Sunday • 更新于 2022-10-19

上一节 09：框架实现：… 11：框架实现：… 下一节

10：源码阅读：生命回调钩子中访问响应性数据

在实际开发中，我们通常都会在生命周期钩子中访问响应式数据，比如我们来看如下测试实例 `packages/vue/examples/imooc/runtime/redner-component-hook-data.html`：

代码块

```
<script>
  const { h, render } = Vue

  const component = {
    data() {
      return {
        msg: 'hello component'
      }
    },
    render() {
      return h('div', this.msg)
    },
    // 组件实例处理完所有与状态相关的选项之后
    created() {
      console.log('created', this.msg);
    },
    // 组件被挂载之后
    mounted() {
      console.log('mounted', this.msg);
    },
  }

  const vnode = h(component)
  // 挂载
  render(vnode, document.querySelector('#app'))
</script>
```

这样的一个代码，在我们的 `vue-next-mini` 中是无法打印出对应数据的。

那么本小节我们期望的就是：**如何可以在生命钩子中访问响应性数据。**

对于这样的一个需求，大家应该是感觉有一点似曾相识的。不知道大家还记不记得，我们之前在 `render` **函数中访问过** `this.msg`。当时的做法是通过 `call` **方法改变了** `this` **指向。**

那么对于当前的场景而言，也是一样的。

我们这里分别去看 `created` 和 `mounted`

created

通过之前的代码我们已经知道， `created` 的回调是在 `applyOptions` 中触发的，所以我们可以直接在这里进行 `debugger`：

1. 进入 `applyOptions`
2. 剔除之前相同的逻辑，代码执行 `if (created) {...}`
    1. 进入 `if`，触发 `callHook` 方法

代码块

```
isArray(hook)
        ? hook.map(h => h.bind(instance.proxy!))
        : hook.bind(instance.proxy!),
```

3. 因为我们当前的 `hook` 不存在数组的情况，所以，我们直接看 `hook.bind(instance.proxy!)` 即可

    1. 对于 bind 方法，大家应该也是熟悉的，它会改变 `this` 指向，并且返回一个方法的引用
    2. 那么换句话而言，也就是在 **此时，改变了 `this` 指向**，和 `render` 一样，我们通过一个新的 `this` 指向了数据源

由以上代码可知：

1. 对于 `created` 而言，想要在它这里访问响应式数据，我们只需要通过 `bind` 改变 `this` 指向即可。

mounted

对于 `mounted` 而言，我们知道它的 **生命周期注册** 是在 `applyOptions` 方法内的 `registerLifecycleHook` 方法中，我们可以直接来看一下源码中的 `registerLifecycleHook` 方法：

代码块

```
function registerLifecycleHook(
  register: Function,
  hook?: Function | Function[]
) {
  if (isArray(hook)) {
    hook.forEach(_hook => register(_hook.bind(publicThis)))
  } else if (hook) {
    register((hook as Function).bind(publicThis))
  }
}
```

该方法中的逻辑非常简单，可以看到它和 `created` 的处理几乎一样，都是通过 `bind` 方法来改变 `this` 指向

由以上代码可知：

1. 无论是 `created` 也好，还是 `mounted` 也好，本质上都是通过 `bind` 方法来修改 `this` 指向，以达到在回调钩子中访问响应式数据的目的。

09：框架实现：组件生命周期回调处理逻辑 ◂ 上一节　　下一节 ▸ 11：框架实现：生命回调钩子中访问响应性...

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

意见反馈　收藏教程　标记书签