Introducción a React.Js

Módulo 05

React Transition Group

La librería de react transition group es un conjunto de componentes para administrar los estados de los componentes (incluido el montaje y desmontaje) a lo largo del tiempo, diseñados específicamente con la animación en mente.

El componente Transición le permite describir una transición de un estado de componente a otro a lo largo del tiempo con una API declarativa simple. Más comúnmente se usa para animar el montaje y desmontaje de un componente, pero también se puede usar para describir estados de transición in situ.

Instalación

Como react-transition-group es bastante pequeño, la sobrecarga de incluir la biblioteca en su aplicación es insignificante. Sin embargo, en situaciones donde puede ser útil beneficiarse de un CDN externo cuando se empaqueta,

npm install react-transition-group --save

Uso

Por defecto, el componente Transition no altera el comportamiento del componente que representa, solo rastrea los estados "enter" y "exit" para los componentes. Depende de usted dar significado y efecto a esos estados. Por ejemplo, podemos agregar estilos a un componente cuando ingresa o sale:

Uso

Como se señaló, el componente Transition no hace nada por sí solo en su
componente secundario. Lo que hace es rastrear los estados de transición a lo
largo del tiempo para que pueda actualizar el componente (como agregar estilos
o clases) cuando cambia de estado.

Hay 4 estados principales en los que puede haber una Transición:

- entering
- entered
- exiting
- exited

```
import Transition from 'react-transition-group/Transition';

const duration = 300;

const defaultStyle = {
  transition: `opacity ${duration}ms ease-in-out`,
  opacity: 0,
}

const transitionStyles = {
  entering: { opacity: 0 },
  entered:  { opacity: 1 },
};

const Fade = ({ in: inProp }) => (
  <Transition in={inProp} timeout={duration}>
    {(state) => (
      <div style={{
        ...defaultStyle,
        ...transitionStyles[state]
      }}>
        I'm a fade Transition!
      </div>
    )}
  </Transition>
);
```

Uso

El estado de transición se alterna a través del in prop. Cuando es verdadero, el
componente comienza la etapa "Entrar". Durante esta etapa, el componente
cambiará de su estado de transición actual a "entrar" durante la transición y luego
a la etapa "ingresada" una vez que esté completo. Tomemos el siguiente ejemplo:

```
state= { in: false };

toggleEnterState = () => {
  this.setState({ in: true });
}

render() {
  return (
    <div>
      <Transition in={this.state.in} timeout={500} />
      <button onClick={this.toggleEnterState}>Click to
Enter</button>
    </div>
  );
}
```

Uso

Cuando se hace clic en el botón, el componente cambiará al estado de "entering" y permanecerá allí durante 500 ms (el valor del tiempo de espera) antes de que finalmente cambie a "entered".

Cuando es falso ocurre lo mismo, excepto que el estado pasa de 'exiting' a 'exited'.

¡Muchas gracias!

¡Sigamos trabajando!