Simple Store with useState

In this lesson, we learn a different pattern with useState and custom hooks, create a store in a couple of lines, and also make use of Immer.

WE'LL COVER THE FOLLOWING

Initial state

The initial state is the same as in the previous lessons.

```
const initialState = {
  todos: [
    { id: 1, title: 'Wash dishes' },
    { id: 2, title: 'Study JS' },
    { id: 3, title: 'Buy ticket' },
  ],
  query: '',
};
```

useValue

The initial state is only needed for `useState`.

```
const useValue = () => useState(initialState);
```

createContainer

We create a store with `createContainer`.

```
const {
  Provider,
  useTrackedState,
  useUpdate: useSetState,
} = createContainer(
  useValue,
);
```

Notice we renamed `useUpdate` to `useSetState`.

useSetDraft

Now, we create a wrapper function on `useSetState`.

```
const useSetDraft = () => {
  const setState = useSetState();
  return useCallback(
    draftUpdater => {
      setState(produce(draftUpdater));
    },
    [setState],
  );
};
```

This `useSetDraft` uses Immer's `produce`. Immer allows a mutating draft state while we get an immutably updated state in the end.

The entire store file

Our `store.js` is simply the concatenation of what we defined above.

```
// store.js

import { useState, useCallback } from 'react';
import { createContainer } from 'react-tracked';
import produce from 'immer';

const initialState = {
  todos: [
    { id: 1, title: 'Wash dishes' },
    { id: 2, title: 'Study JS' },
    { id: 3, title: 'Buy ticket' },
  ],
  query: '',
};

const useValue = () => useState(initialState);

const { Provider, useTrackedState, useUpdate: useSetState } = createContai
ner(
  useValue,
);

const useSetDraft = () => {
  const setState = useSetState();
  return useCallback(
    draftUpdater => {
      setState(produce(draftUpdater));
    },
    [setState],
  );
};

export { Provider, useTrackedState, useSetDraft };
```

Next

Let’s create various custom hooks based on the store.

In the next lesson, we create the first `useTodoList` hook.