

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

10：框架实现：... 12：功能升级：... 下一节 >

11：总结：单一依赖的 reactive

通过以上的努力，我们目前已经构建了一个简单的 reactive 函数，使用 reactive 函数，配合 effect 可以实现出一个 响应式数据渲染功能，那么这一小节，我们把整个的流程做一个总结：

1. 首先我们在 packages/reactivity/src/reactive.ts 中，创建了一个 reactive 函数，该函数可以帮助我们生成一个 proxy 实例对象

2. 通过该 proxy 实例的 handler 可以监听到对应的 getter 和 setter

3. 然后我们在 packages/reactivity/src/effect.ts 中，创建了一个 effect 函数，通过该函数可以创建一个 ReactiveEffect 的实例，该实例的构造函数可以接收传入的回调函数 fn，并且提供了一个 run 方法

4. 触发 run 可以为 activeEffect 进行赋值，并且执行 fn 函数

5. 我们需要在 fn 函数中触发 proxy 的 getter，以此来激活 handler 的 get 函数

6. 在 handler 的 get 函数中，我们通过 WeakMap 收集了 指定对象，指定属性 的 fn，这样的一步操作，我们把它叫做 依赖收集

7. 最后我们可以在 任意时刻，修改 proxy 的数据，这样会触发 handler 的 setter

8. 在 handler 的 setter 中，我们会根据 指定对象 target 的 指定属性 key 来获取到保存的 依赖，然后我们只需要触发依赖，即可达到修改数据的效果

10：框架实现：构建 trigger 触发依赖 < 上一节 下一节 > 12：功能升级：响应数据对应多个 effect

我要提出意见反馈

索引目录

11：总结：单一依

Q ? 手机 评论