



# Webpack 基础

## 核心概念

- entry: 入口, webpack 构建第一步 ;
- output: 输出;
- loader: 模块转换器, 用于将模块的原内容按照需求转换成新内容;
- plugin: 扩展插件, 在 webpack 构建过程的特定时机注入扩展逻辑, 用来改变或优化构建结果;
- mode: 控制打包环境 通过选择 development, production 或 none 之中的一个, 来设置 mode 参数, 你可以启用 webpack 内置在相应环境下的优化。其默认值为 production 环境;
- devserver: 是一个小型的 node.js Express 服务器, 使用 webpack-dev-middleware 中间件来为通过 webpack 打包生成的资源文件提供 web 服务。

## entry point(入口起点)

入口起点(entry point) 指示 webpack 应该使用哪个模块, 来作为构建其内部 依赖图(dependency graph) 的开始。进入入口起点后, webpack 会找出有哪些模块和库是入口起点(直接和间接) 依赖的。

index.js

JavaScript

```
import { add } from "./add";
console.log(add(3, 3));
```

add.js

JavaScript

```
export const add = (a,b) => a + b
```

webpack.config.js

JavaScript

```
module.exports = {
  entry: "./src/index.js"
};
```

---

## output (输出)

output 属性告诉 webpack 在哪里输出它所创建的 bundle，以及如何命名这些文件。主要输出文件的默认值是 ./dist/main.js，其他生成文件默认放置在 ./dist 文件夹中。

JavaScript

```
const path = require("path");
const HtmlWebpackPlugin = require('html-webpack-plugin');
module.exports = {

  output: {
    path: path.resolve(__dirname, "dist"),
    filename: "main.bundle.js",
  },

};
```

---

## Loader 加载器

loader 用于对模块的源代码进行转换。loader 可以使你在 import 或 "load(加载)" 模块时预处理文件。

- 默认只能处理 json 与 js
- 其他文件需要通过专门的加载器处理

hello.txt

```
Hello Raw Loader
```

JSON

```
import hello from './hello.txt';  
document.writeln(hello);
```

JavaScript

```
yarn add raw-loader
```

Bash

webpack.config.js

```
module: {  
  rules: [{ test: /\.txt$/, use: 'raw-loader' }],  
},
```

JavaScript

## plugin 插件

扩展插件，在 webpack 构建过程的特定时机注入扩展逻辑，用来改变或优化构建结果；

```
yarn add html-webpack-plugin -d
```

Bash

webpack.config.js

```
const HtmlWebpackPlugin = require('html-webpack-plugin');  
module.exports = {
```

JavaScript

```
plugins: [new HtmlWebpackPlugin({ template: './src/index.html' })],  
  
};
```

HTML

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta http-equiv="X-UA-Compatible" content="IE=edge">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Document</title>  
</head>  
<body>  
  <h1>Hello webpack3</h1>  
</body>  
</html>
```

## DevServer 开发服务器

提供了一个基本的 web server，并且具有 live reloading(实时重新加载) 功能。。

- 静态服务比如:图片
- live reloading(实时重新加载)
- 反向代理接口

Bash

```
yarn add -D webpack-dev-server
```

Bash

```
npx webpack serve
```