

【架构师（第九篇）】如何让 Node 环境支持 ES Module

一尾流莺 2022-04-06 09:08 👁 1695

关注

了解模块化

Commonjs

- 加载模块: `require`
- 导出模块: `module.exports` / `exports.xxx`

ES Module

- 加载模块: `import`
- 导出模块: `export default` / `export xxx`

痛点分析

使用 `Commonjs` 导出一个模块 `utils`



js 复制代码

```
1 // test-cli-0174\bin\utils.js
2 module.exports = function () {
3   console.log('hello utils');
4 };
```

主文件中通过 `ES Module` 的方式引入模块



js 复制代码

```
1 // test-cli-0174\bin\index.js
2 #!/usr/bin/env node
3
4 import utils from './utils';
```

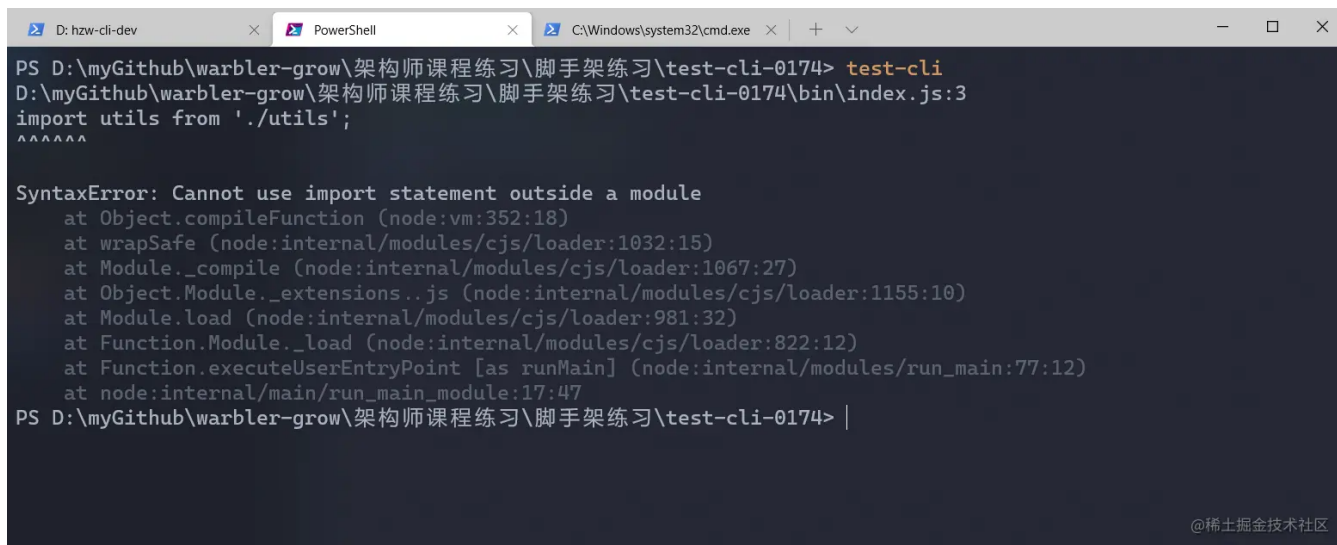
APP内打开

 14

 4

 收藏

运行程序，发现会报错 `Cannot use import statement outside a module`，意思就是不让用 `import` 语法。



```
PS D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174> test-cli
D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174\bin\index.js:3
import utils from './utils';
^^^^^^

SyntaxError: Cannot use import statement outside a module
    at Object.compileFunction (node:vm:352:18)
    at wrapSafe (node:internal/modules/cjs/loader:1032:15)
    at Module._compile (node:internal/modules/cjs/loader:1067:27)
    at Object.Module._extensions..js (node:internal/modules/cjs/loader:1155:10)
    at Module.load (node:internal/modules/cjs/loader:981:32)
    at Function.Module._load (node:internal/modules/cjs/loader:822:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:77:12)
    at node:internal/main/run_main_module:17:47
PS D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174> |
```

@稀土掘金技术社区

那么我们如何让 `Node` 环境支持 `ES Module` 呢？

利用 webpack

安装 webpack

[js 复制代码](#)

```
1 npm i -D webpack webpack-cli
```

修改代码

主文件使用 `require` 去调用 `webpack` 构建后的 `core.js`

[js 复制代码](#)

```
1 // test-cli-0174\bin\index.js
2 #!/usr/bin/env node
3 require('./dist/core');
```

[APP内打开](#)

```
1 // test-cli-0174\bin\core.js
2 import utils from './utils';
3 utils();
```

配置好 webpack.config.js

```
1 // webpack.config.js
2 const path = require('path');
3
4 module.exports = {
5   entry: './bin/core.js',
6   output: {
7     path: path.join(__dirname, '/dist'),
8     filename: 'core.js',
9   },
10  mode: 'development',
11 };
```

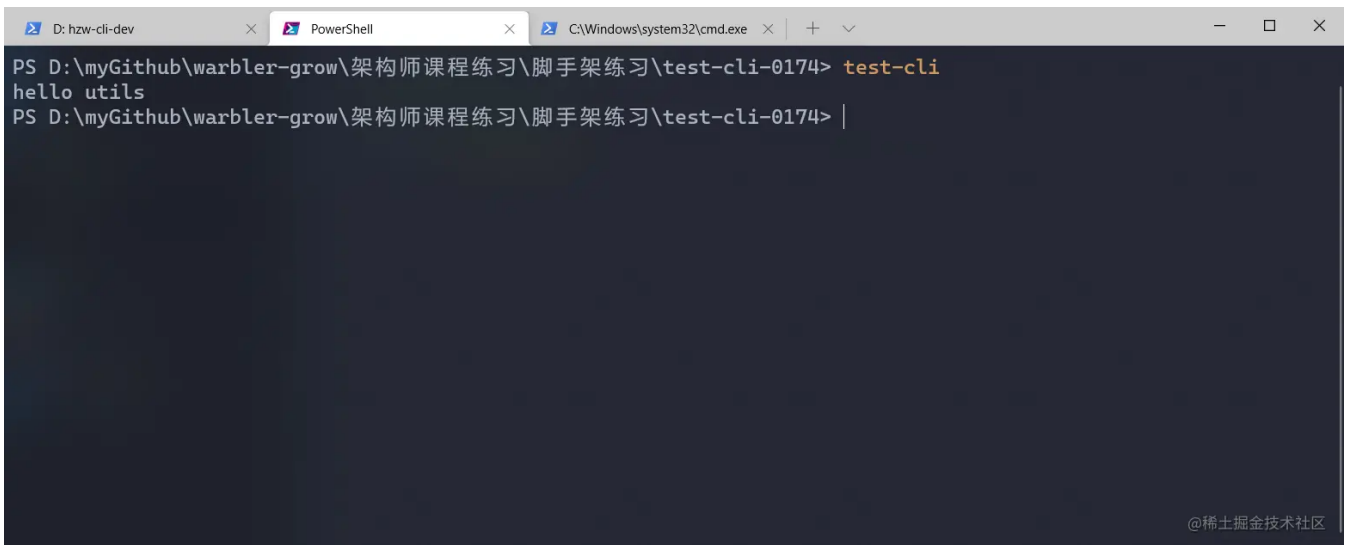
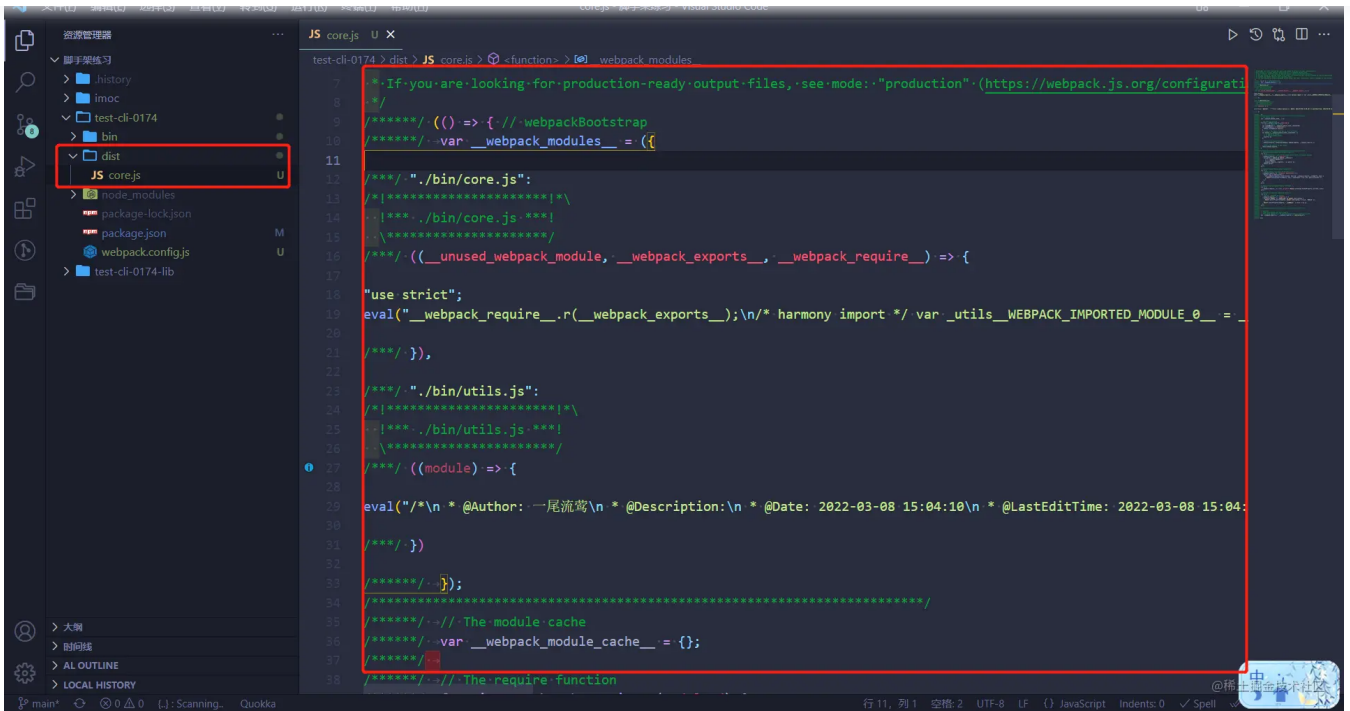
修改 package.json 的 scripts 字段

```
1 "scripts": {
2   "build": "webpack",
3   "dev": "webpack -w"
4 },
```

执行构建

```
1 npm run build
```

APP内打开



js 复制代码

1 npm run dev

通过 webpack target 属性支持 Node

APP内打开

js 复制代码

```
1 // webpack.config.js
2 const path = require('path');
3
4 module.exports = {
5   entry: './bin/core.js',
6   output: {
7     path: path.join(__dirname, '/dist'),
8     filename: 'core.js',
9   },
10   target: 'node', // 默认是web
11 };
```

js 复制代码

```
1 // test-cli-0174\bin\utils.js
2 import { pathExistsSync } from 'path-exists';
3
4 export function exists(p) {
5   return pathExistsSync(p);
6 }
```

js 复制代码

```
1 // test-cli-0174\bin\core.js
2 import path from 'path';
3 import { exists } from './utils';
4
5 console.log(path.resolve('.'));
6 console.log(exists(path.resolve('.')));
```

执行程序，没有什么问题了。

APP内打开

```
true
PS D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174> |
```

@稀土掘金技术社区

利用 babel 兼容低版本 node

安装依赖



js 复制代码

```
1 npm i -D
2 babel-loader
3 @babel/core
4 @babel/preset-env
5 @babel/plugin-transform-runtime
6 @babel/runtime-corejs3
```

配置 webpack



js 复制代码

```
1 // webpack.config.js
2 const path = require('path');
3
4 module.exports = {
5   entry: './bin/core.js',
6   output: {
7     path: path.join(__dirname, '/dist'),
8     filename: 'core.js',
9   },
10  mode: 'development',
11  target: 'node', // 默认是web
12  module: {
13    rules: [
```

APP内打开

```
18 loader: 'babel-loader',
19 options: {
20   presets: ['@babel/preset-env'],
21   plugins: [
22     [
23       '@babel/plugin-transform-runtime',
24       {
25         corejs: 3,
26         regenerator: true,
27         useESModules: true,
28         helpers: true,
29       },
30     ],
31   ],
32 },
33 },
34 },
35 ],
36 },
37 };
```

node 原生如何支持 ES Module

需要修改文件名的后缀为 `.mjs`，并且一旦使用 `mjs`，所有的内容都需要使用 `ES Module`，不可混用，否则会报错。

[js 复制代码](#)

```
1 // test-cli-0174\bin\index.mjs
2 #!/usr/bin/env node
3
4 import './core.mjs';
```

[js 复制代码](#)

```
1 // test-cli-0174\bin\core.mjs
2 import path from 'path';
3 import { exists } from './utils.mjs';
4
5 console.log(path.resolve('.'));
6 console.log(exists(path.resolve('.')));
```

[APP内打开](#)

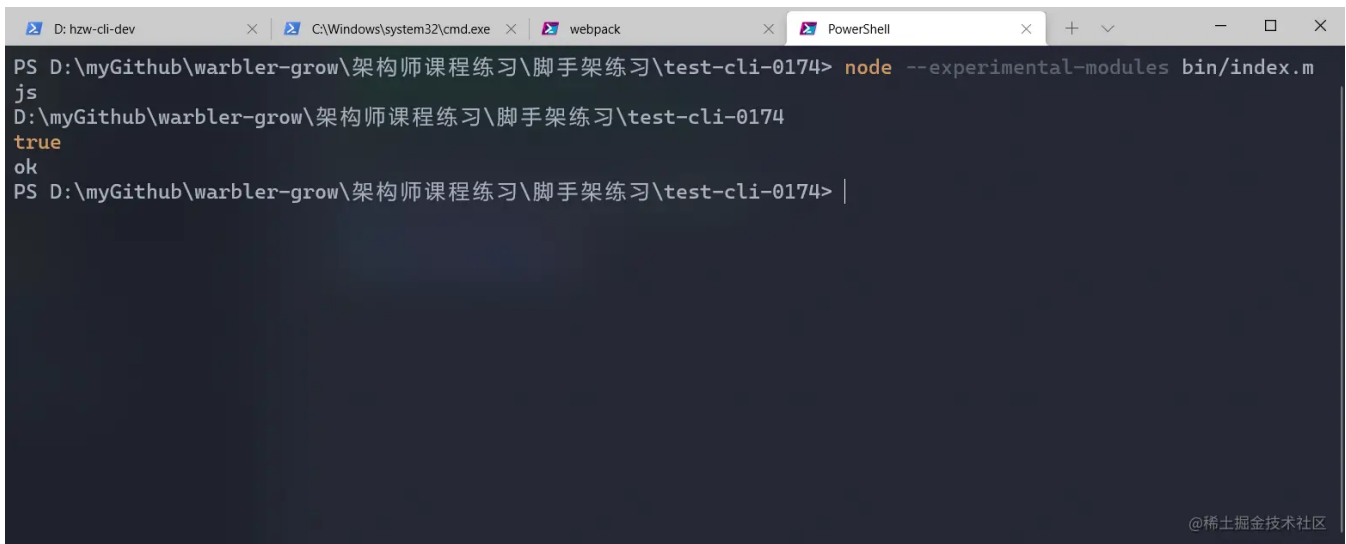
```
3
4 export function exists(p) {
5   return pathExistsSync(p);
6 }
```

最后 通过一个指令来执行程序

[js](#) 复制代码

```
1 node --experimental-modules bin/index.mjs
2 // node14版本之后 不需要加 --experimental-modules 指令也可以
3 node bin/index.mjs
```

同样能得到结果，没有什么问题。



```
PS D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174> node --experimental-modules bin/index.mjs
js
D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174
true
ok
PS D:\myGithub\warbler-grow\架构师课程练习\脚手架练习\test-cli-0174> |
```

@稀土掘金技术社区

如果不希望将后缀名改成 `.mjs`，可以在项目的 `package.json` 文件中，指定 `type` 字段为 `module`。

[js](#) 复制代码

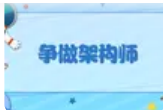
```
1 {
2   "type": "module"
3 }
```

一旦设置了以后，该目录里面的 `js` 脚本，就被解释用 `ES6` 模块。

[APP内打开](#)

标签： 前端 架构 前端框架

 14 4 收藏



我要当架构师

为了提升自己的能力，我在某课网买了一个架构师的课程。 本专栏...

已订阅

相关小册



VIP Babel 插件通关秘籍

zxg_神说...

5480购买

¥49.9



VIP 玩转 CSS 的艺术之美

JowayYo...

4827购买

¥19.9

评论

输入评论 (Enter换行, Ctrl + Enter发送)

全部评论 4

最新

最热



haihaihai130

1天前

我觉得把 `在package.json 文件中, 指定 type 字段为 module`改成文章中第一个方法会好一点, 省去很多折腾, 我只是想把用了esModule的nodejs代码运行起来

👍 点赞 1



一尾流莺

23小时前

哈哈你说的有道理

👍 点赞 回复



GreenTea

1年前

APP内打开

👍 14

💬 4

★ 收藏



一尾流莺

1年前

是的,随着node的升级,自己就支持了

👍 点赞 💬 回复

相关推荐

前端技术小哥 4年前

NPM你真的会吗? (一)

👁 2437 👍 3 💬 2

_ihhu 4年前

babel 用法及其 .babelrc 的配置详解, 想做前端架构, 拒绝一知半解

👁 5544 👍 173 💬 3

CHu 1年前

用 TS 搭建 开源 前端函数工具库

👁 141 👍 1 💬 评论

mykurisu 10月前

H5可回溯方案浅析 -- 以保险投保为例

👁 1708 👍 9 💬 3

null 3年前

一文搞懂 Dynamic Import 和 Top-level await 提案

👁 3034 👍 36 💬 6

龙跃空 2年前

Vue项目中支持可选链（链判断运算符）操作

👁 3069 👍 1 💬 3

鹤云云 1年前

【译】reflect-metadata及相关ECMAScript提案简介

👁 463 👍 7 💬 评论

邵天宇Soy 3年前

前端面试官: 请使用二分法搜索旋转数组

APP内打开

👍 14

💬 4

☆ 收藏

👁 7540 👍 107 💬 10

VJ不是DJ 1年前

Node 模块规范鏖战：难以相容的 CJS 与 ESM

👁 2285 👍 12 💬 6

Jrainlau 5年前

深入认识vue-cli：能做的不仅仅是初始化vue工程

👁 4122 👍 229 💬 4

哈库拉马塔塔 9月前

vue-cli插件安装过程源码解析

👁 567 👍 点赞 💬 1

徐小超 1年前

CHALK - 美化 node 输出的神器

👁 2173 👍 3 💬 评论

思南 2年前

nexus、lerna搭建npm私有仓库

👁 2306 👍 2 💬 2

APP内打开