

GABRIELE ROMANATO

React: l'hook useMemo()

L'hook useMemo() è uno strumento fondamentale per ottimizzare le prestazioni delle applicazioni React. Introdotta in React 16.8, questa funzione permette di evitare calcoli onerosi e ridondanti all'interno di un componente.

Quando un componente React viene renderizzato, tutte le sue variabili e funzioni vengono ricalcolate. In alcune situazioni, però, ciò può comportare uno spreco di risorse, specialmente quando i calcoli richiedono molto tempo o coinvolgono grandi set di dati. È qui che entra in gioco useMemo().

Utilizzando useMemo(), è possibile memorizzare il risultato di un calcolo oneroso e riutilizzarlo solo quando le dipendenze specificate cambiano. In altre parole, l'hook determina se i valori di riferimento sono cambiati rispetto all'ultima volta che il componente è stato renderizzato e, se non ci sono modifiche, restituisce il valore memorizzato precedentemente. Questo evita calcoli inutili e può migliorare significativamente le prestazioni dell'applicazione.

Per utilizzare useMemo(), è necessario specificare una funzione di callback e un array di dipendenze. La funzione di callback contiene il calcolo che si desidera memorizzare, mentre l'array di dipendenze indica quali valori dovrebbero essere monitorati per determinare se il calcolo deve essere eseguito nuovamente.

```
import React, { useState, useMemo } from 'react';

const App = () => {
  const [count, setCount] = useState(0);

  const expensiveCalculation = useMemo(() => {
    console.log('Running...');
    let result = 0;
    for (let i = 0; i < count * 10000000; i++) {
      result += i;
    }
    return result;
  }, [count]);

  return (
    <div>
      <p>Count: {count}</p>
      <p>Result: {expensiveCalculation}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
};

export default App;
```

In questo esempio, abbiamo un componente App che contiene un contatore count e un calcolo oneroso rappresentato dalla variabile expensiveCalculation. L'hook useMemo() viene utilizzato per memorizzare il risultato del calcolo oneroso e viene aggiornato solo quando il valore di count cambia.

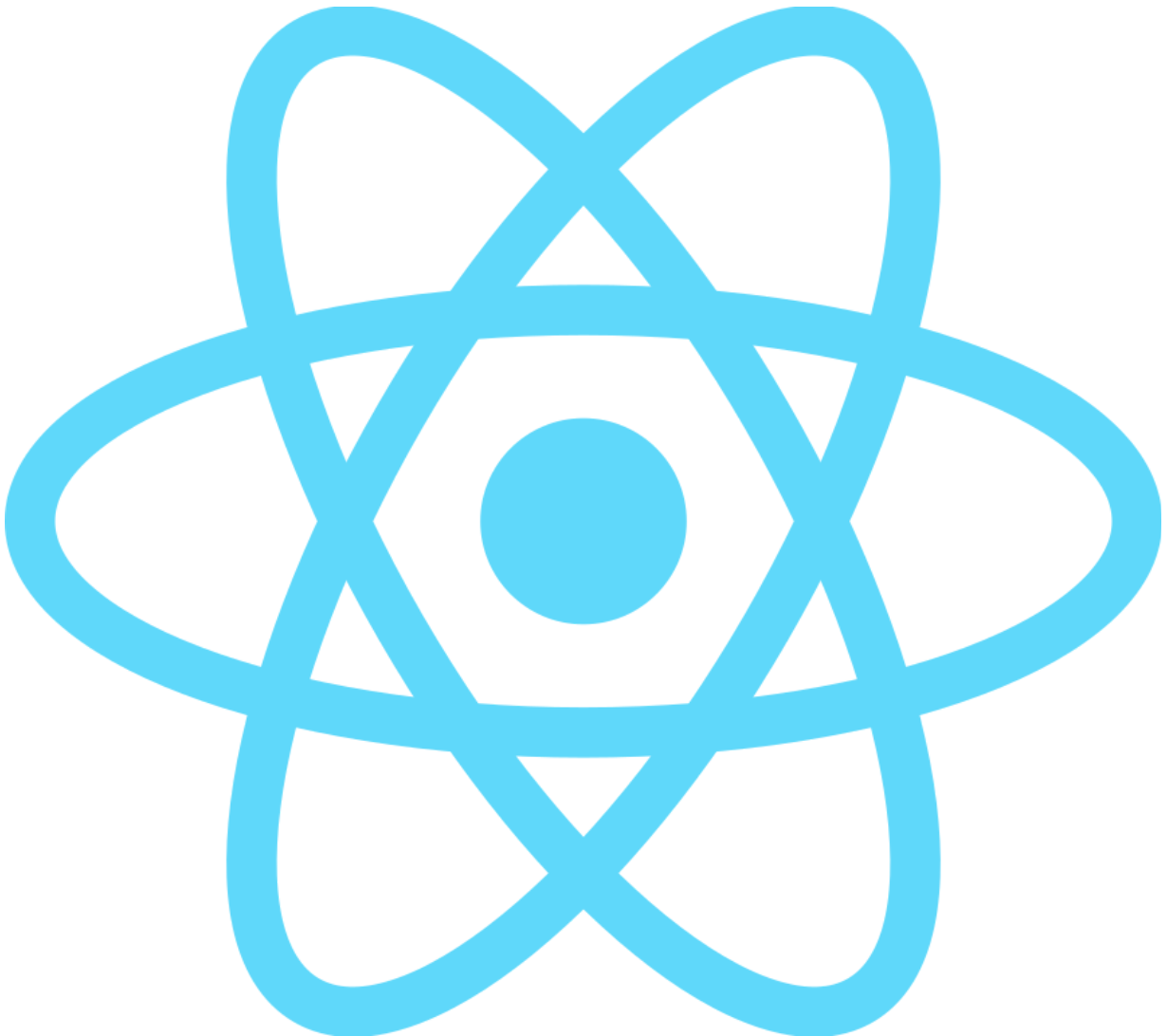
Ogni volta che il pulsante "Increment" viene premuto, il valore di count aumenta di 1. Senza l'utilizzo di `useMemo()`, il calcolo oneroso verrebbe eseguito ad ogni renderizzazione del componente, anche se il valore di count non cambia. Tuttavia, grazie all'utilizzo di `useMemo()`, il calcolo oneroso viene eseguito solo quando count cambia effettivamente, altrimenti viene restituito il valore memorizzato precedentemente.

Utilizzando la `console.log` all'interno della funzione di callback di `useMemo()`, possiamo vedere che il calcolo oneroso viene effettuato solo quando count cambia, dimostrando così il miglioramento delle prestazioni ottenuto grazie all'hook `useMemo()`.

In questo modo, l'uso di `useMemo()` consente di evitare calcoli ridondanti e di migliorare le prestazioni delle applicazioni React, specialmente quando sono presenti operazioni costose o complesse da eseguire.

In conclusione, l'hook `useMemo()` di React è uno strumento potente per ottimizzare le prestazioni delle applicazioni React. Utilizzando `useMemo()` in modo appropriato, è possibile evitare calcoli ridondanti e migliorare l'efficienza complessiva dell'applicazione. Tuttavia, è importante utilizzare questo hook con cautela e solo quando si sono identificati effettivamente dei calcoli onerosi che possono essere evitati.

Applicazioni Correlate



-

Book Store

Un'applicazione in React con Node.js e Fastify come backend.
DockerDocker ComposeNode.jsJavaScriptFastifyReact