

# reactjs 初体验

作者: [flowaters](#)

原文链接: <https://ld246.com/article/1542350633685>

来源网站: [链滴](#)

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

# 背景

reactjs是一款js框架。

## Hello World 初体检

先做一个 hello world的页面。

编辑文件 index.html, 内容如下:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Hello React!</title>
    <!-- 以下引入 react.js, react-dom.js (react 0.14 后将 react-dom 从 react 核心分离, 更符合
    eact 跨平台抽象化的定位) 以及 babel-core browser 版 -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react.min.js"></script>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/react/15.2.1/react-dom.min.js"></scr
    pt>
    <script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/6.18.1/babel.min.js
  ></script>
  </head>
  <body>
    <!-- 这边的 id="example" 的 <div> 为 React Component 要插入的地方 -->
    <div id="example"></div>
    <!-- 以下就是包在 babel (透过进行语言翻译) 中的 React JSX 语法, babel 会将其转为浏览
    看的懂得 JavaScript -->
    <script type="text/babel">
      ReactDOM.render(
        <h1>Hello, world!</h1>,
        document.getElementById('example')
      );
    </script>
  </body>
</html>
```

用浏览器打开即可。

## create-react-apps 框架初体验

### 生成代码框架

```
$ npx create-react-app my-app
npx: installed 63 in 2.963s
```

Creating a new React app in /Users/note/web/my-app.

Installing packages. This might take a couple of minutes.  
Installing react, react-dom, and react-scripts...

```
> fsevents@1.2.4 install /Users/note/web/my-app/node_modules/fsevents
> node install
```

```
[fsevents] Success: "/Users/note/web/my-app/node_modules/fsevents/lib/binding/Release/n
de-v64-darwin-x64/fse.node" already installed
Pass --update-binary to reinstall or --build-from-source to recompile
+ react-dom@16.6.3
+ react@16.6.3
+ react-scripts@2.1.1
added 1768 packages from 678 contributors and audited 35639 packages in 24.233s
found 0 vulnerabilities
```

Initialized a git repository.

Success! Created my-app at /Users/note/web/my-app  
Inside that directory, you can run several commands:

```
npm start
  Starts the development server.
```

```
npm run build
  Bundles the app into static files for production.
```

```
npm test
  Starts the test runner.
```

```
npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!
```

We suggest that you begin by typing:

```
cd my-app
npm start
```

Happy hacking!

## 开发运行

```
cd my-app
npm start
```

## 生产运行

```
npm run build
```

```
> my-app@0.1.0 build /Users/note/web/my-app
> react-scripts build
```

```
Creating an optimized production build...
Compiled successfully.
```

File sizes after gzip:

```
34.45 KB build/static/js/1.2dd891c4.chunk.js
763 B   build/static/js/runtime~main.229c360f.js
711 B   build/static/js/main.c5536c2d.chunk.js
510 B   build/static/css/main.52bcc55a.chunk.css
```

The project was built assuming it is hosted at the server root.  
You can control this with the homepage field in your package.json.  
For example, add this to build it for GitHub Pages:

```
"homepage" : "http://myname.github.io/myapp",
```

The build folder is ready to be deployed.  
You may serve it with a static server:

```
npm install -g serve
serve -s build
```

Find out more about deployment here:

```
http://bit.ly/CRA-deploy
```

## 说明

Create React App使用了Babel和webpack，尽管我们不需要关注其中的细节。

## SpringBoot集成

用简单的示例，没有使用模板。

## 页面

将build的文件，复制到 resource/static 目录下，如下：

```
$ tree resources/static/
resources/static/
├── asset-manifest.json
├── css
│   ├── main.52bcc55a.chunk.css
│   └── main.52bcc55a.chunk.css.map
├── favicon.ico
├── index.html
├── js
│   ├── 1.2dd891c4.chunk.js
│   ├── 1.2dd891c4.chunk.js.map
│   ├── main.c5536c2d.chunk.js
│   ├── main.c5536c2d.chunk.js.map
│   ├── runtime~main.229c360f.js
│   └── runtime~main.229c360f.js.map
└── manifest.json
```

```
├── media
│   └── logo.5d5d9eef.svg
├── precache-manifest.b4ea6b0323ea4d7e2e45065ece29b83e.js
└── service-worker.js
```

## 访问

```
$ curl -s "http://localhost:8080/index.html"
<!doctype html><html lang="en"><head><meta charset="utf-8">
```

## 非根路径

如果我们没有放在根路径下，而在放在 demo 路径下的话，打开页面后，会提示js和css资源找不到。

解析方法是重新打包，将资源文件打包成相对路径，修改 `package.json` 文件，增加 `homepage` 行，下：

```
{
  "name": "my-app",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "react": "^16.6.3",
    "react-dom": "^16.6.3",
    "react-scripts": "2.1.1"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "test": "react-scripts test",
    "eject": "react-scripts eject"
  },
  "eslintConfig": {
    "extends": "react-app"
  },
  "browserslist": [
    ">0.2%",
    "not dead",
    "not ie <= 11",
    "not op_mini all"
  ],
  "homepage": "."
}
```

执行 `npm run build` 打包

再重新复制，即可。

## 其它

## 使用方式

使用reactjs时，自己编写jsx文件，然后编译成js文件，再在浏览器中运行。

这样可以方便的对js进行封装。

## 打包

- Webpack
- Browserify
- Gulp

## 参考

- [React 生态系 \(Ecosystem\) 入门简介](#)
- [React 开发环境设置与 Webpack 入门教学](#)
- [Create a New React App](#)