Building a Task Manager App with React

What does the npx create-react-app command do?

The npx create-react-app command sets up a new React application with everything we need: a project structure, build tools, and preinstalled dependencies like React and React-DOM. It also provides a development environment so you can jump right into coding without worrying about configurations.

Why is it used for setting up a React project?

It simplifies the process of starting a React project by eliminating the need to manually configure Webpack, Babel, or other tools. It's beginner-friendly and includes industry best practices.

What is the purpose of the cd command, and why do you need to navigate to the project directory?

The cd (change directory) command is used to move into the project folder. Navigating into the project directory is necessary to run commands and access the tools specific to your React app.

What happens when you run npm start? How do you check if the React app is working?

When you run npm start, it launches a development server and opens app in the browser (usually at http://localhost:3000). We can confirm it's working by seeing the default React logo or template in the browser.

What is the purpose of the TaskInput component? What state should it manage, and how can it communicate with the parent component to add tasks?

The TaskInput component provides a form for users to input new tasks. It should manage the input's value as state and call a function (passed as a prop from the parent) to add tasks when the form is submitted.

How will the TaskItem component receive and handle data (e.g., task text and completion status)? What props will it need?

The TaskItem component will receive props such as the task text, its completion status, and any actions like deleting or toggling completion. These props will allow it to display the task and handle user interactions.

How can the TaskList component map over the array of tasks passed as a prop and render each TaskItem? What props does TaskList need to receive?

The TaskList component will use the map method to loop over the array of tasks and create a TaskItem for each one. It needs to receive the list of tasks and any functions for deleting or toggling tasks as props.

What is the role of the useState hook in managing state in functional components? How would you use it to store tasks in the App component?

The useState hook allows you to manage state in functional components. In the App component, you would use it to store an array of tasks, where each task is an object with properties like id, text, and completed.

How does the addTask function work in updating the state of tasks? Why is the task text passed as an argument to this function?

The addTask function adds a new task to the state by creating a new task object and appending it to the current array of tasks. The task text is passed as an argument to specify the content of the new task.

How can you remove a task from the list in the App component's state? What method is used to filter out the task based on its id?

You can remove a task by using the filter method to create a new array that excludes the task with the specified id. This new array replaces the current state.

How can you toggle the completion status of a task in the App component? How do you update the task state when the checkbox is checked or unchecked?

To toggle a task's completion, map over the tasks and update the completed property for the task with the specified id. Replace the state with the updated array.

What should you see in the browser after running the npm start command? What functionality should the app have at this point?

You should see the task manager app with an input field, a task list, and options to add, delete, and toggle tasks.

How do you ensure the tasks are being added to the state and rendered properly in the list?

Check the React Developer Tools to verify the state updates correctly and ensure the tasks appear in the browser when added.

How does the delete button work, and how does it interact with the state to remove a task?

The delete button calls a function that filters out the task from the state based on its id. The state is then updated, and the UI re-renders to reflect the change.

How does the completion toggle affect the task's appearance? How does the state reflect the task's completion status?

The toggle changes the completed property of the task in state and applies a visual cue (e.g., line-through) using conditional styling. The state update ensures the UI reflects the correct status.

Mirnes Nuhanovic