慕课网首页 免费课 实战课 体系课 慕课教程 专栏 手记 企业服务

Sunday • 更新于 2022-10-19

Q 🏋 💄 我的课程



Q

从所有教程的词条中查询…

◆ 上一节 02: 源码阅读: ... 04: 框架实现: ... 下一节 ▶

首页 > 慕课教程 > Vue3源码分析与构建方案 > 03:框架实现:构建 reactive 函数,获取 prox...

全部开发者教程 ः≡

uo. 応结

第五章:响应系统-初见 reactivity 模块

01: 前言

02: 源码阅读: reactive 的响应性,跟踪 Vue 3 源码实现逻辑

03: 框架实现: 构建 reactive 函数, 获取 proxy 实例

04:框架实现:什么是 WeakMap? 它和 Map 有什么 区别?

05: 框架实现: createGetter && createSetter

06: 热更新的开发时: 提升开 发体验

07: 框架实现: 构建 effect 函 数, 生成 ReactiveEffect 实例

08: 框架实现: track && trigger

09: 框架实现: 构建 track 依

03: 框架实现: 构建 reactive 函数, 获取 proxy 实例

根据上一小节的内容可知,整个 reactive 函数,本质上是返回了一个 proxy 实例,那么我们这一小节,就先去实现这个 reactive 函数,得到 proxy 实例。

1. 创建 packages/reactivity/src/reactive.ts 模块:

```
<> 代码块
     import { mutableHandlers } from './baseHandlers'
 2
 3
      * 响应性 Map 缓存对象
 4
      * key: target
 5
      * val: proxy
 6
     export const reactiveMap = new WeakMap<object, any>()
9
10
      * 为复杂数据类型, 创建响应性对象
11
      * @param target 被代理对象
12
13
      * @returns 代理对象
14
     export function reactive(target: object) {
16
         return createReactiveObject(target, mutableHandlers, reactiveMap)
17
18
19
20
      * 创建响应性对象
21
      * @param target 被代理对象
22
      * @param baseHandlers handler
23
2.4
     function createReactiveObject(
25
        target: object,
26
         baseHandlers: ProxyHandler<any>,
27
        proxyMap: WeakMap<object, any>
28
29
         // 如果该实例已经被代理,则直接读取即可
30
         const existingProxy = proxyMap.get(target)
         if (existingProxy) {
31
             return existingProxv
32
33
34
         // 未被代理则生成 proxy 实例
35
36
         const proxy = new Proxy(target, baseHandlers)
37
         // 缓存代理对象
38
39
         proxyMap.set(target, proxy)
40
         return proxy
41
     }
```

2. 创建 packages/reactivity/src/baseHandlers.ts 模块:

✓ 代码块1 /**✓ 收藏教程□ 标记书签

索引目录

03: 框架实现: 枚

?

··

```
4 */
export const mutableHandlers: ProxyHandler<object> = {}
```

- 3. 那么此时我们就已经构建好了一个基本的 reactive 方法,接下来我们可以通过 测试案例 测试一下.
- 4. 创建 packages/reactivity/src/index.ts 模块, 作为 reactivity 的入口模块

```
か代码块
1 export { reactive } from './reactive'
```

5. 在 packages/vue/src/index.ts 中, 导入 reactive 模块

```
c> 代码块

1     export { reactive } from '@vue/reactivity'
```

- 6. 执行 npm run build 进行打包, 生成 vue.js
- 7. 创建 packages/vue/examples/reactivity/reactive.html 文件, 作为测试实例:

```
<> 代码块
1 <!DOCTYPE html>
2 <html lang="en">
4
    <head>
     <meta charset="UTF-8">
5
 6
      <script src="../../dist/vue.js"></script>
    <script>
9
     const { reactive } = Vue
1.0
11
    const obj = reactive({
      name: '张三'
12
13 })
14
    console.log(obj);
15 </script>
16
    </html>
17
```

8. 运行到 Live Server 可见打印了一个 proxy 对象实例

那么至此我们已经得到了一个基础的 reactive 函数, 但是在 reactive 函数中我们还存在三个问题:

- 1. WeakMap 是什么? 它和 Map 有什么区别呢?
- 2. mutableHandlers 现在是一个空的, 我们又应该如何实现呢?
- 3. 难不成以后每次测试时,都要打包一次吗?

那么我们一个一个来看~~~

✔ 我要提出意见反馈

* * 6 6 4

?

 \odot