# 环境搭建

初始化npm，建立package.json

npm init -y

初始化项目

cnpm install (使用npm install会自动创建package-lock.json)

初始化git仓库

git init (会自动创建.git文件,默认隐藏)

创建.gitignore文件

node\_modules/

dist/

首次添加并提交，连接到远程库

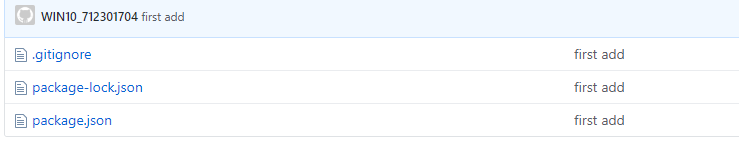
git add .

git commit -m "first commit"

git remote add origin git@github.com:longhaha1998/myblog.git

git push -u origin master

完成首次推送



根目录下创建index.html 打包时的模板html文件

创建dist文件夹存放打包后的静态文件

安装webpack，用于构建打包项目发布到服务器

cnpm install --save-dev webpack webpack-cli

创建build文件夹放置webpack不同环境下的配置文件

创建文件

utils 放置配置文件所需要的工具函数

build/webpack.base.config.js 通用配置

build/webpack.dev.config.js 开发环境配置

build/webpack.prod.config.js 生产环境配置

安装常用的loader和sass

cnpm install --save-dev node-sass

cnpm install --save-dev css-loader file-loader sass-loader style-loader url-loader

安装webpack-merge合并基础配置与制定环境的webpack配置

cnpm install webpack-merge --save-dev

安装webpac-dev-server便于开发调试

cnpm install --save-dev webpack-dev-server

安装常用webpack插件

cnpm install --save-dev webpack-bundle-analyzer html-webpack-plugin clean-webpack-plugin

安装copy-webpack-plugin拷贝静态文件

cnpm install --save-dev copy-webpack-plugin

在package.json中添加快捷命令

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

"dev": "webpack-dev-server --inline --progress --config build/webpack.dev.config.js"

安装并创建.babelrc文件配置babel

安装babel-loader

cnpm install --save-dev @babel/core @babel/preset-env @babel/preset-react

cnpm install --save-dev @babel/plugin-transform-runtime @babel/runtime

cnpm install @babel/runtime-corejs2

babel-loader

安装@babel/plugin-proposal-class-properties @babel/plugin-proposal-decorators

以支持装饰器语法

cnpm install --save-dev @babel/plugin-proposal-decorators

cnpm install --save-dev @babel/plugin-proposal-class-properties

创建static存放引用的静态文件

static/css/resetHtml.css清除页面的默认样式

创建文件夹src 存放项目资源

src/index.js webpack入口文件

自动补全浏览器前缀

cnpm install --save-dev postcss-loader autoprefixer

安装react

cnpm install --save react react-dom

使用mobx管理数据

cnpm install --save-dev mobx mobx-react

使用axios进行ajax数据传递

cnpm install --save axios

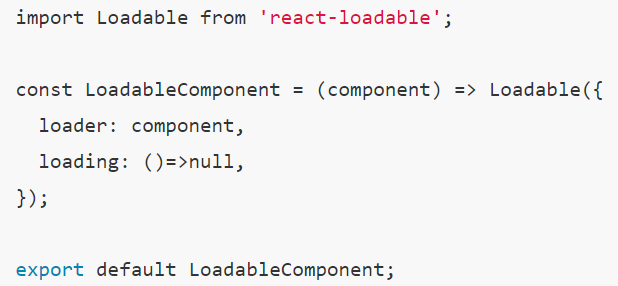
使用react router进行路由管理

cnpm install --save react-router-dom

# 打包优化

1. 路由的异步加载 react-loadable

cnpm install --save-dev react-loadable





1. 压缩css js

cnpm install --save-dev optimize-css-assets-webpack-plugin uglifyjs-webpack-plugin

cnpm install --save-dev mini-css-extract-plugin

1. 拆包

optimization.splitChunks.cachegroups