

## 8.4 实战演练：使用 Webpack 构建一个 React 应用

Webpack 是一个模块加载器兼打包工具，能把各种资源，例如脚本（JavaScript、TypeScript、JSX）、样式（CSS、Less、Sass）、图片（png、jpg、gif）等都作为模块来处理。目前，主流前端开发框架（如 React、Vue.js）的官方实例项目均使用 Webpack 作为工程化工具。

### 8.4.1 安装和配置

本节会创建一个简单的 React 项目，并使用 Webpack 构建开发环境以及打包生产环境所需的静态资源。项目目录结构如图 8.10 所示。

app 文件夹存储打包前的静态文件，里面的文件会被 Webpack 打包，打包后的文件会存储到 public 文件夹。

index.js 文件是入口脚本文件，内容如下：

```
import React, { Component } from 'react'; // 引入 react 包
import ReactDOM from 'react-dom'; // 引入 react-dom 包
import Hello from './hello'; // 引入依赖的 Hello 组件
class App extends Component { // 定义入口组件
  render() { // 绘制组件
    return (
      <div>
        <Hello /> // 使用 Hello 组件
      </div>
    )
  }
}
ReactDOM.render( // 绘制组件到页面
  <App />,
  document.getElementById('root')
);
```

hello.js 文件定义了一个 React 组件，被 index.js 文件使用，代码如下：

```
import React, { Component } from 'react'; // 引入 react 包
export default class Hello extends Component { // 定义 Hello 组件并导出
  render() { // 绘制组件
    return (
      <div>
        Hello World
      </div>
    )
  }
}
```

**注意：**React 的详细使用会在后续章节中介绍，此处只是简单使用。

public 文件夹存储需要被浏览器访问到的文件，打包后生成的入口脚本文件也会存储在这个文件夹当中。

index.html 是入口页面文件，内容如下：

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Webpack Sample Project</title>
  </head>
  <body>
    <!-- React 组件绘制的 DOM 会作为子元素插入 -->
    <div id='root'>
    </div>
    <!-- 引用入口脚本文件 -->
    <script src="./bundle.js"></script>
  </body>
</html>

```

`package.json` 文件是标准的 NPM 说明文件，这个文件可以通过 NPM 命令生成。打开命令行窗口，输入如下命令：

```
npm init
```

控制台会逐步提示用户输入创建 `package.json` 所需要的相关信息，例如项目名称、版本号等。`package.json` 文件内容如下：

```

{
  "name": "webpack-demo",           // 项目名称
  "version": "1.0.0",              // 项目版本
  "description": "webpack demo",   // 项目描述
  "main": "index.js",             // 项目入口文件
  "scripts": {                    // 项目支持的脚本命令
    "start": "webpack-dev-server" // start 命令，用于启动开发环境
    "build": "webpack "          // build 命令，用于打包生产环境静态资源
  },
  "author": "",                   // 项目作者
  "license": "ISC",               // 项目 license
  "dependencies": {               // 项目运行时依赖的包
    "react": "^15.4.1",
    "react-dom": "^15.4.1"
  },
  "devDependencies": {            // 项目开发时依赖的包
    "babel-core": "^6.21.0",
    "babel-loader": "^6.2.10",
    "babel-preset-latest": "^6.16.0",
    "babel-preset-react": "^6.16.0",
    "webpack": "^1.14.0",
    "webpack-dev-server": "^1.16.2" // 提供 Web 服务
  }
}

```

`webpack.config.js` 是 Webpack 配置文件，告诉 Webpack 需要做什么，Webpack 运行时会读取此文件，内容如下：

```

var webpack = require('webpack'); // 引入 webpack 包
module.exports = {                // 导出配置
  entry:  dirname + "/app/index.js", // 入口文件
  output: {
    path:  __dirname + "/public",    // 输出路径
    filename: "bundle.js"           // 输出文件名
  }
}

```

```

},
module: {
  loaders: [ // 加载器列表
    {
      test: /\.js$/, // 处理后缀名为 js 的文件
      exclude: /node_modules/, // 需要忽略的文件夹
      loader: 'babel', // 加载器
      query: {
        presets: [ // 插件集列表
          'es2015', // 处理 es2015 的 Babel 插件
          'react' // 处理 react 的 Babel 插件
        ]
      }
    }
  ]
},
devServer: { // Web Server 配置
  contentBase: './public', // 暴露给 Web Server 的文件夹
  inline: true // Web Server 运行模式
}
}

```

项目配置完成之后，安装项目所需 NPM 依赖包。打开命令行窗口，运行如下命令：

```
npm install
```

npm 包安装成功之后，启动项目。打开命令行窗口，运行如下命令：

```
npm run start
```

项目成功启动之后，可以打开浏览器，输入地址 <http://localhost:8080/index.html> 访问页面。

**注意：**开发环境下，打包生成的静态资源 `bundle.js` 存储在内存里面，不会输出到磁盘，在 `public` 文件夹下无法看到 `bundle.js` 文件。

如果想生成生产环境所需静态资源，打开命令行窗口，输入如下命令：

```
npm run build
```

命令运行成功后，`public` 文件夹下会生成打包后的静态资源 `bundle.js`。