Create React App教程

<https://create-react-app.dev/docs/installing-a-dependency>

<https://dev.to/dabit3/building-micro-frontends-with-react-vue-and-single-spa-52op>

1：新建文件夹：FrontEndMicroService

2：cd到该文件夹

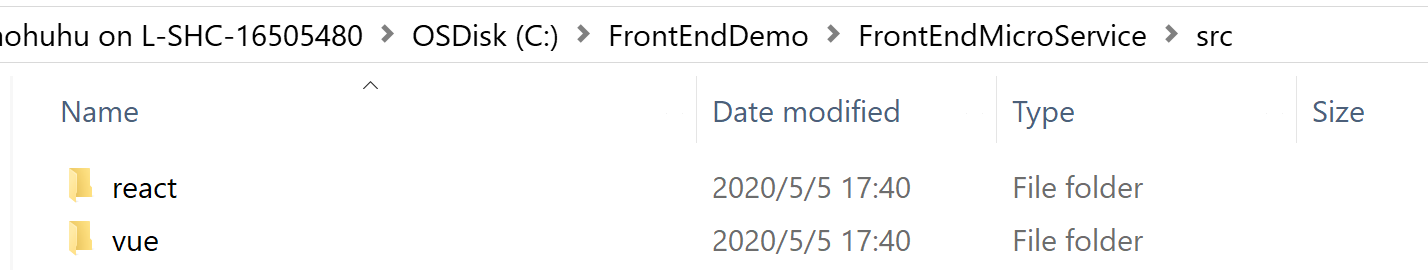
3：npm init -y

4：npm install react react-dom single-spa single-spa-react single-spa-vue vue

5：npm install @babel/core @babel/plugin-proposal-object-rest-spread @babel/plugin-syntax-dynamic-import @babel/preset-env @babel/preset-react babel-loader --save-dev

6：npm install webpack webpack-cli webpack-dev-server clean-webpack-plugin css-loader html-loader style-loader vue-loader vue-template-compiler --save-dev

7：建立src文件夹，再其下分别建react、vue两个文件夹



8：回到项目根目录，创建：.babelrc文件， 注意：文件名要用：.babelrc.

否则会提示无法创建

<https://single-spa.js.org/docs/examples/>

使用CLI：<https://single-spa.js.org/docs/create-single-spa/>

2020/8/21

<https://www.robinwieruch.de/react-micro-frontend>

React中使用SVG

<https://blog.csdn.net/Enl0ve/article/details/88752841>

<https://segmentfault.com/a/1190000021044479>

<https://alili.tech/archive/11052bf4/>

<https://github.com/Vibing/micro-frontend>

**微前端——Single-SPA（二）**

<https://www.jianshu.com/p/7bb131ec6afd>

<https://www.cnblogs.com/zhishaofei/p/12060154.html>

# single-spa-react

<https://single-spa.js.org/docs/ecosystem-react/>

<https://dev.to/dabit3/building-micro-frontends-with-react-vue-and-single-spa-52op>

<https://react.microfrontends.app/>

美团微前端

<https://tech.meituan.com/2020/02/27/meituan-waimai-micro-frontends-practice.html>

<https://tech.meituan.com/2018/09/06/fe-tiny-spa.html>

Single SPA教程比较简单

<https://www.freecodecamp.org/news/developing-and-deploying-micro-frontends-with-single-spa/>

<https://hackernoon.com/how-to-develop-and-deploy-micro-frontends-using-single-spa-framework-0a2l3u6w>

Microfrontends based on React

<https://dev.to/florianrappl/microfrontends-based-on-react-4oo9>

<https://segmentfault.com/a/1190000021044479>



这里面有个qiankun的链接

<https://zhuanlan.zhihu.com/p/78362028>

qiankun貌似是蚂蚁金服的微前端框架

# qiankun

<https://qiankun.umijs.org/zh/guide>

实践基于qiankun的微前端demo

<https://www.jianshu.com/p/ab7f36f914b3>

<https://www.jianshu.com/p/0cbe4e77c1d5>

<https://www.cnblogs.com/cx2016/p/12925875.html>

<https://juejin.im/post/6844904115999342600>

-------------------------QianKun-------------------

<https://www.jianshu.com/p/0cbe4e77c1d5>

1：npm install vue

npm install --global @vue/cli

注意：不要用 npm install –global vue-cli 这个已经是老版本了

vue -V 查看版本号

注意V要大写

2：vue create main

main是文件夹名称

cd main

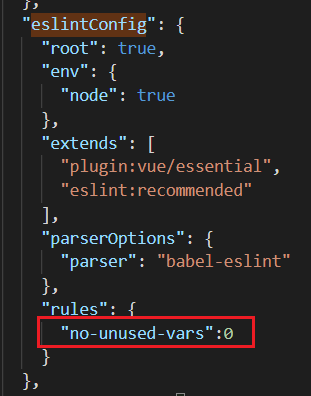
npm run serve

3：配置vue项目作为微前端的基座

4：修改ESLint配置：

<https://blog.csdn.net/mocoe/article/details/86759404>

package.json



奇怪的是，需要重新npm run serve才能生效。 不能热更新。

5：建立子应用

vue create one

cd one

npm install vue-router

vue create two

cd two

npm install vue-router

父子应用间通信 – 参数下发

<https://www.jianshu.com/p/7844076c7d15>

父子应用间通信 – 实时通信

<https://www.jianshu.com/p/fe21355be9b5>

React子应用

create-react-app three

cd three

code .

//为了自定义开发时的port，header等

npm install react-app-rewired customize-cra --save-dev

修改package.json

/\* package.json \*/

"scripts": {

- "start": "react-scripts start",

+ "start": "react-app-rewired start",

- "build": "react-scripts build",

+ "build": "react-app-rewired build",

- "test": "react-scripts test --env=jsdom",

+ "test": "react-app-rewired test --env=jsdom",

}

注意：react-app-wired 的 config-override.js没法修改devServer的port

<https://github.com/timarney/react-app-rewired/issues/436>

要么用qiankun sample里面的react sample，要么

1：添加.env文件，其中指定port

SKIP\_PREFLIGHT\_CHECK=true

BROWSER=none

PORT=7100

WDS\_SOCKET\_PORT=7100

2：或者在package.json -> scripts -> start命令中加上port=’XXXX’

<https://create-react-app.dev/docs/adding-custom-environment-variables#adding-development-environment-variables-in-env>

<https://webpack.js.org/configuration/dev-server/>

WebPack优化

<https://blog.csdn.net/sinat_17775997/article/details/91514967>

## react-hot-loader

<https://www.jianshu.com/p/244e1ffe7501>



# [React] webpack 与 babel 配置详解

<https://www.jianshu.com/p/ee46fe2dd5b8>

React webpack优化：

<https://blog.csdn.net/sinat_17775997/article/details/91514967>

webpack的14个知识点

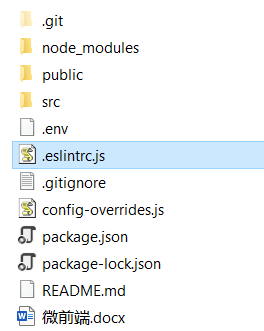
<https://juejin.im/post/6844903853905674248>

--------------ESLint-----------------

npm install eslint -g

cd 到对应文件夹， eslint –init

该命令会创建一个.eslintrc.js文件



Airbnb 等的eslint如何设置

<https://github.com/dustinspecker/awesome-eslint#configs>

webpack关闭sourcemap

<https://www.jianshu.com/p/87cded53ec81>

注意Build命令

  "scripts": {

    "start": "react-app-rewired start",

    "build": "set \"GENERATE\_SOURCEMAP=false\" && react-app-rewired build",

    "test": "react-app-rewired test",

    "lint": "eslint src/App.js"

  },

【React】首屏加载速度优化

<https://www.jianshu.com/p/f08b590e1464>

<https://www.jianshu.com/p/f08b590e1464>

<https://www.jianshu.com/p/72df512a4903>

<https://www.jianshu.com/p/7c6b28814972>

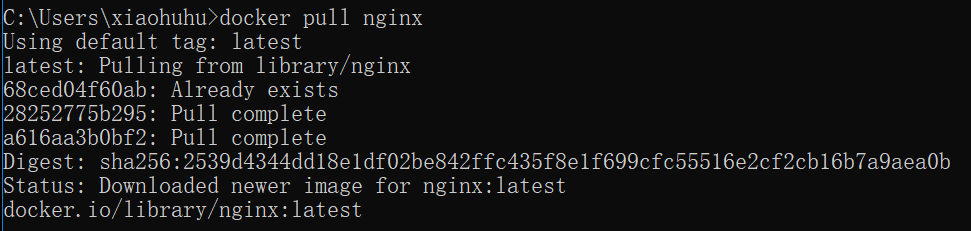
----------------------容器--------------------------------

# React App部署到Docker Container

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

拉取Ningx镜像

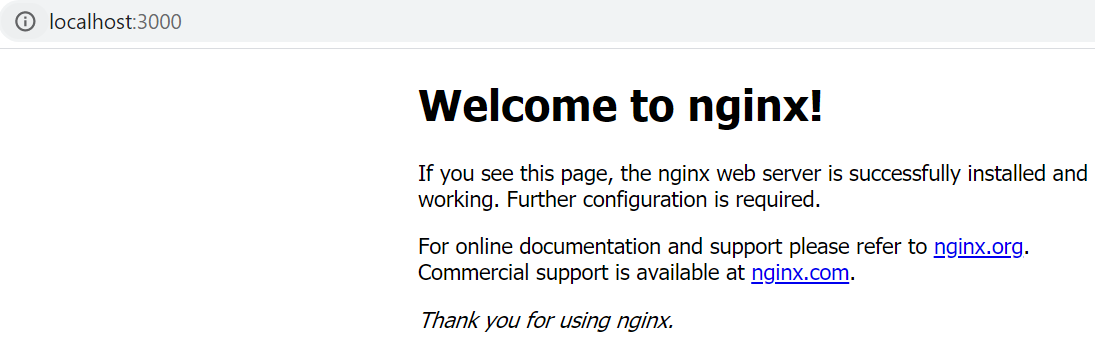
docker pull nginx



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/. a96f99630aed: /usr/share/nginx/html

a96f99630aed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

---------------------------Docker + React + Nginx ------------------------------

<https://www.jianshu.com/p/0fd24e7bb4ef>

# docker部署react项目

<https://www.dazhuanlan.com/2020/03/20/5e73a1779dc2a/>

<https://segmentfault.com/a/1190000010415158>