Sunday • 更新于 2022-10-19

上一节 02：源码阅读：... 04：框架实现：... 下一节

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

```ts
import { mutableHandlers } from './baseHandlers'

/**
 * 响应性 Map 缓存对象
 * key: target
 * val: proxy
 */
export const reactiveMap = new WeakMap<object, any>()

/**
 * 为复杂数据类型，创建响应性对象
 * @param target 被代理对象
 * @returns 代理对象
 */
export function reactive(target: object) {
    return createReactiveObject(target, mutableHandlers, reactiveMap)
}

/**
 * 创建响应性对象
 * @param target 被代理对象
 * @param baseHandlers handler
 */
function createReactiveObject(
    target: object,
    baseHandlers: ProxyHandler<any>,
    proxyMap: WeakMap<object, any>
) {
    // 如果该实例已经被代理，则直接读取即可
    const existingProxy = proxyMap.get(target)
    if (existingProxy) {
        return existingProxy
    }

    // 未被代理则生成 proxy 实例
    const proxy = new Proxy(target, baseHandlers)

    // 缓存代理对象
    proxyMap.set(target, proxy)
    return proxy
}
```

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

代码块

```ts
/**
```

```
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

代码块

```
1     export { reactive } from './reactive'
```

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

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">
3
4   <head>
5     <meta charset="UTF-8">
6     <script src="../../dist/vue.js"></script>
7   </head>
8   <script>
9     const { reactive } = Vue
10
11    const obj = reactive({
12      name: '张三'
13    })
14    console.log(obj);
15  </script>
16
17  </html>
```

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~~~

02：源码阅读：reactive 的响应性，跟踪 Vu... ◂ 上一节　　下一节 ▸ 04：框架实现：什么是 WeakMap？它和 M...

我要提出意见反馈