

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 07：框架实现：... 09：框架实现：... 下一节 ▶

08：框架实现：track && trigger

根据我们在 `packages/reactivity/src/baseHandlers.ts` 中的代码可知，当触发 `getter` 行为时，其实我们会触发 `track` 方法，进行 **依赖收集**，当触发 `setter` 行为时，会触发 `trigger` 方法，来 **触发依赖**

那么这里就涉及到了两个概念：

1. 依赖收集：`track`
2. 触发依赖：`trigger`

所以接下来如果我们想要实现这两个函数，那么就需要先搞清楚什么是 **依赖收集** 和 **触发依赖**。

什么是响应性

根据大家的开发经验和我们在第二小节查看源码时可知，所谓的响应性其实指的就是：**当响应性数据触发 `setter` 时执行 `fn` 函数**

那么想要达到这样的一个目的，那就必须要在：`getter` 时能够收集当前的 `fn` 函数，以便在 `setter` 的时候可以执行对应的 `fn` 函数

但是对于收集而言，如果仅仅是把 `fn` 存起来还是不够的，我们还需要知道，当前的这个 `fn` 是**哪个响应式数据对象的哪个属性**对应的，只有这样，我们才可以在 **该属性** 触发 `setter` 的时候，准确的执行响应性。

那么我们应该如何确定以这一点呢？

如何进行依赖收集

大家还记不记得，我们在 `packages/reactivity/src/reactive.ts` 中创建过一个 `WeakMap`：

<> 代码块

```
1 export const reactiveMap = new WeakMap<object, any>()
```

我们知道 `WeakMap` 它的 `key` 必须是一个对象，并且 `key` 是一个弱引用的。

那么大家想一想我们可不可以这样：

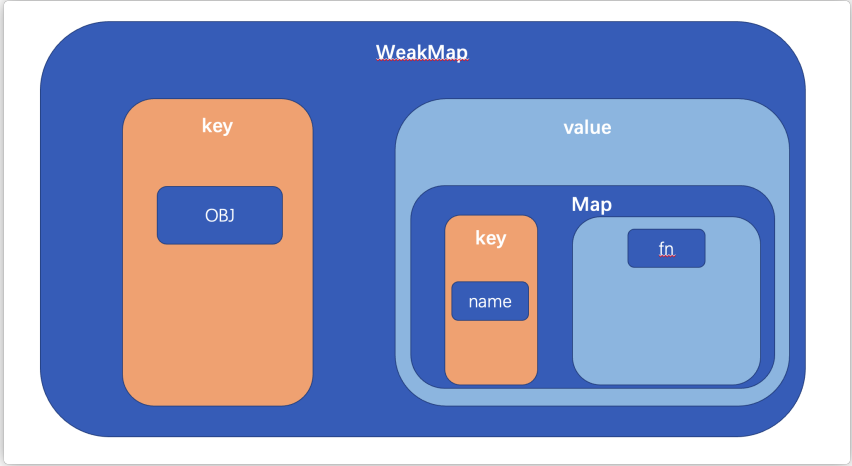
1. `WeakMap`：
 1. `key`：响应性对象
 2. `value`：`Map` 对象
2. `Map`：
 1. `key`：响应性对象的指定属性
 2. `value`：指定对象的指定属性的 执行函数

图表表示：

索引目录

- 08：框架实现：tr
- 什么是响应性
- 如何进行依赖收





那么这样我们就可以关联上 **指定对象的指定属性** 与 **执行函数** `fn` 之间的关系，当触发 `setter` 时，直接执行 **对应对象的对应属性的** `fn` 即可。

那么明确好了这样的一个概念之后，接下来我们就可以根据以上理念进行对应的实现。

07: 框架实现：构建 effect 函数，生成 Rea... [← 上一节](#) [下一节](#) 09: 框架实现：构建 track 依赖收集函数

[✎ 我要提出意见反馈](#)