

全部开发者教程

慕课网 · 更新行力

26: 总结

第十一章：runtime 运行时 - 组件的设计原理与渲染方案

01: 前言

02: 源码阅读：无状态基础组件挂载逻辑

03: 框架实现：完成无状态基础组件的挂载

04: 源码阅读：无状态基础组件更新逻辑

05: 局部总结：无状态组件的挂载、更新、卸载总结

06: 源码阅读：有状态的响应性组件挂载逻辑

07: 框架实现：有状态的响应性组件挂载逻辑

08: 源码阅读：组件生命周期回调处理逻辑

09: 框架实现：组件生命周期回调处理逻辑



Sunday • 更新于 2022-10-19

◀ 上一节 06：源码阅读：无状态基础组件挂载逻辑 08：源码阅读：无状态基础组件更新逻辑 下一节 ▶

07：框架实现：有状态的响应性组件挂载逻辑

那么明确好了有状态组件的挂载逻辑之后，我们接下来就进行对应的实现。

1. 在 `packages/runtime-core/src/component.ts` 中，新增 `applyOptions` 方法，为 `instance` 赋值 `data`：

<> 代码块

```
1 function applyOptions(instance: any) {
2   const { data: dataOptions } = instance.type
3
4   // 存在 data 选项时
5   if (dataOptions) {
6     // 触发 dataOptions 函数，拿到 data 对象
7     const data = dataOptions()
8     // 如果拿到的 data 是一个对象
9     if (isObject(data)) {
10      // 则把 data 包装成 reactive 的响应性数据，赋值给 instance
11      instance.data = reactive(data)
12    }
13  }
14 }
```

2. 在 `finishComponentSetup` 方法中，触发 `applyOptions`：

<> 代码块

```
1 export function finishComponentSetup(instance) {
2   const Component = instance.type
3
4   instance.render = Component.render
5
6   // 改变 options 中的 this 指向
7   applyOptions(instance)
8 }
```

3. 在 `packages/runtime-core/src/componentRenderUtils.ts` 中，为 `render` 的调用，通过 `call` 方法修改 `this` 指向：

<> 代码块

```
1 /**
2  * 解析 render 函数的返回值
3  */
4 export function renderComponentRoot(instance) {
5   + const { vnode, render, data } = instance
6
7   let result
8   try {
9     // 解析到状态组件
10    if (vnode.shapeFlag & ShapeFlags.STATEFUL_COMPONENT) {
11      // 获取到 result 返回值，如果 render 中使用了 this，则需要修改 this 指向
12      + result = normalizeVNode(render!.call(data))
13    }
14  } catch (err) {
```

索引目录

07：框架实现：有



```
17
18     return result
19 }
```

至此，代码完成。

我们可以创建对应测试实例 `packages/vue/examples/runtime/render-comment-data.html`：

<> 代码块

```
1  <script>
2    const { h, render } = Vue
3
4    const component = {
5      data() {
6        return {
7          msg: 'hello component'
8        }
9      },
10     render() {
11       return h('div', this.msg)
12     }
13   }
14
15   const vnode = h(component)
16   // 挂载
17   render(vnode, document.querySelector('#app'))
18 </script>
```

可以发现，渲染成功。

06: 源码阅读: 有状态的响应性组件挂载逻辑 < 上一节 下一节 > 08: 源码阅读: 组件生命周期回调处理逻辑

 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