



# Rollup 基础

官方文档 <https://www.rollupjs.com/>

## 🔥 你将学到 Get

- rollup 的应用场景
- Treeshaking 是什么
- rollup 多种模块化格式的不同
- rollup 的常见插件

## What

rollup 是一款小巧的 javascript 模块打包工具，更适用于库应用的构建工具;可以将小块代码编译成大块复杂的代码，基于 ES6 modules,它可以让你的 bundle 最小化，有效减少文件请求大小,vue 在开发的时候用的是 webpack,但是最后将文件打包在一起的时候用的是 rollup.js。

## Treeshaking 测试

Tree-shaking 的本质是消除无用的 js 代码。无用代码消除在广泛存在于传统的编程语言编译器中，编译器可以判断出某些代码根本不影响输出，然后消除这些代码，这个称之为 DCE (dead code elimination)

## 打包格式

- amd — 异步模块定义，用于像 RequireJS 这样的模块加载器
- cjs — CommonJS，适用于 Node 和 Browserify/Webpack
- es — 将软件包保存为 ES 模块文件
- iife — 一个自动执行的功能，适合作为 `<script>` 标签。（如果要为应用程序创建一个捆绑包，您可能想要使用它，因为它会使文件大小变小。）
- umd — 通用模块定义，以 amd, cjs 和 iife 为一体

### 举例

#### Vue3 源码分析

	UMD	CommonJS	ES Module
Full	vue.js	vue.common.js	vue.esm.js
Runtime-only	vue.runtime.js	vue.runtime.common.js	vue.runtime.esm.js
Full (production)	vue.min.js		
Runtime-only (production)	vue.runtime.min.js		@稀土掘金技术社区

## 模块实战

创建 packages/utils 文件夹

## 初始化 npm

```
npm init
```

Bash

## package.json

```
{
  "name": "@smarty-admin/utils",
  "version": "1.0.0",
  "main": "src/index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "type": "module",
  "license": "ISC",
  "description": ""
}
```

JSON

## /utils/index.js

```
export const whoAmI = 'I am utils'
```

JavaScript

## 安装 rollup

```
npm i rollup -d
```

Bash

## 简单编译

---

Bash

```
./node_modules/.bin/rollup -i src/index.js -o dist/bundle.es.js -f es  
./node_modules/.bin/rollup -i src/index.js -o dist/bundle.umd.js -f umd -n
```

测试 UMD 包的 IIFE 功能

/demo/index.html

HTML

```
<script src="../dist/bundle.umd.js"></script>  
<script>  
  console.log(utils.whoAmI)  
</script>
```

测试 AMD 方式

安装 requirejs

Bash

```
npm i requirejs -d
```

HTML

```
<script src="../node_modules/requirejs/require.js"></script>  
<script type="module">  
  require(['../dist/bundle.umd.js'],utils => {  
    console.log('amd requirejs:',utils.whoAmI)  
  })  
  
</script>
```

测试 CommonJS 方式

JavaScript

```
import { whoAmI } from '../dist/bundle.es.js'
console.log(whoAmI)
```

## 插件的使用

为了更好的支持复杂的应用场景，可以通过插件进行扩展。

- es6 语法
- .vue 文件

### rollup-plugin-babel

`rollup-plugin-babel` 用于转换 es6 语法。

### rollup-plugin-commonjs

rollup 默认是不支持 CommonJS 模块的，自己写的时候可以尽量避免使用 CommonJS 模块的语法，但有些外部库的是 cjs 或者 umd（由 webpack 打包的），所以使用这些外部库就需要支持 CommonJS 模块。

### rollup-plugin-postcss

处理 css 需要用到的插件是 `rollup-plugin-postcss`。它支持 css 文件的加载、css 加前缀、css 压缩、对 scss/less 的支持等等。

### rollup-plugin-vue

### rollup-plugin-terser

在生产环境中，代码压缩是必不可少的。我们使用 `rollup-plugin-terser` 进行代码压缩。

## rollup-plugin-serve

`rollup-plugin-serve` 用于启动一个服务器,

## rollup-plugin-livereload

`rollup-plugin-livereload` 用于文件变化时, 实时刷新页面

### 代码压缩插件实战

```
npm i rollup-plugin-terser -d
```

Bash

### rollup.config.js

```
import { terser } from "rollup-plugin-terser";
export default {
  input: "./src/index.js",
  plugins: [terser()],
  output: [
    {
      file: "./dist/es.min.js",
      format: "es",
    },
  ],
};
```

JavaScript

```
./node_modules/.bin/rollup -c rollup.config.js
```

Bash

