

Michèle Legait

Senior Frontend Developer



{riviera dev}







Immutable state

"Never update an object, always create a new version"

Change detection easy and predictable.

```
if (oldState !== newState) {  
  // Something changed  
  //   → React: let's re-render!  
  //   → Redux: let's notify the listeners  
}
```

Enables powerful DevTools features:

- Compare before/after
- Undo/Redo
- Time-travel debugging



Aria

```
const gameState = {  
  player: {  
    name: "Aria",  
    level: 5,  
    inventory: [  
      { id: 1, name: "Health Potion", quantity: 3 },  
      { id: 2, name: "Traveler's Map" },  
      { id: 3, name: "Old boots" },  
    ],  
    stats: {  
      health: 87,  
      mana: 30,  
      equipment: {  
        weapon: "Iron Sword",  
        armor: {  
          body: "Iron Armor",  
          legs: "Traveler's Pants",  
        },  
      },  
    },  
  },  
};
```



Spread operator

• • •



Aria levels up

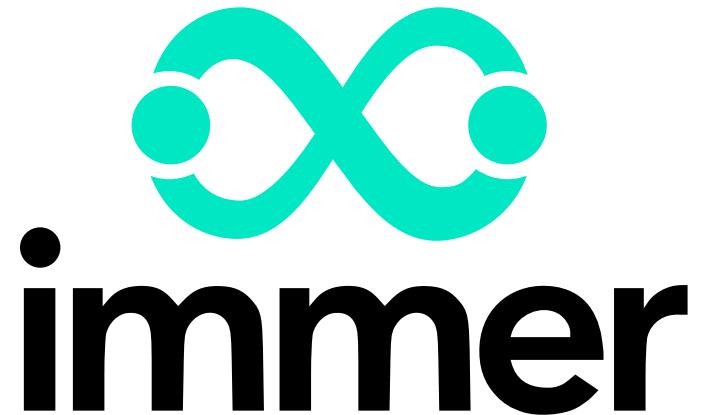
```
const newState = {  
  ...gameState,  
  player: {  
    ...gameState.player,  
    level: gameState.player.level + 1,  
  },  
};
```



Aria changes weapon

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    stats: {
      ...gameState.player.stats,
      equipment: {
        ...gameState.player.stats.equipment,
        weapon: "Flaming Sword",
      },
    },
  },
};
```





4.7kB (minified + gzipped)

produce

2016



Current



immer
→

Draft



immer
→

Next



Your edits here.

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    stats: {
      ...gameState.player.stats,
      equipment: {
        ...gameState.player.stats.equipment,
        weapon: "Flaming Sword",
      },
    },
  },
};
```



```
const newState = produce(gameState, (draft) => {
  draft.player.stats.equipment.weapon = "Flaming Sword";
});
```

Aria gets ready for battle

- She uses a health potion
- She equips a new helmet
- She drops her old boots
- She finds a rare item that increases her mana



JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    ],
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  },
  stats: {
    ...gameState.player.stats,
    mana: gameState.player.stats.mana + 10
    equipment: {
      ...gameState.player.stats.equipment,
      armor: {
        ...gameState.player.stats.equipment.armor,
        head: "Steel Helmet",
      },
    },
  },
},
```



```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    {
      id: 3,
      name: "Phoenix Feather",
      rarity: "legendary",
    },
    ],
    stats: {
      ...gameState.player.stats,
      mana: gameState.player.stats.mana + 10
      equipment: {
        ...gameState.player.stats.equipment,
        armor: {
          ...gameState.player.stats.equipment.armor,
          head: "Steel Helmet",
        },
      },
    },
  },
};
```



```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    ],
    stats: {
      ...gameState.player.stats,
      mana: gameState.player.stats.mana + 10
      equipment: {
        ...gameState.player.stats.equipment,
        armor: {
          ...gameState.player.stats.equipment.armor,
          head: "Steel Helmet",
        },
      },
    },
  },
};
```



```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    ],
    stats: {
      ...gameState.player.stats,
      mana: gameState.player.stats.mana + 10
      equipment: {
        ...gameState.player.stats.equipment,
        armor: {
          ...gameState.player.stats.equipment.armor,
          head: "Steel Helmet",
        },
      },
    },
  },
};
```



```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    ],
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  },
  stats: {
    ...gameState.player.stats,
    mana: gameState.player.stats.mana + 10
    equipment: {
      ...gameState.player.stats.equipment,
      armor: {
        ...gameState.player.stats.equipment.armor,
        head: "Steel Helmet",
      },
    },
  },
},
```



```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```

JS

```
const newState = {
  ...gameState,
  player: {
    ...gameState.player,
    inventory: [
      ...gameState.player.inventory
        .map((item) =>
          item.name === "Health Potion"
            ? { ...item, quantity: item.quantity - 1 }
            : item,
        )
        .filter((item) => item.name !== "Old Boots"),
    ],
    stats: {
      ...gameState.player.stats,
      mana: gameState.player.stats.mana + 10
      equipment: {
        ...gameState.player.stats.equipment,
        armor: {
          ...gameState.player.stats.equipment.armor,
          head: "Steel Helmet",
        },
      },
    },
  },
};
```



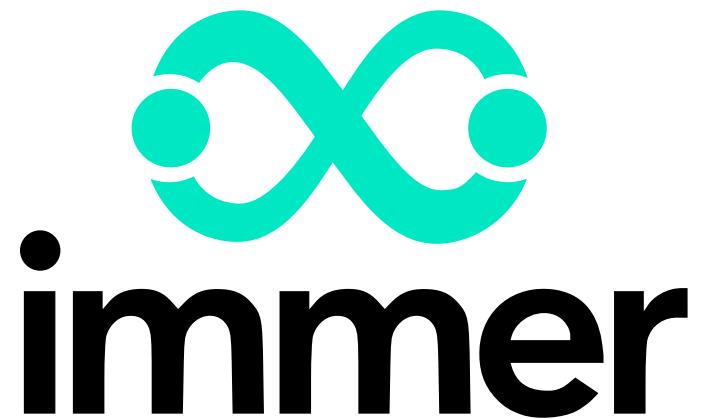
```
const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
  });
  draft.player.stats.mana += 10;
});
```



```
import { produce } from "immer";
import { gameState } from './state';

const newState = produce(gameState, (draft) => {
  const potion = draft.player.inventory.find(
    (i) => i.name === "Health Potion"
  );
  if (potion) {
    potion.quantity -= 1;
  }
  draft.player.stats.equipment.armor.head = "Steel Helmet";
  draft.player.inventory = draft.player.inventory.filter(
    (i) => i.name !== "Old Boots",
  );
  draft.player.inventory.push({
    id: 3,
    name: "Phoenix Feather",
    rarity: "legendary",
    quantity: 1
  });
  draft.player.stats.mana += 10;
});
```





<https://immerjs.github.io/immer/>

