

全部开发者教程

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 依赖收集函数

10: 框架实现：构建 trigger 触发依赖

11: 总结：单一依赖的 reactive

12: 功能升级：响应数据对应多个 effect

13: 框架实现：构建 Dep 模块，处理一对多的依赖关系

14: reactive 函数的局限性

15: 总结

第六章：响应系统 - ref 的响应性

01: 前言



Sunday • 更新于 2022-10-19

上一节 11: 总结：单一... 13: 框架实现：... 下一节

12: 功能升级：响应数据对应多个 effect

在我们之前的实现中，还存在一个小的问题，那就是：每个响应性数据属性只能对应一个 effect 回调

我们来看下面这个例子 packages/vue/examples/reactivity/reactive-dep.html：

<> 代码块

```
1 <body>
2   <div id="app">
3     <p id="p1"></p>
4     <p id="p2"></p>
5   </div>
6 </body>
7
8 <script>
9   const { reactive, effect } = Vue
10
11   const obj = reactive({
12     name: '张三'
13   })
14
15   // 调用 effect 方法
16   effect(() => {
17     document.querySelector('#p1').innerText = obj.name
18   })
19   effect(() => {
20     document.querySelector('#p2').innerText = obj.name
21   })
22
23   setTimeout(() => {
24     obj.name = '李四'
25   }, 2000);
26 </script>
```

在以上的代码中，我们新增了一个 effect 函数，即：name 属性对应两个 DOM 的变化。

但是当我们运行该代码时发现，p1 的更新渲染是无效的！

那么这是因为什么呢？

查看我们的代码可以发现，我们在构建 KeyToDepMap 对象时，它的 Value 只能是一个 ReactiveEffect，所以这就导致了一个 key 只能对应一个有效的 effect 函数。

那么假如我们期望：一个 key 可以对应多个有效的 effect 函数的话，那么应该怎么做呢？

可能有些同学已经想到了，我们只需要让 KeyToDepMap 的 Value 可以对应一个数组 不就可以了吗？

索引目录

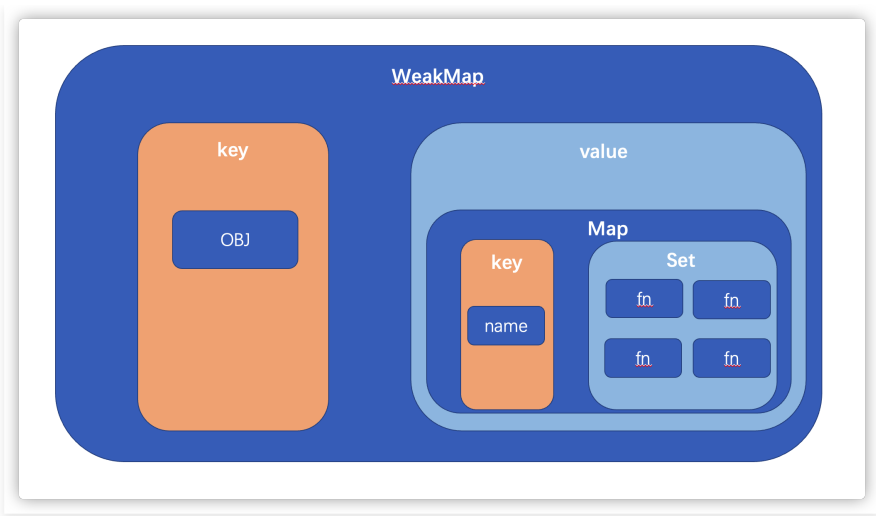
12: 功能升级：响



意见反馈

收藏教程

标记书签



如上图所示，我们可以构建一个 `Set`（`set` 是一个“数组”，值不会重复）类型的对象，作为 `Map` 的 `value`。

我们可以把它叫做 `Dep`，通过 `Dep` 来保存 **指定 key 的所有依赖**

那么明确好了这样的概念之后，接下来我们到项目中，进行一个对应的实现。

11: 总结：单一依赖的 reactive ◀ 上一节 下一节 ▶ 13: 框架实现：构建 `Dep` 模块，处理一对多...

 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