

快速开始

运行 webpack5 需要 Node.js 最低版本是 10.13.0 (LTS)。推荐使用 [nvm](#) 来安装和管理 node.js 版本。

1、初始化一个项目

```
mkdir webpack5-demo
cd webpack5-demo
npm init -y
npm install webpack webpack-cli --save-dev
```

初始化为 `webpack5-demo` 的项目，并安装 `webpack` 和 `webpack-cli`（此工具用于在命令行中运行 `webpack`）作为开发环境依赖。

2、编写示例代码

新建文件 `src/index.js`

```
import { sayHello } from './utils.js'

sayHello('Webpack5')
```

`src/utils.js` 文件内容如下：

```
export const sayHello = (words) => {
  console.log(`Hello ${words}`)
}
```

在 `src/index.js` 文件中，使用 ES6 语法，从 `src/utils.js` 文件导出了一个方法 `sayHello` 然后调用。

此时项目目录机构如下：

```
→ webpack5-demo
├─ package-lock.json
├─ package.json
└─ src
   └─ index.js
      └─ utils.js
```

3、添加scripts

在 `package.json` 中添加一条 `build` 命令来执行 webpack 的构建

```
"scripts": {  
  "build": "webpack"  
}
```

4、执行webpack构建

执行 `npm run build` 来进行webpack的构建:

```
→ webpack5-demo npm run build  
  
> webpack5-demo@1.0.0 build /Users/ludis/Desktop/webpack5/webpack5-demo  
> webpack  
  
asset main.js 53 bytes [emitted] [minimized] (name: main)  
orphan modules 64 bytes [orphan] 1 module  
./src/index.js + 1 modules 123 bytes [built] [code generated]  
  
WARNING in configuration  
The 'mode' option has not been set, webpack will fallback to 'production'  
for this value.  
Set 'mode' option to 'development' or 'production' to enable defaults for  
each environment.  
You can also set it to 'none' to disable any default behavior. Learn more:  
https://webpack.js.org/configuration/mode/  
  
webpack 5.65.0 compiled with 1 warning in 237 ms
```

在执行完构建的终端输出中有个警告，提示没有配置 `mode` 参数，默认会按照 `production` 生产模式构建，可以选择 `development` 开发模式。

根据不同的模式，webpack构建时会做不同的处理，例如生产模式会进行代码压缩等操作。这部分内容后面会细说，这里先忽略此配置。

观察我们的项目，会发现多出来一个 `dist` 目录，里面是本次构建后的产物 `main.js`:

```
((=>{"use strict";console.log("Hello Webpack5")}))();
```

可以看到webpack已经完成了构建，`main.js`是一个自执行函数，内容就是我们`index.js`和`utils.js`合并执行的最终结果。

这里我们已近通过零配置，实现了webpack的构建。这是从webpack4开始提供的新特性，即零配置也能运行！（社区中常有webpack工程师的说法，意为webpack配置过于复杂，这里零配置启动是webpack改进的一小步）

5、使用自定义配置

实现了零配启动，接下来我们看如何让webpack使用自定义的配置。在项目根目录新建配置文件文件：
`webpack.config.js`:

```
const path = require('path');

module.exports = {
  entry: './src/index.js', // 项目的入口文件
  output: {
    filename: 'main.js', // 构建完输出的文件
    path: path.resolve(__dirname, 'dist'), // 构建完产物的存储目录
  },
};
```

我们先删除`dist`文件夹，重新执行构建`npm run build`。会发现，构建完的结果，跟我们之前零配置的没有任何区别。

其实，我们零配置使用webpack构建时，webpack会给我们使用一套默认的配置，这套配置就是我们刚才创建的`webpack.config.js`中的内容。所以要使用webpack的零配置启动，是有前提的，那就是要按他默认的文件目录结构来才行。大家可以尝试删掉自定义的配置，将入口文件`src/index.js`命名为`src/entry.js`，再尝试零配置启动，就会发现webpack报错。

揭露了零配启动的面纱，会发现其实这个零配启动比较鸡肋，但也聊胜于无。

Webpack 在构建时会默认读取名为 `webpack.config.js` 及 `webpack.config.json` 的配置文件，我们也可以自定义命名此配置文件，然后在构建命令指定配置文件：`webpack --config custom.config.js`

6、启动本地开发服务器

在日常开发中，我们大部分时间都是在本地进行开发，只有上线时才会进行build打包。在本地开发时，我们需要一个本地的开发服务器，来运行我们的项目，并且我们的js代码需要嵌入到一个HTML文件中，这样我们才能在浏览器中运行我们的js。

1. 使用`webpack-dev-server`来启动一个本地的开发服务器

安装依赖：`npm install webpack-dev-server --save-dev`

配置`webpack.config.js`:

```
const path = require('path');

module.exports = {
  entry: './src/index.js', // 项目的入口文件
  output: {
    filename: 'main.js', // 构建完输出的文件
    path: path.resolve(__dirname, 'dist'), // 构建完产物的存储目录
  },
  mode: 'development', // 指明当前为开发环境
  devServer: {
```

```
    port: 9000, // 指定开发服务器启动端口
  }
};
```

2. 创建一个`public/index.html`文件，引入我们的js代码，这里我们需要引入经过webpack构建后产生的js，也就是 `main.js`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Webpack5</title>
</head>
<body>
  <script src="./main.js"></script>
</body>
</html>
```

3. 新增一个启动本地开发服务器的命令 `dev`:

```
"scripts": {
  "build": "webpack",
  "dev": "webpack serve"
}
```

执行`npm run dev`，会发现webpack在本地启动了一个开发服务器：

```
→ webpack5-demo npm start

> webpack5-demo@1.0.0 start /Users/ludis/Desktop/webpack5/webpack5-demo
> webpack serve

<i> [webpack-dev-server] Project is running at:
<i> [webpack-dev-server] Loopback: http://localhost:9000/
<i> [webpack-dev-server] On Your Network (IPv4): http://192.168.43.8:9000/
<i> [webpack-dev-server] On Your Network (IPv6): http://[fe80::1]:9000/
<i> [webpack-dev-server] Content not from webpack is served from
' /Users/ludis/Desktop/webpack5/webpack5-demo/public ' directory
...
...
webpack 5.65.0 compiled successfully in 353 ms
```

可以看到，本地启动了一个服务器 `http://localhost:9000/`，我们通过浏览器打开，可以在控制台看到如下内容：

说明我们本地服务器启动了，并成功运行了webpack构建完产生的js模块，输出了正确的结果。

小结

1. 了解了webpack的零配置启动及其原理
2. 如何编写使用简单的配置文件
3. 如何在本地启动开发服务器

练习及思考

自行实现本节中的内容，并思考探索：

1. 如何自动设置mode，在本地开发环境为development，在生产模式构建时为production？
2. 为什么开发服务器默认能打开public/index.html？index.html可以放在其他地方而不是public/index.html吗？如果可以，需要什么额外的配置？
3. npm run build构建完的js是在dist/main.js，为什么public/index.html文件中引入了./main.js没有报错？如果删除了dist构建目录，对本地开发环境有影响吗？