慕课网首页 免费课 慕课教程 专栏 手记 实战课 体系课 企业服务

Sunday • 更新于 2022-10-19

Ē 我的课程

Q

从所有教程的词条中查询…

↓ 上一节 02: 基于 rende...04: 总结 下一节 →

首页 > 慕课教程 > Vue3源码分析与构建方案 > 03: 基于 template 渲染的 createApp 的构建...

全部开发者教程 Ξ

06:响应性数据的编辑器处

理: generate 生成 render 函

07: 响应性数据的编辑器处

理: render 函数的执行处理

08: 多层级模板的编辑器处

理: 多层级的处理逻辑

09: 基于编辑器的指令(v-xx) 处理: 指令解析的整体逻辑

10: 基干编辑器的指令(v-xx) 处理: AST 解析逻辑 (困难)

11: 基于编辑器的指令(v-xx) 处理: JavaScript AST , 构建 vif 转化模块 (困难)

12: 基于编辑器的指令(v-xx) 处理: JavaScript AST, transform 的转化逻辑

13: 基干编辑器的指令(v-xx) 处理: 生成 render 函数

14 . 总结

```
03: 基于 template 渲染的 createApp 的构建逻辑
```

对于组件而言, 我们不光要支持 render 还需要支持 template 的模板渲染:

```
<>代码块
     <script>
      const { createApp, h } = Vue
      const APP = {
        template: `<div>hello world</div>`
5
6
7
      const app = createApp(APP)
8
       app.mount('#app')
     </script>
```

但是如果现在我们尝试运行这个测试实例的话,那么将会得到一个对应的错误:

```
TypeError: Cannot read properties of undefined (reading 'call') at renderComponentRoot (<a href="mailto:componentRenderUtils.ts:15:36">componentRenderUtils.ts:15:36</a>) at ReactiveEffect.componentUpdateFn [as fn] (<a href="mailto:renderer.ts:178">renderer.ts:178</a>; at ReactiveEffect.run (<a href="mailto:effect.ts:130:15">effect.ts:130:15</a>)
                                                                                                  fn] (renderer.ts:178:41)
                                          late (renderer.ts:221:50
                                                    t (<u>renderer.ts:224:3</u>)
                    ountComponent (renderer.ts:158:3
                      cessComponent (renderer.ts:144:4)
ch (renderer.ts:572:6)
der (renderer.ts:592:4)
                        t (<u>apiCreateApp.ts:16:5</u>)
► Uncaught TypeError: Cannot read properties of undefined (reading 'type') at patch (renderer.ts:553:39)
                       ctiveEffect.componentUpdateFn [as fn] (renderer.ts:181:5)
ctiveEffect.run (effect.ts:130:15)
tance.update (renderer.ts:221:50)
                              enderEffect (renderer.ts:224:3)
omponent (renderer.ts:158:3)
                            ssComponent (renderer.ts:144:4)
(renderer.ts:572:6)
r (renderer.ts:592:4)
                             (apiCreateApp.ts:16:5)
```

查看我们当前的代码,我们可以发现:

- 1. 在 createAppAPI 中的 mount 函数中,我们直接触发了 createVNode ,传递了当前的组件实例,从 而得到 vnode。
- 2. 但是我们知道,我们当初是先实现了 render ,后实现了 compiler ,这也就意味着在 renderer 渲 染器中,是**不存在**模板解析器的。
- 3. 所以也就意味着, createVNode 将无法解析 template 模板。

那么这样应该怎么做呢?

template 模板的解析,必然需要依赖于 compiler ,所以这就意味着,我们需要在 renderer 中导入 compiler 才可以。

1. 在 packages/runtime-core/src/component.ts 中, 创建 registerRuntimeCompiler 方法, 获取 co mpile 实例

<> 代码块 ▶ 意见反馈 ♡ 收藏教程 □ 标记书签

索引目录

03: 基干 templa

?

··

```
3 */
4 let compile
5
6 /**
7 * 用来注册编译器的运行时
8 */
9 export function registerRuntimeCompiler(_compile: any) {
10 compile = _compile
11 }
```

2. 接下来我们需要在 finishComponentSetup , 判断当前组件是 template 还是 render , 从而通过不同的方式进行渲染:

```
<> 代码块
    export function finishComponentSetup(instance) {
     const Component = instance.type
      // 组件不存在 render 时,才需要重新赋值
4
      if (!instance.render) {
5
          // 存在编辑器,并且组件中不包含 render 函数,同时包含 template 模板,则直接使用编辑
 6
    + if (compile && !Component.render) {
    + if (Component.template) {
    + // 这里就是 runtime 模块和 compile 模块结合点
    + const template = Component.template
10
11
    + Component.render = compile(template)
12
13
     __
// 为 render 赋值
·
14
15
          instance.render = Component.render
16
17
      // 改变 options 中的 this 指向
18
      applyOptions(instance)
19
20 }
```

3. 最后我们只需要在 packages/vue-compat/src/index.ts 中注册 compile 即可:

```
/**
2 * 注册 compiler
3 */
4 registerRuntimeCompiler(compileToFunction)
```

此时,再次运行测试实例,实例可正常运行。

✔ 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

* % 6 4

?

 \odot

Copyright © 2022 imooc.com All Rights Reserved | 京ICP备 12003892号-11 京公网安备11010802030151号