手记





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 > 05:框架实现:合并渲染架构,得到可用的 ren...

全部开发者教程 Ξ

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

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 基本架构

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

Sunday • 更新于 2022-10-19

◆ 上一节 04: 框架实现: ... 06: 源码阅读: ... 下一节 →

05: 框架实现: 台 源代码

实现

?

··

索引目录

05: 框架实现: 合并渲染架构,得到可用的 render 函数

现在三大模块中,该完成的业务代码我们都已经完成了。但是还存在一个问题,那就是我们应该 如何使 用 render 函数呢?

我们知道,在源码中,我们可以直接:

```
<> 代码块
     const { render } = Vue
2
     render(vnode, document.querySelector('#app'))
```

但是我们来看咱们现在的代码,发现是 不可以 直接这样导出并使用的。

那么 vue 是怎么做的呢?

源代码

查看源代码,我们可以发现在 packages/runtime-dom/src/index.ts 中,存在这样的一段代码:

```
<> 代码块
     export const render = ((...args) => {
      ensureRenderer().render(...args)
     }) as RootRenderFunction<Element | ShadowRoot>
```

而这里导出的 render 就是我们解构出来的 render。

在这段代码中涉及到一个方法 ensureRenderer, 我们查看它的实现, 可以发现:

```
<> 代码块
     function ensureRenderer() {
       return (
         renderer ||
         (renderer = createRenderer<Node, Element | ShadowRoot>(rendererOptions))
5
6
     }
```

它触发了我们导出的 createRenderer , 并且传递了一个 rendererOptions 参数 , 查看 rendererOption s:

```
()代码块
     const rendererOptions = /*#__PURE__*/ extend({ patchProp }, nodeOps)
```

可以发现 rendererOptions 就是一个合并了 prop 和 nodeOps 的对象。

这样:我们就成功的把 options 参数传递到了 createRenderer 函数中,得到了 renderer 渲染器,从 而最终拿到 render 函数了。

那么明确好了源代码的实现之后,下面我们就尝试自己实现一下。

实现



```
Tumboile / ciseacevendeise. } illoum (mana\indictue-coise
    import { extend } from '@vue/shared'
    import { nodeOps } from './nodeOps'
    import { patchProp } from './patchProp'
    5
        const rendererOptions = extend({ patchProp }, nodeOps)
    6
    7
        let renderer
    8
    9
        function ensureRenderer() {
    10
          return renderer || (renderer = createRenderer(rendererOptions))
    11
    12
    13 export const render = (...args) \Rightarrow \{
    14
          ensureRenderer().render(...args)
    16
2. 在 packages/runtime-core/src/index.ts 中导出 createRenderer
3. 在 packages/vue/src/index.ts 中导出 render
4. 在测试实例 packages/vue/examples/runtime/render-element.html :
    <> 代码块
    1 <script>
         const { h, render } = Vue
    3
    4
         const vnode = h('div', {
    5
           class: 'test'
        }, 'hello render')
     6
        console.log(vnode);
    8
    10
        render(vnode, document.querySelector('#app'))
    11 </script>
vnode 可以正常渲染。
   ▶ 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

<> 代码块

★ % 6

?

0

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

口 标记书签