

MEMOIZATION IN REACT

ANDREI PFEIFFER

[e-spres-oh]

CODE DESIGNER



EVENT ORGANIZER

M

TECHNICAL WRITER

WHAT IS MEMOIZATION

Memoization

From Wikipedia, the free encyclopedia

In computing, **memoization** or **memoisation** is an optimization technique used primarily to speed up computer programs by storing the results of expensive function calls and returning the cached result when the same inputs occur again.

factorial(6)

6 * 5 * 4 * 3 * 2 * 1 = 720

pure functions

// 🔎 ⏳ 🗂️ calculate & store result

factorial(6)

factorial(4)

// 🔎 return from cache

factorial(6)

{

6: 720,

4: 24,

...

}

// no memoization

CALCULATE

// memoization with few stored results



CALCULATE

// memoization with many stored results



CALCULATE

max cache size

WHAT ABOUT REACT

React.memo()

// added in v16.6.0

React.useCallback()

React.useMemo()

// added in v16.8.0, with hooks



function()
components only

```
// Count.jsx  
function Count({ nr }) {  
  return <span>{nr}</span>  
}  
  
export default Count
```

```
// Count.jsx  
function Count({ nr }) {  
  return <span>{nr}</span>  
}  
  
export default React.memo(Count)
```

```
// Count.jsx  
function Count({ nr }) {  
  return <span>{nr}</span>  
}  
  
export default React.memo(Count, [isEqual])
```

```
// Function components  
function Count() {}  
export default React.memo(Count)
```

```
// Class components  
class Count extends React.PureComponent {}
```

cache size

1

`React.memo()`, `React.useMemo()` & `React.useCallback()`

memoize only the last result

```
// Count.jsx  
function Count({ nr }) {  
  const x = factorial(nr)  
  
  return <span>{x}</span>  
}
```

```
// Count.jsx

function Count({ nr }) {
  const x = React.useMemo(() => factorial(nr), [nr])

  return <span>{x}</span>
}
```

```
// Count.jsx

function Count({ nr }) {
  const x = React.useMemo(() => factorial(nr), [nr])

  return <span>{x}</span>
}
```

```
// returns a value, executes the first argument  
React.useMemo(() => sort(items), [items])
```

```
// returns a memoized callback / function  
React.useCallback(() => fetch(id), [id])  
React.useMemo(() => () => fetch(id), [id])
```





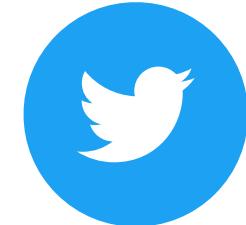
ANTIBIOTICS

TAKE THEM ONLY **IF** AND **WHEN** NEEDED

WORKSHOP

- in-depth hands-on workshop
- understand React's reconciliation process
- how to (not) use **keys**
- using **keys** to force re-renders
- using **useRef()** to skip unneeded re-renders
- using **React.memo()** custom update function
- exercises

THANK YOU



@pfeiffer_andrei



/@andrei.pfeiffer



revojs.ro