

全部开发者教程

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

09: 引入代码格式化工具：prettier 让你的代码结构更加规范

10: 模块打包器：rollup

11: 初见框架雏形：配置路径映射

12: 总结

第四章：响应系统 - 响应系

Sunday • 更新于 2022-10-19

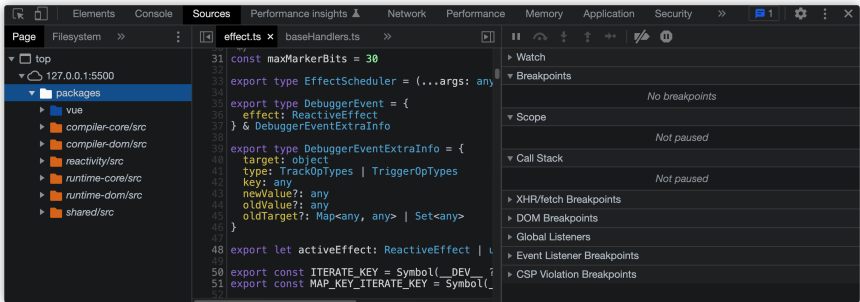
上一节 04: 跟踪解析运... 06: 授人以渔：... 下一节

05: 授人以鱼：如何针对源码进行 debugger

此时我们已经成功的开启了 SourceMap，那么开启了 sourceMap 之后有什么改变呢？

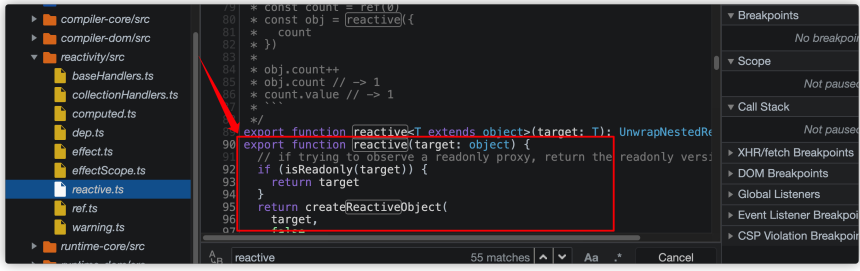
此时我们在来看刚才启动的项目。

在刚才启动的项目中，按 F12 打开控制台，进入 Sources 模块，此时可以看到如下内容：



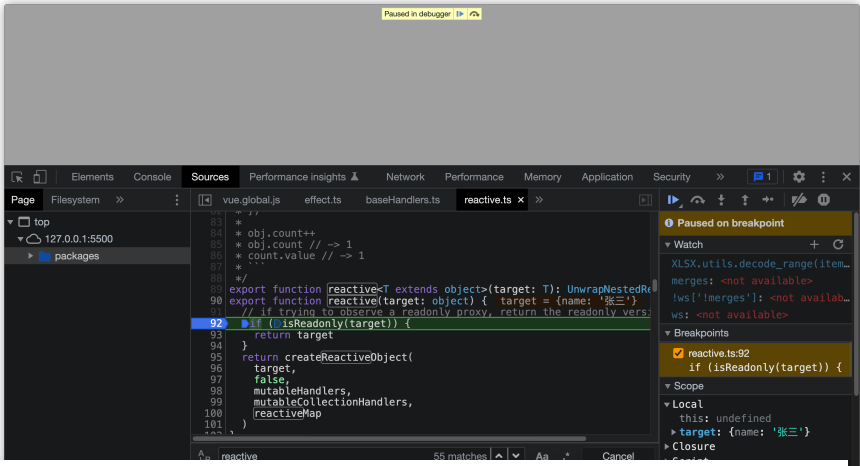
其中左侧所展示的，就是当前使用到的 vue 源代码。

那么我们知道此时我们是使用了 reactive 方法声明的响应式数据，reactive 方法对应的代码位置在 packages/reactivity/src/reactive.ts 中第 90 行：



那么此时我们就可以在这里打上一个断点，来跟踪整个 reactive 的代码执行逻辑。

刷新页面，可以看到，此时代码已经进入了 debugger



因为我们目前还没有开始学习 `vue` 源代码内容，所以我们此时先到此为止，大家也可以根据自己的情况，看一下是否要继续跟踪执行。

那么这样我们就已经成功的为 `vue` 的测试实例开启了 `debugger` 功能，后续我们的开发之中，就可以利用这样的方式，来跟踪并查看 `vue` 源码的执行逻辑。

那么最后我们再总结一下这几个小节所介绍的步骤：

想要对 `vue` 代码执行 `debugger` 那么共分为以下步骤：

1. 下载 `vue` 源代码，推荐通过 [该仓库](#) 下载指定版本（注意：**直接下载 ZIP 文件会导致 build 出错**）
2. 为源代码开启 `sourcemap`，以方便后续进行 `debugger`
3. 在 `packages/vue/examples` 中，创建文件，导入 `../../dist/vue.global.js`，书写测试实例
4. 通过 `Live Server` 启动服务
5. 在浏览器控制台的 `Sources` 中查看运行代码，并进行 `debugger`

04: 跟踪解析运行行为: 为 vue 开启 Sourc... ◀ 上一节 下一节 ▶ 06: 授人以渔: 如何阅读源码

 我要提出意见反馈

[企业服务](#) [网站地图](#) [网站首页](#) [关于我们](#) [联系我们](#) [讲师招募](#) [帮助中心](#) [意见反馈](#) [代码托管](#)



Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号



 意见反馈

 收藏教程

 标记书签