



the app instance of a Vue custom element
解释这个概念？

“the app instance of a Vue custom element”（Vue 自定义元素的应用实例）这个概念，涉及 Vue 在通过 `defineCustomElement()` 创建 Web Component（自定义元素）时，Vue 内部如何管理与之相关联的“应用实例”。

我们拆开解释这个短语：

一、Vue Custom Element（自定义元素）

Vue 3 支持将组件封装为原生 Web Component，即所谓的“自定义元素（custom element）”，通常通过如下方式生成：

jsCopy

```
import { defineCustomElement } from 'vue'
import MyComponent from './MyComponent.ce.vue'

const MyElement = defineCustomElement(MyComponent)
customElements.define('my-element', MyElement)
```

这会创建一个 **符合 Web Components 规范的类** `MyElement`，你可以像使用普通 HTML 元素一样使用它：

htmlCopy

```
<my-element></my-element>
```

二、App Instance（应用实例）

在标准 Vue 应用中，我们通常这么创建一个“应用实例”：

jsCopy

```
const app = createApp(App)
app.mount('#app')
```

这个 `app` 是一个 **Vue 应用实例**，包含组件注册、插件安装、全局配置等功能。

三、Vue Custom Element 的 App Instance 是什么？

当你使用 `defineCustomElement()` 定义组件时，Vue 会在**内部自动创建一个轻量级的应用实例**（app instance），用于支撑这个组件的生命周期和响应式逻辑。

虽然你没有显式调用 `createApp()`，但 Vue 实际上在 custom element 被挂载到 DOM 时，会在背后执行等效逻辑：

- 自动创建一个 app 实例
- 把你的组件视为根组件挂载上去
- 建立响应式系统
- 执行 `setup()`、生命周期钩子等

四、这个“内部 app instance”有什么用？

它的作用是：

1. 支撑该自定义元素的完整 Vue 生命周期；
2. 确保每个 `<my-element>` 是“自包含的、隔离的”实例，不会污染全局；
3. 让你可以像用普通 HTML 元素那样用 Vue 组件，同时保留响应式能力。

你一般不需要直接接触这个 app instance，但**理解它存在**，能帮助你更好地掌握：

- 每个 custom element 是一个**独立 Vue 应用**；
- 它拥有自己的响应式作用域；
- 它不会继承你在外部 `createApp` 中注册的插件或组件，除非手动注入。

总结

“the app instance of a Vue custom element”指的是：
Vue 内部为每个通过 `defineCustomElement()` 创建的自定义元素实例，自动生成并维护的 Vue 应用实例，它负责驱动该组件的响应式逻辑与生命周期，是该元素“背后的 Vue 引擎”。

如你希望进一步控制这个 app instance（例如注入插件、提供依赖等），可以在定义 custom element 时通过配置实现。

需要我举例说明如何注入全局依赖或与主应用交互吗？

