Sofascore Frontend Academy Lecture 08,
May, 2023

More React hooks. Performance

Alen Murtić

Sofascore

TABLE OF CONTENTS

01 Official hooks

02 Custom hooks

03 React Performance

04 App themes

01

Official hooks

Official hooks

useRouter

- useRouter is not a React, but a Next hook
- Official docs
- Can change current page using router.push method (user doesn’t have to click on a link)
    - Redirects from protected and empty pages
- Useful for obtaining current path and route name

Official hooks

useCallback

- Caching a method between re-renders
- Official docs
- If we have a custom onClick method inside our component, each time during a re-render, the method gets a new memory location (reference)
    - useCallback caches that method given dependency array so reference doesn’t change
    - React compares object equality with Object.is method, which compares memory references for objects, not property values
- Sometimes is an overkill, more on that later
- Sometimes makes sense in combination with React.memo, which we’ll see in Performance part of today’s lesson 🔜

Official hooks

useMemo

- Caching a **method result** between re-renders given dependency array
- Official docs
- Useful for expensive calculations, not just object memory reference
    - Works with data returned from useSWR (thankfully)
    - Therefore, more useful than useCallback - useCallback is actually useMemo which necessarily returns a function
- React doesn't guarantee that useMemo results will be always cached, sometimes they might be re-calculated

useMemo example - useMemo.md

Sofascore

02

Custom hooks

Custom hooks

- Custom hooks are code extraction - hooks calls extractions, specifically
  - They must still obey **rules of hooks**
  - Of course, not every line has to be a hook call
- Often bundled in unofficial, but widely used libraries
  - Sometimes with small bugs 🙃
- Common examples:
  - usePersistedState - be careful, not every version works with SSR
  - useIsServer - next slide, checks if React is currently rendered on server or client
  - useMediaQuery - hook which checks if given media query passes
  - useTheme - styled-components hook for theme, alternative: useContext(ThemeContext)

Custom hook example - useIsServer.md

Sofascore

03

React Performance

Performance basics

- Browser DOM is slow and inconsistent
- Reuse DOM
- SyntheticEvent - All DOM events in React are wrapped in the SyntheticEvent wrapper
  - Browser universal (works for all browsers) event wrapper
  - Has all properties of native browser event (target, currentTarget, stopPropagation, preventDefault, ...)
- Before React 17: Events were pooled and reused, it caused problems for developers

Virtual DOM

- Virtual representation of a UI in the memory
- Abstracts DOM manipulations, event handling
- Synced with real DOM -> reconciliation process
    - That’s how React works

Performance - Developers

- Minimize expensive DOM operations -> follow reconciliation rules
- **Minimize number of unnecessary rerenders**
- When will rerender happen:
    - Prop change - (prop value or reference)
    - State change
    - Context change
    - this.forceUpdate() - class components
    - When parent rerenders

React.memo

- Skip re-rendering a component when its props are unchanged
- Official docs
- const MemoizedComponent = memo(SomeComponent, arePropsEqual?)
- Without arePropsEqual - compares shallow equality, i.e. objects by reference
  - That’s why we need useCallback and useMemo
  - Custom deep comparison can terribly hinder performance
- Great when used in right situations
  - When to useMemo and useCallback by Kent C. Dodds

Performance example - memoization.md

04

App themes

App themes

- Designers must provide common semantic names for theme variables
- Full support using styled components library

Themes example - themes.md

Sofascore

Thank you for your attention!

Sofascore