

全部开发者教程

13：扩展：所谓良好的`TypeScript`支持，是如何提供的？

14：总结

第三章：Vue 3源码结构 - 搭建框架雏形

01：前言

02：探索源码设计：Vue3 源码设计大解析

03：创建测试实例：在 Vue 源码中运行测试实例

04：跟踪解析运行行为：为 vue 开启 SourceMap

05：授人以鱼：如何针对源码进行 debugger

06：授人以渔：如何阅读源码

07：开始搭建自己的框架：创建 vue-next-mini

08：为框架进行配置：导入 ts

Sunday • 更新于 2022-10-19

◀ 上一节 03：创建测试实... 05：授人以鱼：... 下一节 ▶

04：跟踪解析运行行为：为 vue 开启 SourceMap

此时，我们已经成功的运行了一个测试实例代码，但是在这样的一个测试实例代码中，Vue 内部是如何执行的呢？

如果想要知道这个，那么我们需要对 vue 的代码进行 debugger 来跟踪 vue 代码的执行。

那么问题就来了，如何对 Vue 进行 debugger 操作呢？

如果想要对 Vue 进行 debugger 操作，那么我们必须开启 vue 的 source-map 功能。

开启 Vue 的 SourceMap

那么如何开启 Vue 的 source-map 呢？

1. 打开 package.json 可以发现，当我们执行 npm run build 时，其实执行的是 node scripts/build.js 指令

2. 这就意味着，它的配置文件读取的是 scripts/build.js 这个文件

3. 那么在该文件中存在这样的一行代码：

<> 代码块

1 ...
2 sourceMap ? `SOURCE_MAP:true` : ``
3 ...

4. 也就是说，这里的 sourceMap 变量，决定了 SOURCE_MAP:true 还是 "

5. 而这个值，最终会被设置到环境变量中，在 rollup.config.js 中，通过：

<> 代码块

1 output.sourcemaps = !!process.env.SOURCE_MAP

的形式，赋值给 output.sourcemaps

6. 而 output.sourcemaps 则决定了，最终的打包，是否会包含 source-map：

<> 代码块

1 sourceMap: output.sourcemaps

7. 所以，根据以上代码，只要 scripts/build.js 中的 sourceMap 变量的值为 true，则最终会打包包含 sourcemaps 的包。

8. 那么 sourceMap 变量的值是如何确定的呢？

9. 在 scripts/build.js 中，我们可以看到如下代码：

<> 代码块

1 const sourceMap = args.sourcemaps || args.s

10. 而 args 的值为：

索引目录

04：跟踪解析运行行为：为 vue 开启 SourceMap

<> 代码块

```
1 const args = require('minimist')(process.argv.slice(2))
```

11. 从代码可知, `args` 是 `minimist` 的导出对象。

12. 所以我们需要看下 `minimist` 这个依赖包是干什么呢?

13. 根据官网的实例代码可知:

<> 代码块

```
1 var argv = require('minimist')(process.argv.slice(2));
2 console.log(argv);
3
4 $ node example/parse.js -a beep -b boop
5 { _: [], a: 'beep', b: 'boop' }
```

14. 我们可以在执行 `npm` 指令时, 通过 `-a beep` 的形式为 `require('minimist')(process.argv.slice(2))`; 导出的值增加属性

15. 所以, 根据以上代码, 我们可以在 `package.json` 中修改 `build` 指令为:

<> 代码块

```
1 "build": "node scripts/build.js -s"
```

16. 其中的 `-s` 表示: 我们将为 `scripts/build.js` 文件中的 `args` 新增一个属性 `s`

17. 而这个 `s` 将决定了 `sourceMap` 常量的值为 `true`

18. 此时, 我们再执行 `npm run build` 可以发现, 打包出的所有文件都将包含一个 `xxxx.map` 文件

19. 这样我们就开启了源代码的 `source-map`

20. 有了 `source-map` 之后, 接下来我们就可以对代码进行 `debugger` 了。

03: 创建测试实例: 在 Vue 源码中运行测试... < 上一节 下一节 > 05: 授人以鱼: 如何针对源码进行 debugger

 我要提出意见反馈

