The provided code is a React functional component named Spinner. This component is likely used to display a loading spinner or indicator on a webpage. Below is an explanation of the key elements in the code:

Component Structure:

Import Statements:

The component imports the React library for building UI components and a CSS file named 'Spinner.css' for styling.

Spinner Component:

Spinner is a functional component that returns JSX code representing a loading spinner.

HTML Structure:

The spinner is wrapped in a <div> element with the following properties:

className='fixed inset-0 z-50 bg-black/80 grid place-items-center': This sets the position of the spinner to fixed, covering the entire viewport (inset-0), giving it a high z-index (z-50), and placing it at the center of the screen (grid place-items-center). The background is semi-transparent black (bg-black/80).

Inside this <div>, there is another <div> with the class name 'loader1'. This is likely the visual representation of the loading spinner.

Usage in Documentation:

To explain this component in documentation, you can highlight its purpose as a loading indicator, its styling with a transparent overlay, and the use of a custom loader style defined in the 'Spinner.css' file.

The provided code is a React functional component named SpinnerBtn. This component appears to be a customizable loading spinner, likely intended for use within buttons. Here's an explanation of its key features:

Component Structure:

Import Statements:

The component imports the React library for building UI components and a CSS file named 'Spinner.css' for styling.

SpinnerBtn Component:

SpinnerBtn is a functional component that takes two props, parentClass and childClass, for customization.

HTML Structure:

The spinner is wrapped in a <div> element with the class name 'loader2' and the additional class specified by the parentClass prop.

Inside this <div>, there are five <div> elements representing rectangles with class names 'rect1' through 'rect5'. These rectangles have an additional class specified by the childClass prop.

Customization:

The parentClass prop is intended for customization of the overall spinner container.

The childClass prop is meant for customization of each rectangle, allowing for a more personalized appearance.

Usage in Documentation:

In the documentation, you can emphasize the flexibility of the SpinnerBtn component, allowing users to tailor both the overall appearance and individual components of the spinner.

The provided code is a React functional component named TableLoadingSkeleton. This component is designed to create a loading skeleton for a table, commonly used to display temporary content placeholders while data is being fetched. Below is an explanation of the key elements in the code:

Component Structure:

Import Statements:

The component imports the React library for building UI components and a CSS file named 'Spinner.css' for styling.

TableLoadingSkeleton Component:

TableLoadingSkeleton is a functional component that takes two props: tr\_count (defaulted to 8) and td\_count.

It returns JSX code representing a loading skeleton for a table.

Mapping Rows and Cells:

The component uses the map function to generate rows (<tr>) based on the value of tr\_count. Each row has a unique key and a border-bottom (border-b) style.

Within each row, it maps cells (<td>) based on the value of td\_count. Each cell also has a unique key.

Conditional Styling:

The className of each cell is dynamically determined using a template string. It adds different classes ('px-5' or 'pr-5') based on whether the cell is the first one in the row (td === 0), controlling the padding to create a visual distinction.

Loading Placeholder:

Inside each cell, there's a <div> element with the class name 'table\_loading'. This is likely the visual representation of a loading spinner or placeholder for data that is being fetched.

Usage in Documentation:

To explain this component in documentation, you can highlight its purpose as a loading skeleton for tables, its customization through props, and the dynamic styling applied to create a visually appealing placeholder.