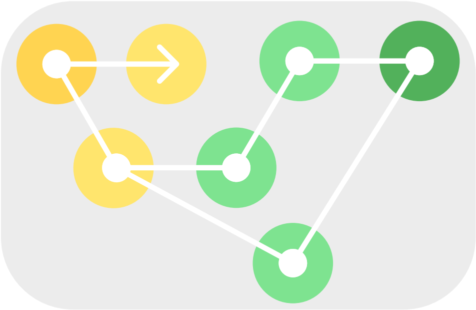
Local Storage with Pinia

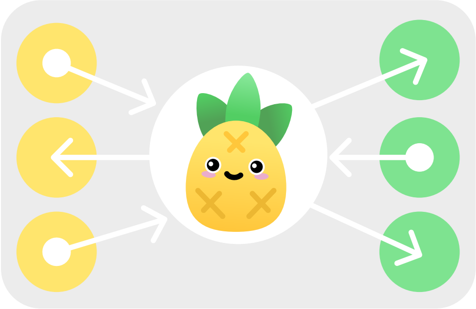
Ethan Mooney

02/08/2023

Instead of using basic state management with props and events, which can get confusing, we can use a local storage state management tool like Pinia to keep track of it all for us.



vs



Pinia stores are simple to use! They are created similar to components, and function kind of like classes in other programming languages. They're also really simple to implement (see code on the right)!

Getting Started

```
npm install pinia
```

```
npm install @vue/composition-api
```

Demo Code and Pinia Docs

https://github.com/ethanimooney/pinia-demo

https://pinia.vuejs.org/

Simple Store Definition

```
import { defineStore } from 'pinia';

export const useCounterStore = defineStore('counter', {
  state: () => {
    return { count: 0 };
  },
  getters: { doubleCount: (state) => state.count * 2 },
  actions: {
    increment() {
      this.count++;
    },
    randomizeCounter() {
      this.count = Math.round(100 * Math.random());
    },
  },
});
```

Simple Store Usage

```
<template>
   <div>
   <h1>Curent Count: {{ counterStore.count }}</h1>
   <h1>Doubled Count: {{ counterStore.doubleCount }}
   </h1>
   <h1>Increment Count: {{ incrementCount() }} </h1>
   </div>
</template>

<script>
   import { useCounterStore } from "@/stores/
CounterStore";

   import { mapStores } from "pinia";

   export default {
   name: "HomeComponent",
   computed: {
      ...mapStores(useCounterStore),
   },
   methods: {
      incrementCount() {
         this.counterStore.incrememt();
         return this.counterStore.count;
      },
   },
   };
</script>
```