

RecipeFusion

Ans: Uncontrolled components in React are form elements that maintain their own internal state. The state is managed by the browser, not by React, and accessed through the DOM. Controlled components, on the other hand, are form elements whose state is controlled by React. The values of these elements are stored in state and updated through `setState()` or other state management techniques.

1. Differences between Uncontrolled and Controlled Components in React?

Ans: `PropTypes` is a built-in feature in React that enables you to check the type of props passed to a component. To use `PropTypes`, you need to import it from the 'prop-types' package and define a `propTypes` object in your component. This object will define the expected types of your props, which React will validate at runtime.

2. How to Validate React Props using `PropTypes`?

Ans: `Node.js` is a server-side JavaScript runtime that allows developers to run JavaScript on the server. It provides a platform for building scalable, high-performance web applications. `Express.js`, on the other hand, is a lightweight web application framework that runs on top of `Node.js`. It provides a set of tools and features that make it easy to build web applications and APIs.

3. Difference between `Node.js` and `Express.js`?

Ans: A custom hook in React is a JavaScript function that allows you to reuse logic across different components. It is a way to extract stateful logic from a component and reuse it in other components. Custom hooks are created using the 'use' prefix, such as 'useFetch' or 'useLocalStorage'. You might create a custom hook when you find yourself repeating the same code in different components, or when you want to share functionality between unrelated components.

4. What is a Custom Hook, and Why will you create a Custom Hook?

Custom hooks can also make your code more readable and maintainable by abstracting away complex logic and reducing the amount of code in your components.

RecipeFusion