

vite 构建 react 项目——与 webpack 构建进行对比

作者: [limanting](#)

原文链接: <https://ld246.com/article/1629130413179>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

为什么使用vite

<https://cn.vitejs.dev/>

开发时能提高效率。

vite让浏览器接管了打包程序的部分工作，以原生 ESM 方式提供源码。所以vite只需要在浏览器请求码时进行转换并按需提供源码，根据情景动态导入代码，即只有在当前屏幕上使用时才会被处理。

安装

```
npm init vite@latest
yarn create vite
```

通过附加的命令行选项直接指定项目名称和你想要使用的模板。例如，要构建一个 Vite + React-ts 目，运行：

```
npm init vite@latest my-vue-app --template react-ts

# yarn
yarn create vite my-vue-app --template react-ts
```

这样我们就初始化了一个react项目，粗看上去和用webpack搭建的项目差不多，好像只是 `index.html` 在不在public文件夹当中，和 `main.tsx` 在src文件夹当中。

需要注意的地方（与webpack对比）

构建概念不同

webpack处理应用程序时，会在内部构建依赖图来映射项目需要的每个模块并生成bundle。webpac会默认将 `./src/index.js` 作为构建依赖图的开始。进入入口起点后，webpack 会找出有哪些模块和库入口起点（直接和间接）依赖的。

而vite的做法是依赖预构建。开发期间 Vite 是一个服务器，`index.html` 作为入口文件，所以 `index.html` 中需要有 `<script type="module" src="./src/main.tsx"></script>` .将 `index.html` 视为源码和块图的一部分

我们在webpack构建的项目中，运行 `yarn start` 就能将在本地运行该项目了。但是当我们使用vite构项目后，就需要 `yarn run vite` 才能使我们项目正常运行起来。这是为什么呢？因为webpack构建的目中,项目的入口是 `public/index.html` ,而vite的项目入口是 `index.html` 。

正是因为如此，我们在进行ts和eslint配置时，需要区别根目录。

常见问题

tslint报错

代码中出现 找不到tsconfig.js 的错误。

将 `eslint.config.js` 文件中的 `parserOptions` 字段中添加 `tsconfigRootDir: path.join(__dirname, '/src')`, // 因为 vite 的 `main.tsx` 的路径和 `webpack` 不同。

示例:

```
const { mkdReactConfig } = require('@monkey-design/eslint-config-mkd-react');

const config = mkdReactConfig({});

module.exports = {
  ...config,
  overrides: [
    ...config.overrides,
    {
      files: ['*.ts', '*.tsx'],
      extends: ['plugin:rxjs/recommended'],
      rules: {
        'react/sort-comp': [0],
        'jsx-a11y/click-events-have-key-events': [0],
        'react/require-default-props': ['off'],
        'no-console': 'off',
        'jsx-a11y/interactive-supports-focus': 'off',
        'react/jsx-props-no-spreading': ['off'],
        'no-unnecessary-type-assertion': ['off'],
        'no-debugger': ['off'],
        'react-hooks/exhaustive-deps': [0],
      },
      parserOptions: {
        project: './tsconfig.json',
        tsconfigRootDir: path.join(__dirname, '/src'), // 因为 vite 的 main.tsx 的路径和 webpack 不
      },
    },
  ],
};
```

代码中 `import React from 'react'` 报错。

改为 `import * as React from 'react'` .