

# GABRIELE ROMANATO

## React: usare axios

Axios è una libreria JavaScript ampiamente utilizzata per effettuare richieste HTTP in modo semplice ed efficiente. Quando si lavora con React, Axios è una scelta popolare per gestire le chiamate API. Ecco come puoi utilizzare Axios in React per ottenere e inviare dati dal tuo server.

Innanzitutto, dovrai installare Axios nel tuo progetto React. Puoi farlo eseguendo il seguente comando nel tuo terminale:

```
npm install axios --save
```

Puoi usare Axios nei tuoi componenti nel modo seguente:

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

const MyComponent = () => {
  const [data, setData] = useState(null);

  useEffect(() => {
    axios.get('url_del_tuo_server')
      .then(response => {
        setData(response.data);
      })
      .catch(error => {
        console.error(error);
      });
  }, []);

  return (
    <div>
      {data ? (
        <ul>
          {data.map(item => (
            <li key={item.id}>{item.name}</li>
          ))}
        </ul>
      ) : (
        <p>Caricamento dei dati...</p>
      )}
    </div>
  );
};

export default MyComponent;
```

In questo esempio, abbiamo un componente chiamato MyComponent che utilizza lo stato data per memorizzare i dati ottenuti dalla richiesta GET. Utilizziamo l'hook useEffect per effettuare la richiesta

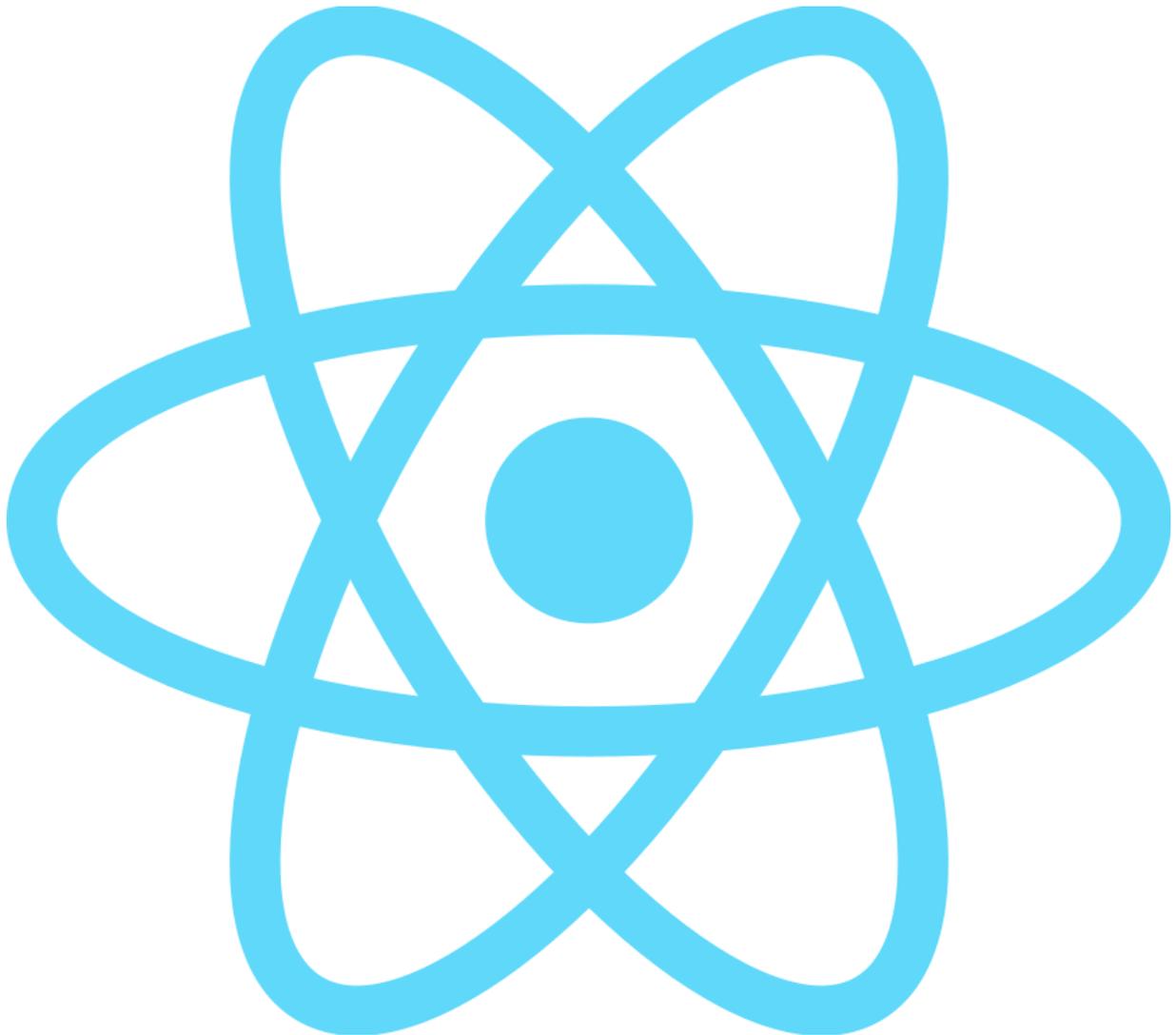
quando il componente viene montato. Una volta ricevuti i dati, vengono visualizzati all'interno di una lista `<ul>`.

Nel caso in cui i dati non siano ancora disponibili, viene mostrato il messaggio "Caricamento dei dati...". Questo è utile per gestire lo stato di caricamento durante il recupero dei dati dal server.

Ricorda di sostituire `'url_de1_tuo_server'` con l'URL effettivo del tuo server.

Questo esempio ti offre un punto di partenza per utilizzare Axios all'interno di un componente di React. Puoi estendere questa logica per gestire altre richieste o implementare ulteriori funzionalità nel tuo progetto React.

## Applicazioni Correlate



- 

### **Book Store**

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