Sunday • 更新于 2022-10-19

上一节 10：源码阅读：… 12：源码阅读：… 下一节

11：框架实现：生命回调钩子中访问响应性数据

根据上一小节的描述，我们只需要 **改变生命周期钩子的 `this` 指向即可**

1. 在 `packages/runtime-core/src/component.ts` 中为 `callHook` 方法增加参数，以此来改变 `this` 指向：

代码块

```
/**
 * 触发 hooks
 */
function callHook(hook: Function, proxy) {
    hook.bind(proxy)()
}
```

2. 在 `applyOptions` 方法中为 `callHook` 的调用，传递第二个参数：

代码块

```
// hooks
if (beforeCreate) {
    callHook(beforeCreate, instance.data)
}

...

// hooks
if (created) {
    callHook(created, instance.data)
}
```

3. 在 `registerLifecycleHook` 中，为 `hook` 修改 `this` 指向

代码块

```
function registerLifecycleHook(register: Function, hook?: Function) {
    register(hook?.bind(instance.data), instance)
}
```

至此，代码完成。

创建对应测试实例 `packages/vue/examples/runtime/redner-component-hook-data.html`：

代码块

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

  const component = {
    data() {
      return {
        msg: 'hello component'
      }
    },
    render() {
```

```
13        // 组件实例处理完所有与状态相关的选项之后
14        created() {
15          console.log('created', this.msg);
16        },
17        // 组件被挂载之后
18        mounted() {
19          console.log('mounted', this.msg);
20        },
21      }
22
23      const vnode = h(component)
24      // 挂载
25      render(vnode, document.querySelector('#app'))
26    </script>
```

数据访问成功

10：源码阅读：生命回调钩子中访问响应性... 上一节 下一节 12：源码阅读：响应性数据改变，触发组件...