

Vue3 状态管理 - Pinia

1. 什么是Pinia

Pinia 是 Vue 的专属的最新状态管理库，是 Vuex 状态管理工具的替代品



1. 提供更加简单的API（去掉了 mutation）
2. 提供符合组合式风格的API（和 Vue3 新语法统一）
3. 去掉了 modules 的概念，每一个 store 都是一个独立的模块
4. 配合 TypeScript 更加友好，提供可靠的类型推断

2. 手动添加Pinia到Vue项目

后面在实际开发项目的时候，Pinia可以在项目创建时自动添加，现在我们初次学习，从零开始：

1. 使用 Vite 创建一个空的 Vue3项目

```
npm init vite@latest
```

2. 按照官方文档安装 pinia 到项目中

3. Pinia基础使用

1. 定义store
2. 组件使用store

定义Store (state + action)

```
1 import { defineStore } from 'pinia'
2 import { ref } from 'vue'
3 export const useCounterStore = defineStore('counter', () => {
4   // 数据 (state)
5   const count = ref(0)
6
7   // 修改数据的方法 (action)
8   const increment = () => {
9     count.value++
10  }
11
12  // 以对象形式返回
13  return { count, increment }
14 })
```



组件使用Store

```
1 <script setup>
2   // 1. 导入 `useCounterStore` 方法
3   import { useCounterStore } from '@/stores/counter'
4   // 2. 执行方法得到counterStore对象
5   const counterStore = useCounterStore()
6 </script>
7
8 <template>
9   <button @click="counterStore.increment">
10     {{ counterStore.count }}
11   </button>
12 </template>
13
```

4. getters实现

Pinia中的 getters 直接使用 computed函数 进行模拟, 组件中需要使用需要把 getters return出去

```
1 // 数据 (state)
2 const count = ref(0)
3
4 // getter
5 const doubleCount = computed(() => count.value * 2)
```

5. action异步实现

方式：异步action函数的写法和组件中获取异步数据的写法完全一致

- 接口地址：http://geek.itheima.net/v1_0/channels
- 请求方式：get
- 请求参数：无

```
1 // 异步action
2 const getList = async ()=>{
3   const res = await axios.request<接口数据类型>({})
4 }
```

需求：在Pinia中获取频道列表数据并把数据渲染App组件的模板中

`object {2}`

▼ data	<code>object {1}</code>
▼ channels	<code>array[object] {2}</code>
id	<code>integer</code>
name	<code>string</code>
message	<code>string</code>

6. storeToRefs工具函数

使用storeToRefs函数可以辅助保持数据（state + getter）的响应式解构



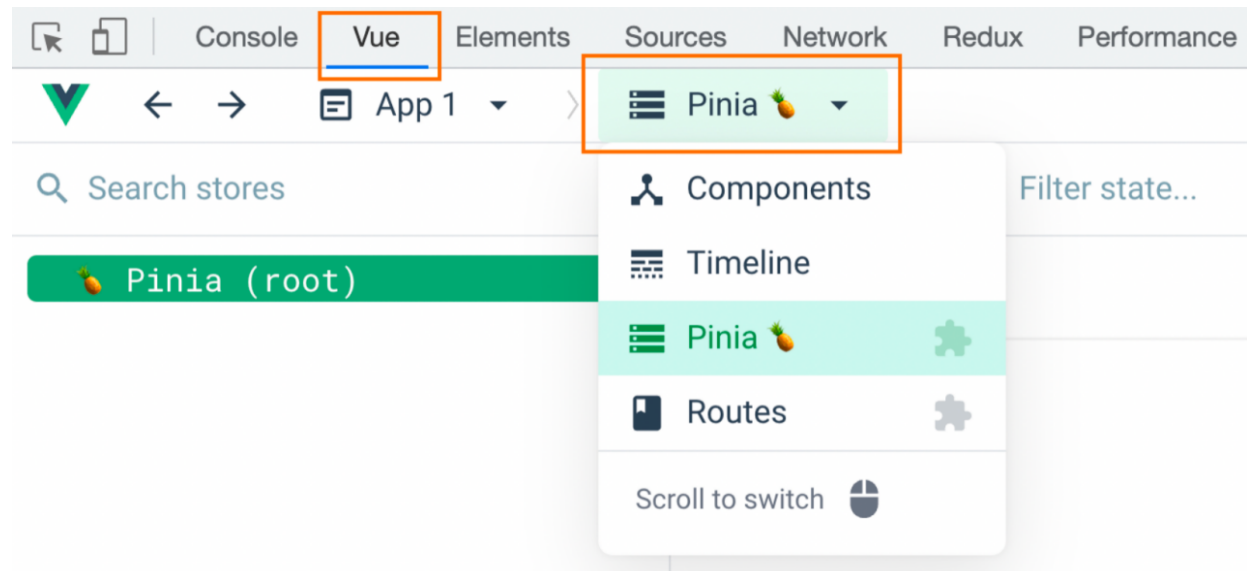
```
1 // 响应式丢失 视图不再更新
2 const { count, doubleCount } = counterStore
```



```
1 // 保持数据响应式
2 const { count, doubleCount } = storeToRefs(counterStore)
```

7. Pinia的调试

Vue官方的 dev-tools 调试工具 对 Pinia直接支持，可以直接进行调试



8. Pinia持久化插件

官方文档: <https://prazdevs.github.io/pinia-plugin-persistedstate/zh/>

1. 安装插件 pinia-plugin-persistedstate

```
npm i pinia-plugin-persistedstate
```

2. 使用 main.js

```
import persist from 'pinia-plugin-persistedstate'
...
app.use(createPinia().use(persist))
```

3. 配置 store/counter.js

```
import { defineStore } from 'pinia'
import { computed, ref } from 'vue'

export const useCounterStore = defineStore('counter', () => {
  ...
  return {
    count,
    doubleCount,
    increment
  }
}, {
  persist: true
})
```

4. 其他配置，看官网文档即可