

Proxy State



Cory House

Consultant

@housecor | reactjsconsulting.com

Overview



What is proxy state?

Mobx vs. Valtio

Demo: Valtio



What Is Proxy State?



Proxy State

Your state is “wrapped” in an object that monitors when it changes.



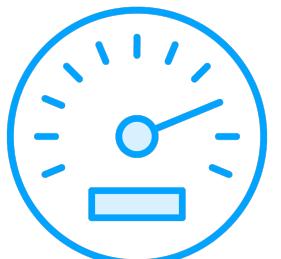
Proxy State



Can directly mutate state



Changes are tracked behind the scenes



Renders are *automatically* optimized



Handling Immutable State

```
const [user, setUser] = useState();  
  
setUser({  
  ...user,  
  city: "New York",  
})
```



Handling Immutable Data in React

`Object.assign`

`{ ...myObj }`

`.map`

Object.assign

Spread operator

**Immutable-friendly
array methods**
(map, filter, reduce...)



With Proxy State, We Can Safely Mutate

```
myState = "newVal"      user.name = "Bob"           .push  
                        .pop
```

Direct assignment

Mutate property

**Mutable array methods
(push, pop, sort...)**



Demo



Valtio



Organizing Actions

```
import { proxy } from 'valtio'

export const state = proxy({ count: 0, name: 'foo', })

export const inc = () => { ++state.count }

export const setName = (name) => { state.name = name }
```



Gotcha:

Valtio triggers a render when the property you access changes.

So, if you don't reference a specific property, it will trigger a render when any property in the object changes.

Suppose we have this state (or store).

```
const state = proxy({
  obj: {
    count: 0,
    text: 'hello',
  },
})
```

If using the snapshot with accessing count,

```
const snap = useSnapshot(state)
snap.obj.count
```

it will re-render only if `count` changes.

If the property access is obj,

```
const snap = useSnapshot(state)
snap.obj
```

then, it will re-render if `obj` changes. This includes `count` changes and `text` changes.



Mobx vs. Valtio

Mobx

VS.

Valtio

Automatically optimizes renders

Many separate proxies

Requires wrapping app in Provider

Use higher order component

Large ecosystem

Automatically optimizes renders

Many separate proxies

No Provider required

Uses `useSnapshot` hook

Simple and focused



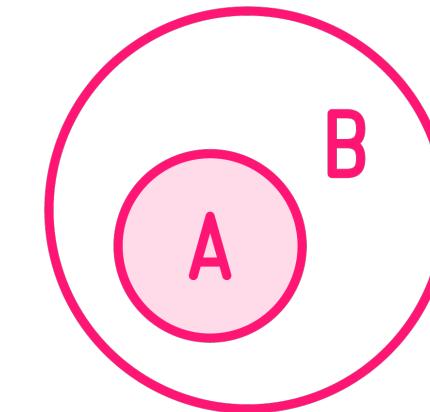
Valtio



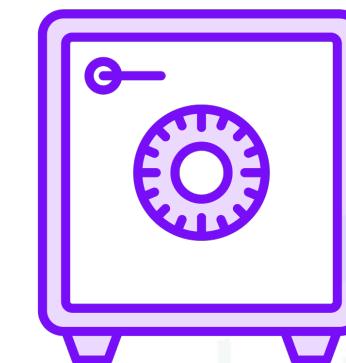
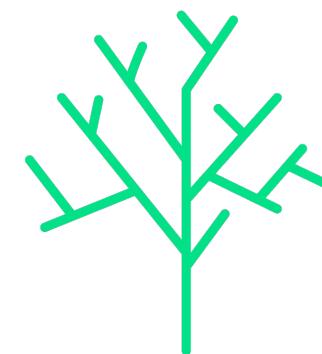
General vs. Specific



Mutable vs. Immutable



External vs. Internal



Auto vs. Manual

One Store vs. Multiple

Protected vs. Unprotected



Summary



Can mutate state

Automatically optimizes renders

Examples: Mobx and Valtio



Up Next:

Remote State

