

GABRIELE ROMANATO

React: uso del metodo `filter()` degli array JavaScript

Il metodo `filter()` in JavaScript è uno strumento potente quando si lavora con gli array, ed è particolarmente utile in combinazione con React per manipolare e filtrare i dati visualizzati nell'interfaccia utente.

In React, si può utilizzare il metodo `filter()` per creare una copia filtrata di un array di elementi. Ad esempio, se si ha un array di oggetti che rappresentano utenti e si desidera visualizzare solo gli utenti attivi, si può utilizzare `filter()` per filtrare l'array in base a un criterio specifico.

Ecco un esempio di utilizzo di `filter()` in React:

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

function UserList() {
  const [users, setUsers] = useState([
    { id: 1, name: 'Mario', active: true },
    { id: 2, name: 'Luigi', active: false },
    { id: 3, name: 'Peach', active: true },
    { id: 4, name: 'Toad', active: false }
  ]);

  const activeUsers = users.filter(user => user.active);

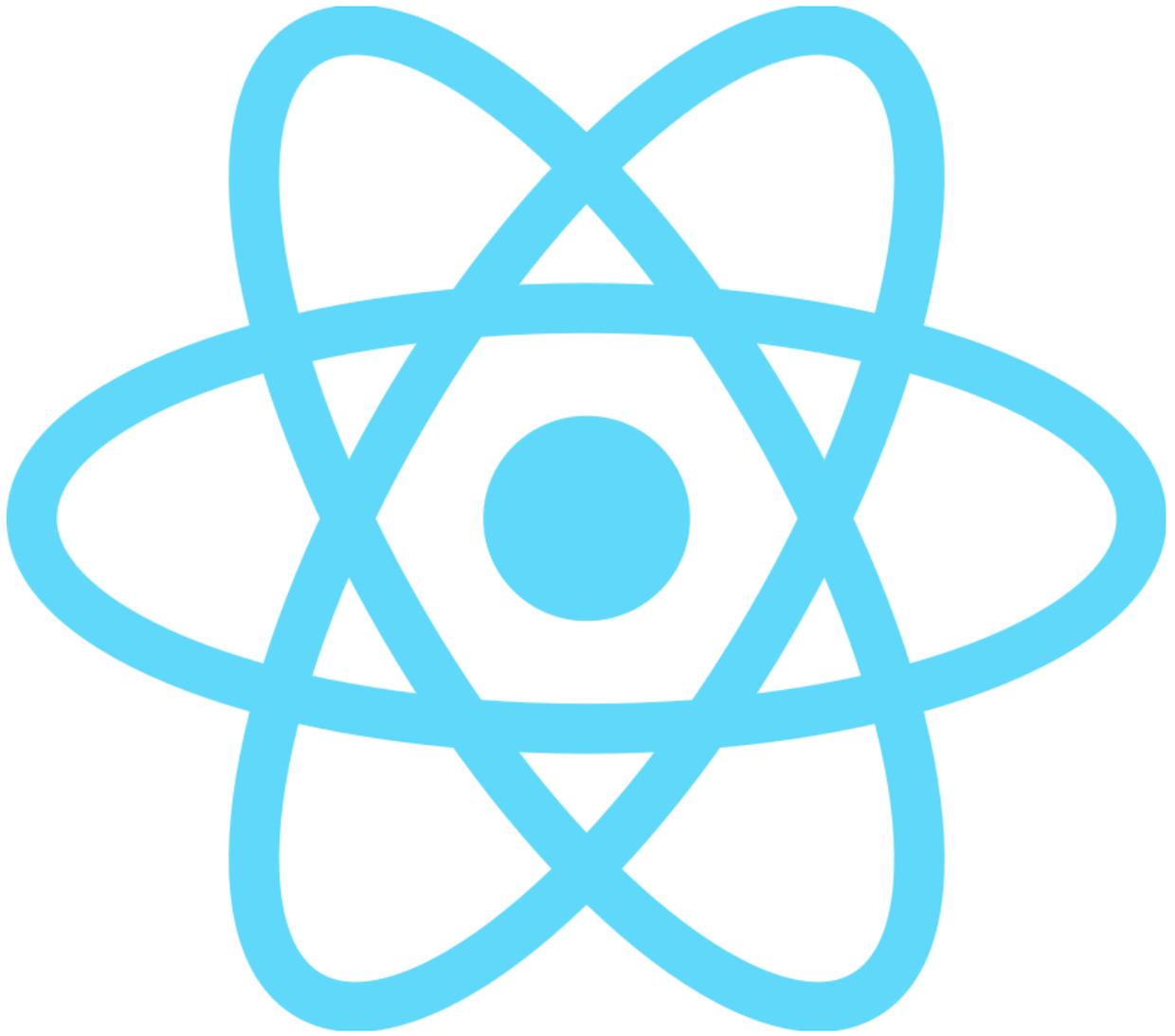
  return (
    <div>
      <h1>Lista Utenti Attivi</h1>
      <ul>
        {activeUsers.map(user => (
          <li key={user.id}>{user.name}</li>
        ))}
      </ul>
    </div>
  );
}

export default UserList;
```

Nell'esempio sopra, l'array `users` contiene una lista di utenti con proprietà come `id`, `name` e `active`. Il metodo `filter()` viene utilizzato per ottenere una nuova copia dell'array che include solo gli utenti attivi. Questo array filtrato viene quindi mappato all'interno di un elemento `` per visualizzare i nomi degli utenti attivi nella lista.

Il metodo `filter()` offre una flessibilità notevole, poiché consente di definire criteri di filtro personalizzati per adattarsi alle esigenze specifiche dell'applicazione. È possibile filtrare in base a diverse proprietà degli oggetti o applicare logiche più complesse. L'utilizzo di `filter()` in combinazione con React consente di gestire facilmente i dati e di renderizzare l'interfaccia utente in base a criteri specifici.

Applicazioni Correlate



•

Book Store

Un'applicazione in React con Node.js e Fastify come backend.
Docker Docker Compose Node.js JavaScript Fastify React