

## GABRIELE ROMANATO

### Creazione di un componente loader in React

Un "loader" è un componente essenziale in molte applicazioni web, specialmente quando si tratta di caricare dati in modo asincrono o eseguire operazioni complesse che richiedono del tempo. Un loader fornisce un feedback visivo agli utenti, indicando che l'applicazione sta lavorando nel background per soddisfare una richiesta. In questo articolo, imparerai come creare un componente loader in React, un popolare framework JavaScript per la creazione di interfacce utente interattive.

### Creazione del componente Loader

Ora creeremo un nuovo componente React chiamato "Loader". Puoi farlo creando un nuovo file chiamato `Loader.js` nella cartella `src` del tuo progetto e quindi definendo il componente all'interno del file.

```
// src/Loader.js

import React from 'react';

const Loader = () => {
  return (
    <div className="loader">
      <div className="loader-spinner"></div>
    </div>
  );
};

export default Loader;
```

In questo componente, abbiamo creato un `div` con la classe "loader" e al suo interno un altro `div` con la classe "loader-spinner". Questi elementi costituiranno il nostro loader.

### Stilizzazione del Loader

Ora che abbiamo definito il nostro componente, dobbiamo aggiungere alcune regole CSS per renderlo visivamente attraente. Aggiungi le seguenti regole CSS nel tuo file `App.css` o in un nuovo file CSS dedicato:

```
/* src/Loader.css */

.loader {
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100%;
  width: 100%;
  min-height: 100vh;
  position: fixed;
  top: 0;
  left: 0;
  z-index: 999999;
  background-color: #fff;
```

```

}

.loader-spinner {
  border: 4px solid rgba(255, 255, 255, 0.3);
  border-top: 4px solid #3498db;
  border-radius: 50%;
  width: 40px;
  height: 40px;
  animation: spin 1s linear infinite;
}

@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

```

Questo CSS definisce un loader al centro della pagina con un'icona a forma di cerchio che ruota durante il caricamento.

## Utilizzo del Loader

Ora che hai creato il componente Loader e lo hai stilizzato, puoi utilizzarlo nelle tue pagine o componenti React per fornire un feedback visivo durante le operazioni di caricamento. Ecco come puoi utilizzare il Loader all'interno di un componente:

```

// src/App.js

import React, { useState, useEffect } from 'react';
import './Loader.css';
import Loader from './Loader';

function App() {
  const [loading, setLoading] = useState(true);

  useEffect(() => {
    // Simuliamo un ritardo di 3 secondi per il caricamento
    setTimeout(() => {
      setLoading(false);
    }, 3000);
  }, []);

  return (
    <div className="App">
      {loading ? <Loader /> : <h1>Contenuto caricato con successo!</h1>}
    </div>
  );
}

export default App;

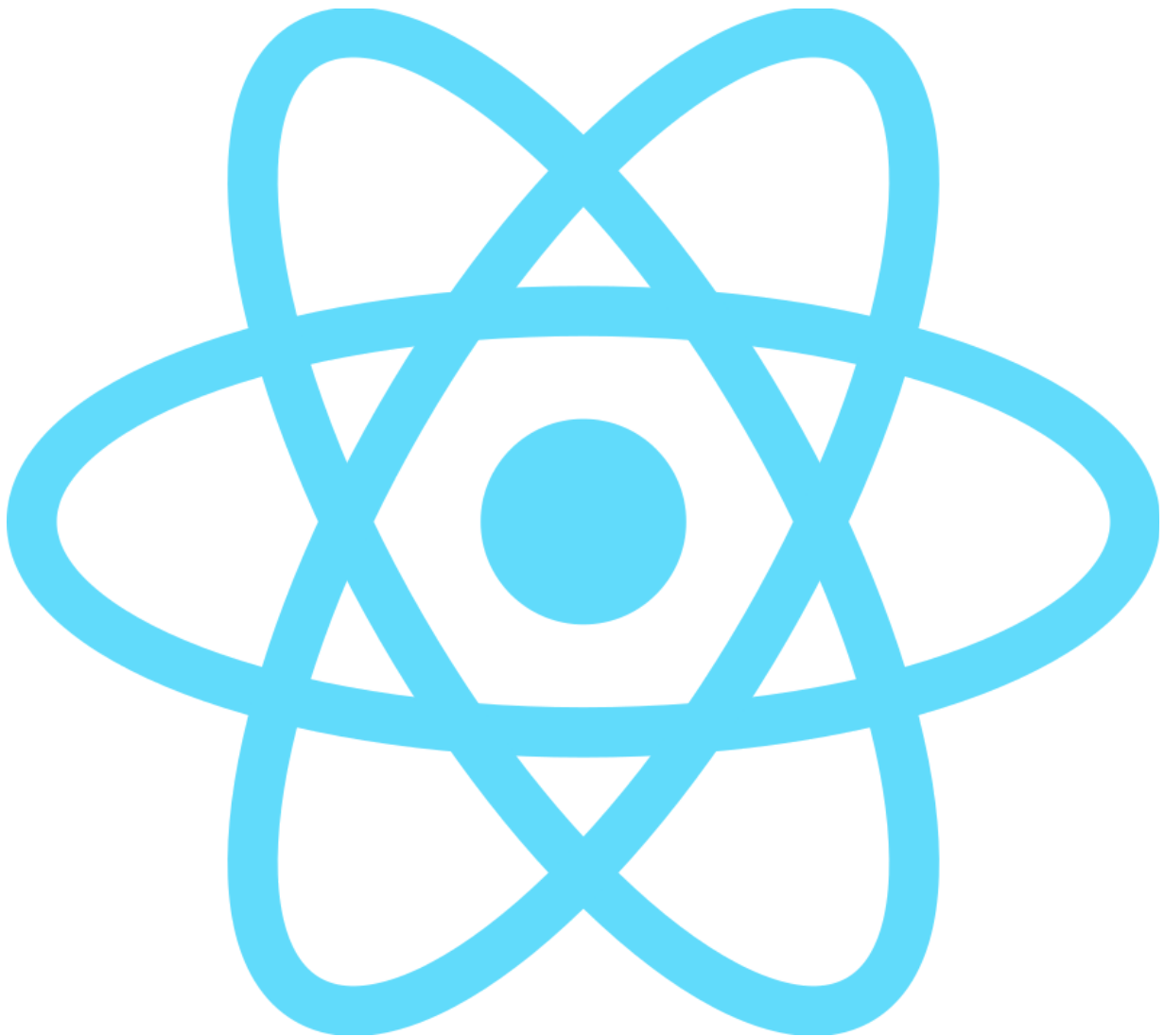
```

In questo esempio, il componente Loader verrà visualizzato finché la variabile di stato `loading` è `true`. Dopo un ritardo di 3 secondi (che simula un'operazione di caricamento), la variabile `loading` viene impostata su `false`, e il componente "Contenuto caricato con successo!" verrà visualizzato al suo posto.

## Conclusioni

Hai appena imparato a creare un componente loader in React per fornire feedback visuale durante le operazioni di caricamento. Questo è solo un esempio di come puoi personalizzare e utilizzare un componente loader. Puoi adattarlo alle esigenze specifiche della tua applicazione, migliorando l'esperienza utente durante le operazioni di caricamento asincrone.

## Applicazioni Correlate



.

### Book Store

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