

React performance Tips & Tricks





Jakub Miškech



Staff software engineer
Outreach

- GitHub: <https://github.com/yakubko>
- LinkedIn: www.linkedin.com/in/jakub-miskech
- Presentation repo: <https://github.com/Yakubko/frontkon-2023>



Agenda for the day

- 01 Debugging environment
- 02 When to useMemo and useCallback
- 03 Troubleshooting context value
- 04 Power of HOC
- 05 References



Debugging environment

React Developer Tools

General → Highlight updates when components re-render

Profiler → Enable recording why each component re-rendered

Eslint plugin

Eslint-plugin-react-hooks

- rules-of-hooks: error
- exhaustive-deps: warn

Build-in Tools

Performance

- CPU/Network throttling

Recorder

- Comparing performance of same steps

Performance monitor

- Real-time page overview

Memory

- Snapshot comparison

When to useMemo and useCallback

When it makes sense to use it and how to avoid
losing reference



Therefore when?

You have a proof that it's really improving something

You solved parent re-render issues



How to write a good useCallback

 Bad

```
const handleFavorite = React.useCallback(() => {  
  void callSomeApi(!favorite);  
}, [ favorite ]);  
  
return <button onClick={handleFavorite}>Toggle favorite</button>
```

 Better

```
const handleFavorite = React.useCallback((favorite) => {  
  void callSomeApi(!favorite);  
}, [ ]);
```



How to write a good useMemo

✗ Bad

```
const options = React.useMemo<Intl.DateTimeFormatOptions>(  
  () => { weekday: 'long', year: 'numeric', month: 'long', da : 'numeric' },  
  [ ], ( y  
);  
  
const formattedTime = React.useMemo(() => {  
  const date = new Date(createdAt);  
  return date.toLocaleDateString('de-DE', options);  
}, [ options, createdAt ]);
```

✓ Better

```
const formattedTime = React.useMemo(() => {  
  const date = new Date(createdAt);  
  const options = { weekday: 'long', year: 'numeric', month: 'long', da : 'numeric' };  
  return date.toLocaleDateString('de-DE', options);  
}, [ createdAt ]);
```

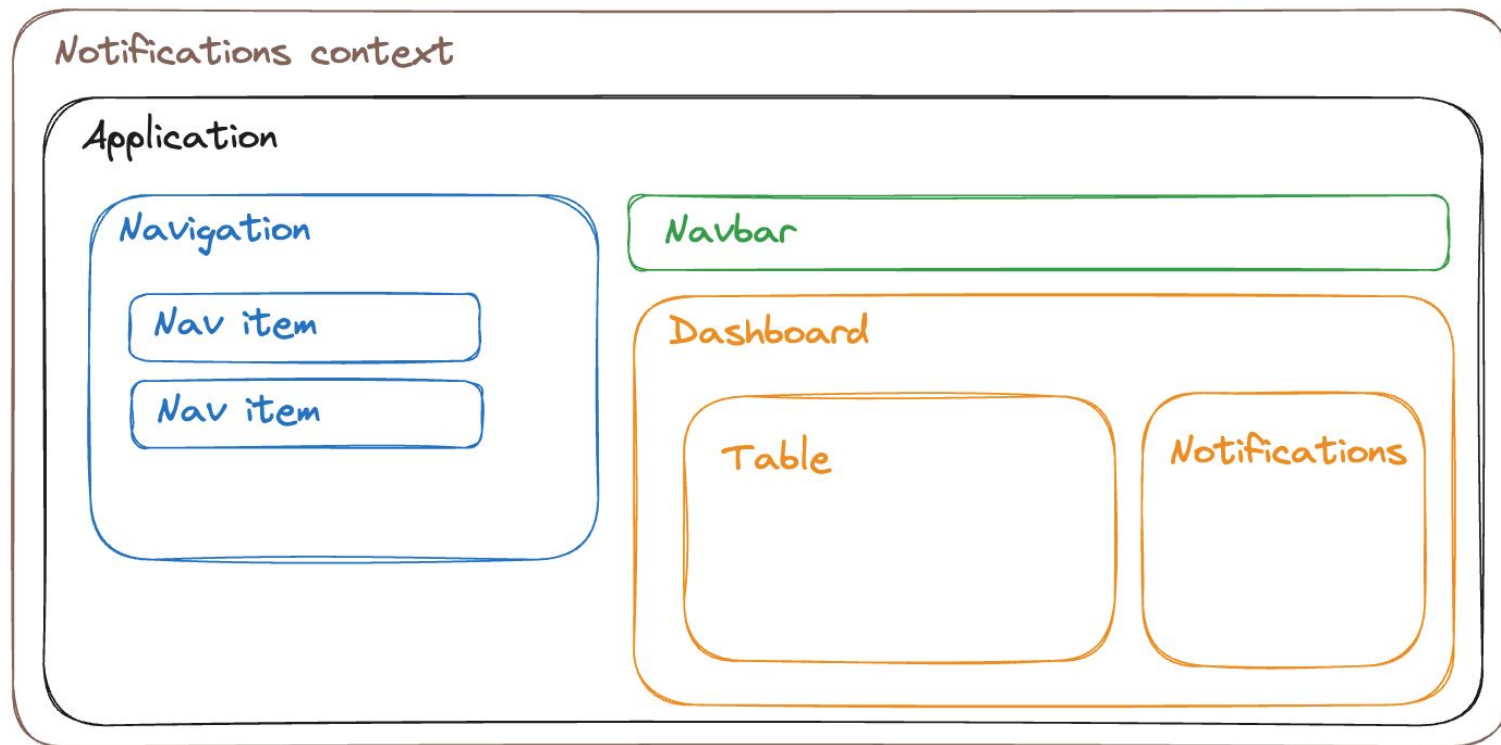
✓ Better

```
const options = { weekday: 'long', year: 'numeric', month: 'long', da : 'numeric' };  
y  
function ListItem({ name, favorite, createdAt }: Item): React.ReactElement {  
  // ...  
  const formattedTime = React.useMemo(() => {  
    const date = new Date(createdAt);  
    return date.toLocaleDateString('de-DE', options);  
  }, [ createdAt ]);
```


Troubleshooting context value

How to avoid unnecessary re-renders when context
value change

App structure



Power of HOC

How HOC can help avoid re-renders and different
ways how to pass component to HOC



How to pass an component

As an Element

```
interface ButtonWithIconElementProps {  
  icon: ReactElement<IconProps>;  
  // ...  
};  
  
<ButtonWithIconElement icon={<AccessAlarmIconGoogle />} />
```

As a Component

```
interface ButtonWithIconElementProps {  
  Icon: ComponentType<IconProps>;  
  // ...  
};  
  
<ButtonWithIconElement icon={AccessAlarmIconGoogle} />
```

As a Function

```
interface ButtonWithIconElementProps {  
  renderIcon: () => ReactElement<IconProps>;  
  // ...  
};  
  
<ButtonWithIconElement renderIcon={() => <AccessAlarmIconGoogle />} />
```



References



About me

- GitHub: <https://github.com/yakubko>
- LinkedIn: www.linkedin.com/in/jakub-miskech
- Presentation repo: <https://github.com/Yakubko/frontkon-2023>



Tools

- React Developer Tools: <https://react.dev/learn/react-developer-tools>
- eslint-plugin-react-hooks: <https://legacy.reactjs.org/docs/hooks-rules.html>



Mentioned links

- When to useMemo and useCallback: <https://kentcdodds.com/blog/usememo-and-usecallback>
- React component the right way: <https://www.developerway.com/posts/react-component-as-prop-the-right-way>

