Blogs

1. Tell us the differences between uncontrolled and controlled components.

Author: Jacks Read Time: 5min Likes: 2.3k

Uncontrolled and controlled components are two ways of handling form input elements in web development. Uncontrolled components are simpler and require less code to set up, but they have less control over the form data and can be harder to validate and manipulate before submitting. Controlled components, on the other hand, are more powerful and flexible but require more code to set up and manage. They provide greater control over the form data and make it easier to validate, manipulate, and submit the data as needed. In summary, the main difference between uncontrolled and controlled components is the level of control over the form data. Uncontrolled components are easier to work with but have less control, while controlled components provide more control but are more complex to manage.

2. How to validate React props using PropTypes

Author: Norche   Read Time: 7min   Likes: 5.3k

PropTypes is a library in React that provides a set of validators to check the type and shape of props being passed to a component. To use PropTypes, it must be installed separately from the core React library. Once installed, PropTypes can be imported into the component, and the PropTypes for each prop that is being passed to the component can be defined using the PropTypes object. By defining PropTypes for the props, we can ensure that the props being passed to the component are of the correct type and shape, which helps to avoid runtime errors and improve the reliability of the code. Defining PropTypes is a simple and effective way to validate props in React.

3. Tell us the difference between nodejs and express js.

Author: Jhon

Read Time: 9min

Likes: 8.3k

Node.js and Express.js are two popular technologies used in web development, particularly for building server-side applications. While they are often used together, they are not the same thing. Node.js is a server-side JavaScript runtime environment that allows developers to run JavaScript code on the server. It provides a non-blocking I/O model and an event-driven architecture, which makes it well-suited for building scalable and high-performance server-side applications. With Node.js, developers can write server-side code in JavaScript, which allows for greater consistency and easier code sharing between the client and server. Express.js, on the other hand, is a popular web application framework built on top of Node.js. It provides a set of features and tools for building web applications and APIs, such as routing, middleware, and HTTP request handling. Express.js is designed to be simple, flexible, and lightweight, which makes it easy to get started

4. What is a custom hook, and why will you create a custom hook?

Author: Cathleen

Read Time: 10min

Likes: 1.3k

In React, a custom hook is a JavaScript function that starts with the use keyword and allows you to reuse stateful logic across different components. Custom hooks are a way to abstract away complex logic and make it easy to reuse code in multiple places. Custom hooks can be created to encapsulate logic related to things like data fetching, form validation, or even simple animations. By creating a custom hook, you can separate concerns and keep your code modular and easy to maintain. There are several reasons why you might want to create a custom hook in your React application. For example, you might find yourself writing the same code multiple times across different components, or you might have complex logic that is hard to manage and debug when it is scattered throughout your codebase. By creating a custom hook, you can abstract away this logic and reuse it wherever it is needed, which can save you time and make your code more readable and maintainable. Additionally, custom hooks can be shared across projects or with the wider React community, which can help to improve the quality and usability of your code.