Immer

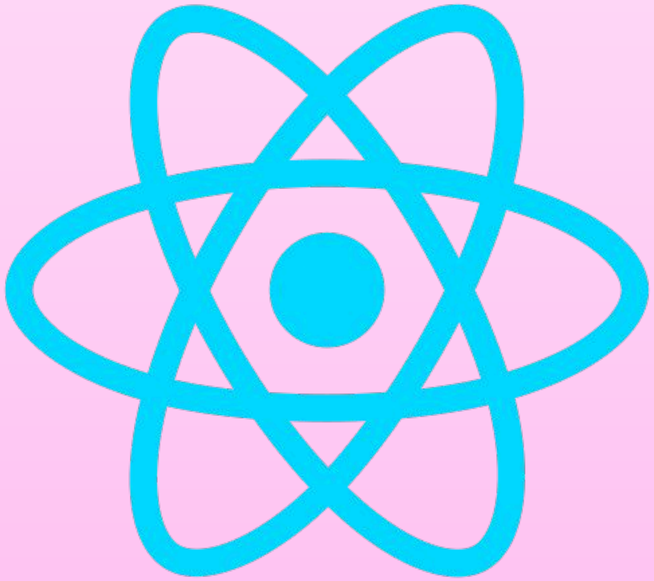
쉬운 불변성 유지

2025-08-02 김홍빈

Immer

독일어로 ‘Always(항상)’ 라는 뜻

immer가 나오게 된 배경



불변성을 지켜야하는 상태관리

불변성을 지키기 위한 방법

불변성을 지키기 위한 방법

만약 병합된 셀의 정보를 변경한다면?

```
const updatedTable = {
  ...table,
  cells: table.cells.map((rowCells, rIdx) =>
    rowCells.map((cell, cIdx) => {
      if (rIdx === row && cIdx === col) {
        return {
          ...cell,
          merge: {
            ...cell.merge,
            start: { row: newRow, col: newCol },
          },
        }
      }
      return cell
    })
  ),
}

```

불변성을 지키기 위한 방법

기존 방법

```
const updatedTable = {
  ...table,
  cells: table.cells.map((rowCells, rIdx) =>
    rowCells.map((cell, cIdx) => {
      if (rIdx === row && cIdx === col) {
        return {
          ...cell,
          merge: {
            ...cell.merge,
            start: { row: newRow, col: newCol },
          },
        }
      }
      return cell
    })
  ),
}
```

immer 적용

```
import { produce } from 'immer'

const updatedTable = produce(table, (draft) => {
  const cell = draft.cells[row][col]
  if (cell.merge.isMerged) {
    cell.merge.start = { row: newRow, col: newCol }
  }
})
```

immer 사용방법 - produce(기본)

```js
import { produce } from "immer";

const baseState = { count: 0 };

const nextState = produce(baseState, (draft) => {
  draft.count += 1; // 직접 수정하는 것처럼 동작
});

console.log(baseState); // { count: 0 }
console.log(nextState); // { count: 1 }
```

immer 사용방법 - useImmer(with react)

```jsx
import React, { useCallback } from "react";
import { useImmer } from "use-immer";

const TodoList = () => {
  const [todos, setTodos] = useImmer([
    {
      id: "React",
      title: "Learn React",
      done: true
    },
    {
      id: "Immer",
      title: "Try Immer",
      done: false
    }
  ]);

```

```jsx
const handleToggle = useCallback((id) => {
  setTodos((draft) => {
    const todo = draft.find((todo) => todo.id === id)
    todo.done = !todo.done
  })
}, [])

const handleAdd = useCallback(() => {
  setTodos((draft) => {
    draft.push({
      id: 'todo_' + Math.random(),
      title: 'A new todo',
      done: false,
    })
  })
}, [])

```

immer - set, map

```
1 import { enableMapSet } from 'immer';
2 import Router from './Router';
3
4 // immer에서 Map과 Set을 사용하기 위해 활성화
5 enableMapSet();
6
7
8 /**
9  * 에셋 저작도구 최상단 컴포넌트
10  */
11 export default function App() {
12     return <Router />;
13 }
14
```

immer 장점

간편하게 불변성 유지

spread operator를 중첩적으로 사용하지 않고도 간편하게 불변성 유지

자동 동결

한번 immer를 적용한 객체는 자동으로 내부 속성이 object.freeze() 적용되어 실수 방지

가독성

간결한 코드로 인해 가독성 향상

zustand middleware

```ts
import { create } from 'zustand'
import { immer } from 'zustand/middleware/immer'

type State = {
  count: number
}

type Actions = {
  increment: (qty: number) => void
  decrement: (qty: number) => void
}

export const useCountStore = create<State & Actions>()(
  immer((set) => ({
    count: 0,
    increment: (qty: number) =>
      set((state) => {
        state.count += qty
      }),
    decrement: (qty: number) =>
      set((state) => {
        state.count -= qty
      }),
  })),
)
```