GABRIELE ROMANATO

Comunicazione tra componenti in Vue.js con Pinia

In un'applicazione Vue.js, la comunicazione tra componenti può diventare complessa man mano che l'app cresce. Pinia è lo store ufficiale di Vue 3 e fornisce un modo semplice e reattivo per condividere lo stato tra componenti. In questo articolo vedremo come utilizzare Pinia per facilitare la comunicazione tra componenti.

Installazione di Pinia

Per iniziare, è necessario installare Pinia e configurarlo nell'app Vue.

```
npm install pinia
```

```
// main.js
import { createApp } from 'vue'
import { createPinia } from 'pinia'
import App from './App.vue'

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

Creazione di uno store

Creiamo uno store per condividere lo stato tra componenti. Supponiamo di voler condividere un contatore.

```
// stores/counter.js
import { defineStore } from 'pinia'

export const useCounterStore = defineStore('counter', {
  state: () => ({
    count: 0
  }),
  actions: {
    increment() {
      this.count++
    }
  }
})
```

Utilizzo dello store in un componente

Ora possiamo utilizzare lo store all'interno di un componente e modificarne lo stato.

```
<template>
  <div>
    <p>Conteggio: {{ counter.count }}</p>
    <button @click="counter.increment">Incrementa</button>
```

```
  </div>
</template>

<script setup>
import { useCounterStore } from '@/stores/counter'

const counter = useCounterStore()
</script>
```

Comunicazione tra componenti

Con Pinia, più componenti possono leggere e scrivere nello stesso store senza dover passare props o emettere eventi.

Esempio di secondo componente che legge lo stato:

```
<template>
  <div>
    <h3>Valore attuale: {{ counter.count }}</h3>
  </div>
</template>

<script setup>
import { useCounterStore } from '@/stores/counter'

const counter = useCounterStore()
</script>
```

Entrambi i componenti condividono lo stesso stato in modo reattivo. Quando un componente modifica il valore, l'altro lo rileva automaticamente.

Conclusione

Pinia semplifica notevolmente la comunicazione tra componenti Vue, evitando la necessità di soluzioni complesse come event bus o prop drilling. È reattivo, modulare e facile da usare, rendendolo una scelta eccellente per la gestione dello stato in applicazioni Vue 3.

Applicazioni Correlate

- ### JavaScript Password Mask

  Un esempio in JavaScript di mascheramento di una password con l'aggiunta della funzionalità di copia negli appunti.
  DockerDocker ComposeJavaScript

- **Python Placeholder Image**

  Applicazione sviluppata in Python con Flask per la creazione di immagini segnaposto.
  DockerDocker ComposePythonFlaskJavaScript

-

  Go Placeholder Image

  Applicazione in Go per la creazione di immagini segnaposto.
  DockerDocker ComposeGoJavaScript