

全部开发者教程

06: 响应性数据的编辑器处理: generate 生成 render 函数

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: 基于编辑器的指令(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: 基于编辑器的指令(v-xx)处理: 生成 render 函数

14: 总结



Sunday • 更新于 2022-10-19

◀ 上一节 02: 基于 rende... 04: 总结 下一节 ▶

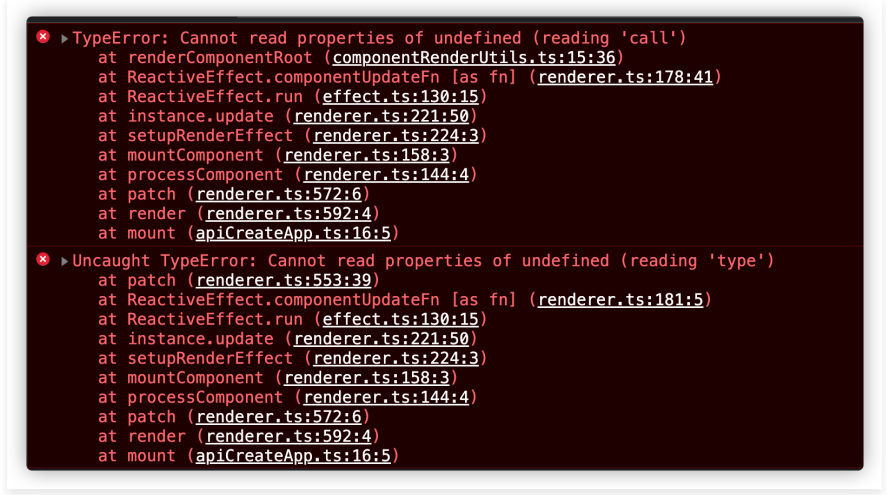
03: 基于 template 渲染的 createApp 的构建逻辑

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

```
1  <script>
2    const { createApp, h } = Vue
3    const APP = {
4      template: `<div>hello world</div>`
5    }
6
7    const app = createApp(APP)
8    app.mount('#app')
9  </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：



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 compiler 实例

<> 代码块

索引目录

03: 基于 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
1   export function finishComponentSetup(instance) {
2     const Component = instance.type
3
4     // 组件不存在 render 时，才需要重新赋值
5     if (!instance.render) {
6       + // 存在编辑器，并且组件中不包含 render 函数，同时包含 template 模板，则直接使用编译
7       + if (compile && !Component.render) {
8       +   if (Component.template) {
9       +     // 这里就是 runtime 模块和 compile 模块结合点
10      +     const template = Component.template
11      +     Component.render = compile(template)
12      +   }
13      + }
14      // 为 render 赋值
15      instance.render = Component.render
16    }
17
18    // 改变 options 中的 this 指向
19    applyOptions(instance)
20  }

```

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

```

<> 代码块
1   /**
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。

02: 基于 render 渲染的 createApp 的构建... < 上一节 下一节 > 04: 总结

 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