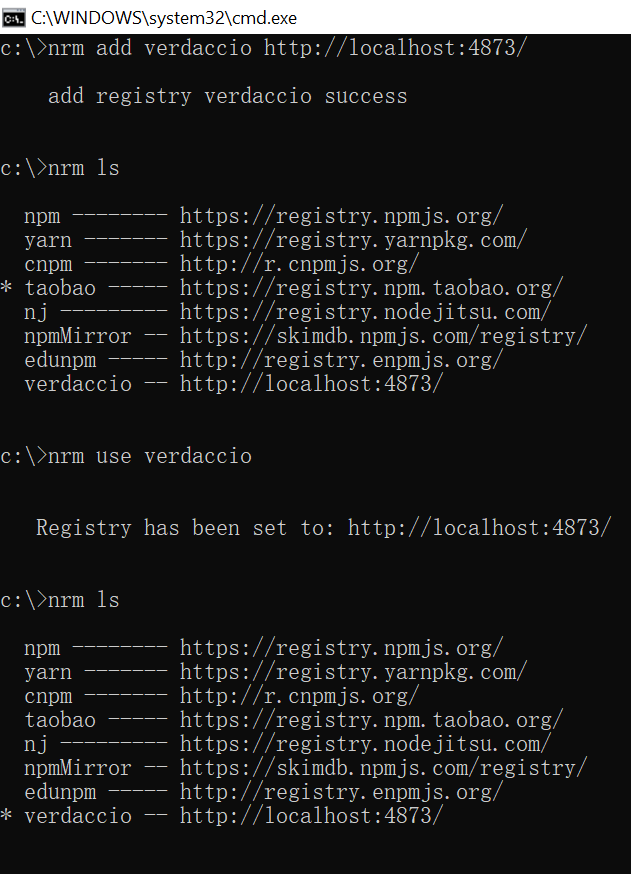
npm i -D babel-loader @babel/core @babel/preset-env @babel/preset-react webpack webpack-dev-server webpack-cli html-webpack-plugin webpack-node-externals css-loader style-loader

npm install -g nrm

nrm add verdaccio <http://localhost:4873/>

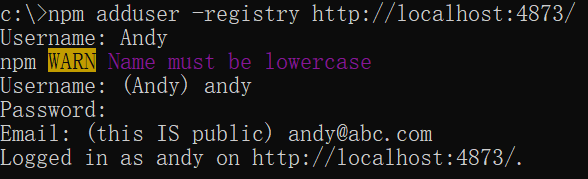
使用nrm ls 指令可以查看所有注册的仓库

nrm use verdaccio

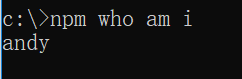


npm adduser –registry <http://localhost:4873/>

输入用户名密码，andy，123456



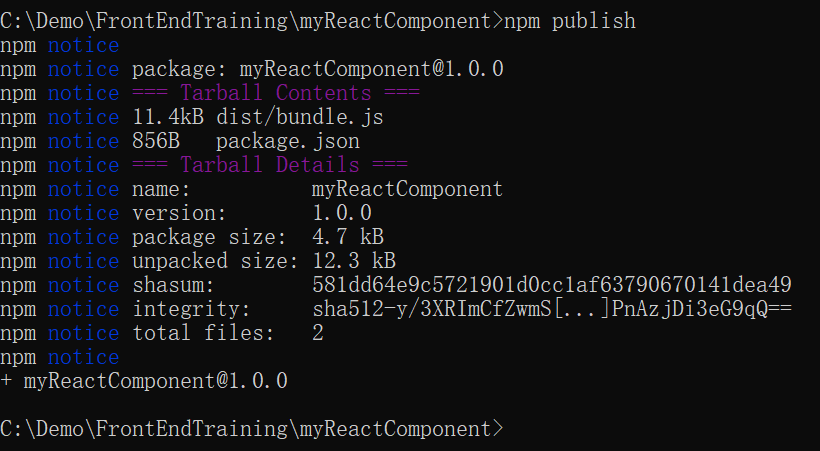
Npm who am I



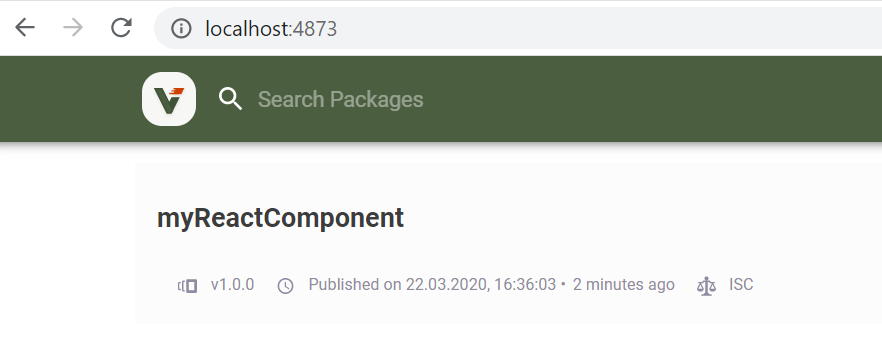
在package.json文件夹路径下执行 npm publish

会自动寻找main节点的配置做publish

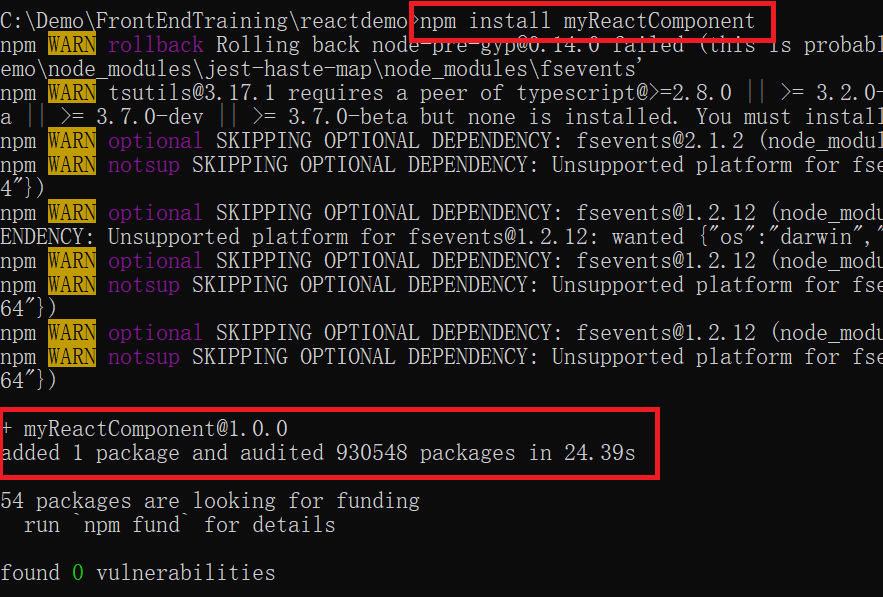




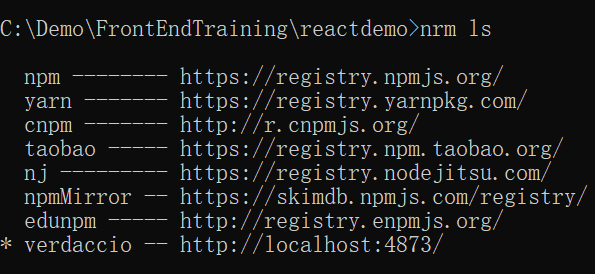
可以在私有仓库中看到上传的组件



可以在其他项目中复用该组件



结束之后别忘了使用nrm use命令把repository换回来(国内用taobao)

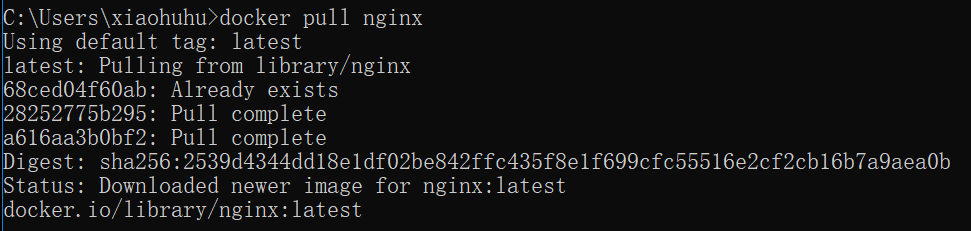


# React App部署到Docker Container

<https://blog.csdn.net/Alexia23/article/details/100700512>

拉取Ningx镜像

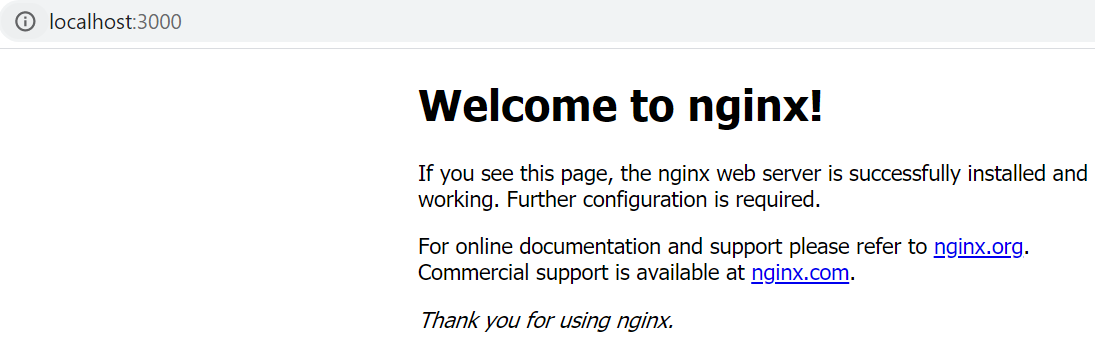
docker pull nginx



将容器端口80映射到主机端口3000

docker run -p 3000:80 nginx

浏览器访问localhost:3000



Npm run build 编译React JS项目

在React JS项目的根目录下执行：

docker cp ./build/. 7b255166965b:/usr/share/nginx/html

7b255166965b是容器ID，该命令会将build文件夹下的所有文件都copy到容器的指定文件夹下

再访问localhost:3000 就是我们的React App了

使用以下命令可以查看Container中的部署情况

docker exec -it a96f99630aed /bin/sh

cd /usr/share/nginx/html

ls