

link: null
title: 珠峰架构师成长计划
description: publicindex.html
keywords: null
author: null
date: null
publisher: 珠峰架构师成长计划
stats: paragraph=64 sentences=111, words=592

1. roadhog

- roadhog 是基于 webpack 的封装工具，目的是简化 webpack 的配置
- 提供 server、build 和 test 三个命令，分别用于本地调试和构建
- 提供了特别易用的 mock 功能
- 命令行体验和 create-react-app 一致，配置略有不同，比如默认开启 css modules
- 还提供了 JSON 格式的配置方式。
- [roadhog \(https://www.npmjs.com/package/roadhog\)](https://www.npmjs.com/package/roadhog)

1.1 安装

1.1.1 全局安装 roadhog

```
$ npm i roadhog -g
```

1.1.2 创建项目

```
mkdir zhufeng_roadhog  
cd zhufeng_roadhog  
cnpm init -y  
cnpm i react react-dom -S
```

1.2 编码

- 默认入口文件为 src/index.js，编译后会放置到 dist\index.js **1.2.1 index.js #src\index.js**

```
import React from 'react';  
import ReactDOM from 'react-dom';  
ReactDOM.render(<h1>hello!</h1>, document.getElementById('root'));
```

** 1.2.2 index.html #**

- public 目录也是静态文件根目录

publicindex.html

```
<html lang="en">  
<body>  
  <div id="root">div</div>  
  <script src="index.js">script</script>  
</body>  
</html>
```

** 1.2.3 sum.js #**

src\sum.js

```
module.exports = function (a, b) {  
  return a + b;  
}
```

** 1.2.4 sum.test.js #**

test\sum.test.js

```
let sum = require('../src/sum');  
describe('sum', function () {  
  it('1+2=3', function () {  
    expect(sum(1, 2)).toBe(3);  
  });  
});
```

1.3 启动

** 1.3.1 本地开发 #**

```
$ roadhog server
```

** 1.3.2 打包发布 #**

```
$ roadhog build
```

** 1.3.2 测试 #**

1.3.2.1 生成配置文件

```
npmx jest --init
```

jest.config.js

```
module.exports = {  
  testEnvironment: "jsdom",  
  testURL: "http://localhost"  
}
```

1.3.2.2 执行测试命令

```
$ roadhog test
```

2. package.json

```
"scripts": {
  "dev": "roadhog server",
  "build": "roadhog build",
  "test": "roadhog test"
}
```

3. .roadhogrc

- 配置文件修改的修改会触发 roadhog server 的自动重启
- 配置存于 .roadhogrc 文件中（如果你不喜欢 JSON 配置，可以用 .roadhogrc.js 以 JS 的方式编写，支持 ES6）
- 格式为 JSON，允许注释

3.1 默认配置

```
{
  "entry": "src/index.js",
  "disableCSSModules": false,
  "publicPath": "/",
  "outputPath": "./dist",
  "extraBabelPlugins": [],
  "extraPostCSSPlugins": [],
  "autoprefixer": null,
  "proxy": null,
  "externals": null,
  "library": null,
  "libraryTarget": "var",
  "multipage": false,
  "define": null,
  "env": null,
  "theme": null,
}
```

参数 含义 entry 指定 webpack 入口文件，支持 glob 格式 outputPath 配置输出路径，默认是 ./dist disableCSSModules 禁用 CSS Modules publicPath 配置生产环境的 publicPath，开发环境下永远为 / extraBabelPlugins 配置额外的 babel plugin。babel plugin 只能添加，不允许覆盖和删除 extraPostCSSPlugins 配置额外的 postcss 插件 autoprefixer 配置 autoprefixer 参数 proxy 配置代理 externals 配置 webpack 的 externals 属性 library 配置 webpack 的 library 属性 libraryTarget 配置 webpack 的 libraryTarget 属性 multipage 配置是否多页应用。多页应用会自动提取公共部分为 common.js 和 common.css define 配置 webpack 的 DefinePlugin 插件，define 的值会自动做 JSON.stringify 处理 env 针对特定的环境进行配置。server 的环境变量是 development，build 的环境变量是 production theme 配置主题，实际上是配 less 的 modifyVars

4. .roadhogrc.mock.js

```
export default {
  'GET /api/users': { users: [1,2] },
  'GET /api/users/1': { id: 1 },
  'POST /api/users/create': (req, res) => { res.end('OK'); },
};
```

5. 实现

5.1 初始化项目

```
mkdir zfroadhog
cd zfroadhog
cnpm init -y
cnpm i chalk cross-spawn webpack -S
```

5.2 package.json

```
{
+   "bin": {
+     "zfroadhog": "./bin/zfroadhog.js"
+   },
+ }
}
```

5.3 roadhog.js

bin\roadhog.js

```
const spawn = require('cross-spawn');
const chalk = require('chalk');
const script = process.argv[2];

switch (script) {
  case "-v":
  case "--version":
    console.log(require("../package.json").version);
    break;
  case "build":
  case "server":
    result = spawn.sync(
      "node",
      [require.resolve('../lib/${script}')],
      { stdio: "inherit" }
    );
    process.exit(result.status);
    break;
  default:
    console.log(`Unknown script ${chalk.cyan(script)}.`);
    break;
}
```

5.4 build.js

lib\build.js

```
const webpack = require("webpack");
const chalk = require("chalk");
function doneHandler(err, stats) {
  console.log(stats.toJson().assets);
}
function build() {
  let config = require("./config/webpack.config.prod");
  var compiler = webpack(config);
  compiler.run(doneHandler);
}
build();
```

5.5 webpack.config.prod

lib\config\webpack.config.prod.js

```
const path = require("path");
module.exports = {
  entry: "./src/index.js",
  output: {
    path: path.resolve('dist'),
    filename: "[name].js"
  }
};
```

5.参考

- [roadhog \(https://www.npmjs.com/package/roadhog\)](https://www.npmjs.com/package/roadhog)
- [zhufeng-roadhog \(https://gitee.com/zhufengpeixun/zhufeng-roadhog\)](https://gitee.com/zhufengpeixun/zhufeng-roadhog)



Baidu 百科
OVERWATCH™