

Pinia快速入门

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

手动添加Pinia到Vue项目

后面在实际开发Vue医疗项目的时候，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TS + Vue3项目

```
npm create vite@latest vue-pinia-ts -- --template vue-ts
```

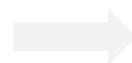
2. 按照官方文档安装 pinia 到项目中

Pinia基础使用 - 计数器案例

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
```

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)
```

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

```
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 | object {1} |
| ▼ channels | array[object] {2} |
| id | integer |
| name | string |
| message | string |

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



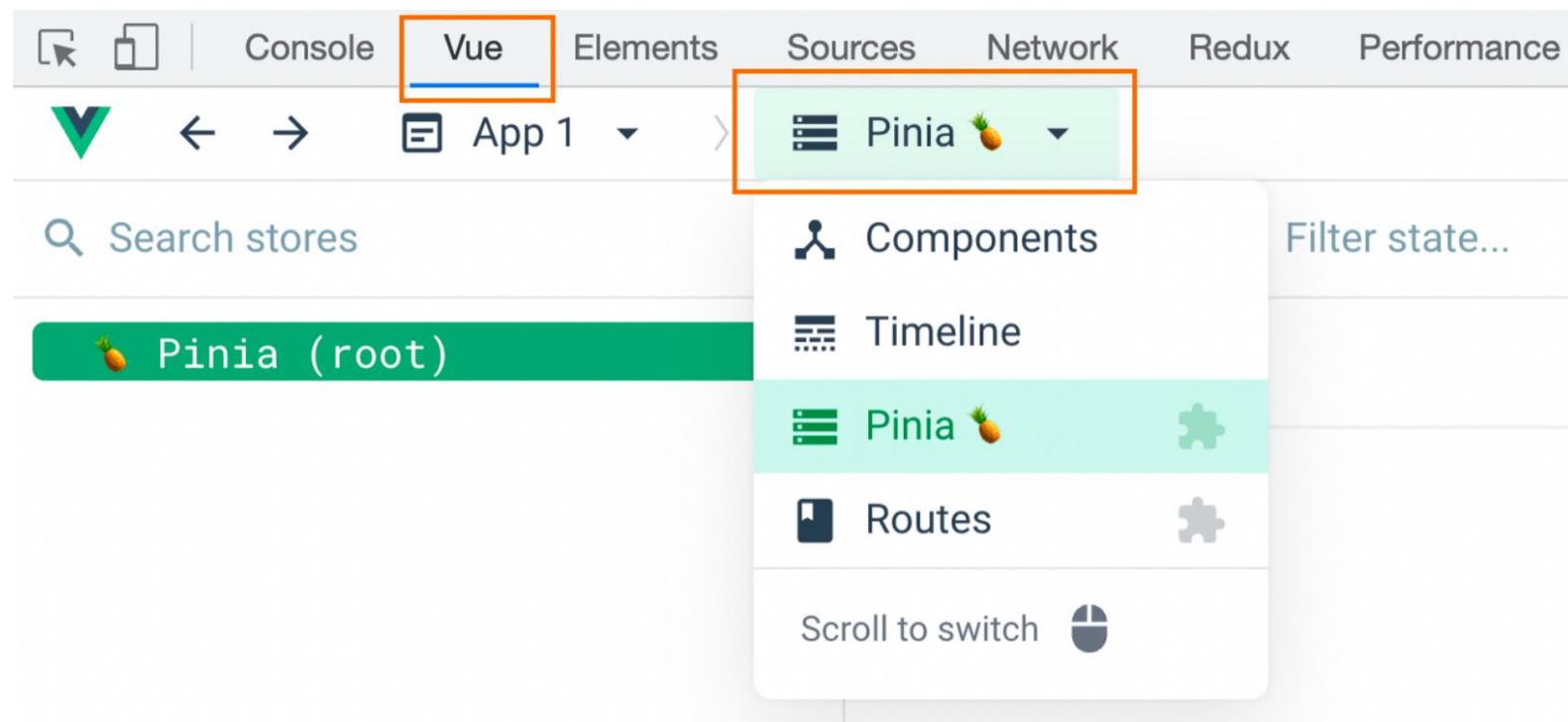
```
1 // 响应式丢失 视图不再更新  
2 const { count, doubleCount } = counterStore
```



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

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





总结

1. Pinia是用来做什么的?

集中状态管理工具，新一代的vuex

2. Pinia中还需要mutation吗?

不需要，action既支持同步也支持异步

3. Pinia如何实现getter?

computed计算属性函数

4. Pinia产生的Store如何解构赋值数据保持响应式?

storeToRefs



传智教育旗下高端IT教育品牌