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

Q 📜 💄 我的课程



Q

从所有教程的词条中查询…

首页 > 慕课教程 > Vue3源码分析与构建方案 > 04:框架实现:基于 renderer 完成 ELEMENT ...

全部开发者教程

 \equiv

11:框架实现:完成虚拟节点 下的 class 和 style 的增强

12: 总结

第十章:runtime 运行时 -构建 renderer 渲染器

01: 前言

基本架构

02:源码阅读:初见 render 函数,ELEMENT 节点的挂载 操作

03: 框架实现: 构建 renderer

04:框架实现:基于 renderer 完成 ELEMENT 节点挂载

05: 框架实现: 合并渲染架构, 得到可用的 render 函数

06:源码阅读:渲染更新, ELEMENT 节点的更新操作

07:框架实现:渲染更新, ELEMENT 节点的更新实现

08. 源码阅读:新旧节占不同

索引目录

04: 框架实现: 基

?

··

04:框架实现:基于 renderer 完成 ELEMENT 节点挂载

根据源码我们知道 Element 的挂载主要依赖于 processElement 方法, 所以我们可以直接构建该方法。

1. 在 packages/runtime-core/src/renderer.ts 中, 创建 processElement 方法:

```
<> 代码块
         /**
 1
          * Element 的打补丁操作
 2
         const processElement = (oldVNode, newVNode, container, anchor) => {
             if (oldVNode == null) {
                 // 挂载操作
                 mountElement(newVNode, container, anchor)
 8
             } else {
                 // TODO: 更新操作
 9
10
11
         }
12
13
          * element 的挂载操作
14
15
         const mountElement = (vnode, container, anchor) => {
16
17
             const { type, props, shapeFlag } = vnode
18
19
             // 创建 element
20
             const el = (vnode.el = hostCreateElement(type))
2.1
22
             if (shapeFlag & ShapeFlags.TEXT_CHILDREN) {
                 // 设置 文本子节点
23
24
                  hostSetElementText(el, vnode.children as string)
             } else if (shapeFlag & ShapeFlags.ARRAY_CHILDREN) {
25
26
                  // TODO: 设置 Array 子节点
27
2.8
             // 处理 props
29
30
             if (props) {
                 // 遍历 props 对象
31
32
                 for (const key in props) {
33
                     hostPatchProp(el, key, null, props[key])
34
                 }
             }
35
36
37
             // 插入 el 到指定的位置
38
             hostInsert(el, container, anchor)
39
40
         const patch = (oldVNode, newVNode, container, anchor = null) => {
41
             if (oldVNode === newVNode) {
42
43
                 return
45
46
             const { type, shapeFlag } = newVNode
47
             switch (type) {
48
                 case Text:
```

```
51
               case Comment:
52
                   // TODO: Comment
53
                   break
54
                case Fragment:
                   // TODO: Fragment
55
                    break
56
57
                default:
                    if (shapeFlag & ShapeFlags.ELEMENT) {
59
                      processElement(oldVNode, newVNode, container, anchor)
                    } else if (shapeFlag & ShapeFlags.COMPONENT) {
60
61
                      // TODO: 组件
62
63
            }
```

根据源码的逻辑, 我们在这里主要做了五件事情:

- 1. 区分挂载、更新
- 2. 创建 Element
- 3. 设置 text
- 4. 设置 class
- 5. 插入 DOM 树

✔ 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



·