Modifying the Counter App with React Tracked

In this lesson, we use React Tracked in the counter app and see how it behaves.

WE'LL COVER THE FOLLOWING

Installing React Tracked

In this course, the platform already has React Tracked installed.

If you run the code locally, please follow the instructions here.

Modify code

Previously, we had the following code for our counter app:

```
const Context = createContext(null);

const useGlobalState = () => {
  const value = useContext(Context);
  if (value === null) throw new Error('Please add GlobalStateProvider');
  return value;
};

const GlobalStateProvider = ({ children }) => (
  <Context.Provider value={useValue()}>{children}</Context.Provider>
);
```

This creates a context, a hook, and a provider.

In React Tracked, this is done by `createContainer`.

```
import { createContainer } from 'react-tracked';

const {
  useTracked: useGlobalState,
  Provider: GlobalStateProvider,
} = createContainer(useValue);
```

Notice we simply rename a hook and a provider from the default names in a container.

Check the app

Now, let's try the modified app.

```
import React from 'react';
require('./style.css');

import ReactDOM from 'react-dom';
import App from './app.js';

ReactDOM.render(
  <App />,
  document.getElementById('root')
);
```

Behavior with React Tracked

If you click the "+1" button in the app above, you'll notice that only two components re-render. If `count1` is changed, Count1 components are flashed, and if `count2` is changed, Count2 components are flashed.

Renders are optimized based on the state usage in components. This is effortless. The only change we made was to create a container using the library, no changes were made to the components and reducer.

Technique behind React Tracked

The library utilizes Proxy. It tracks how a state object property is accessed in render and marks it as "used." If the used property is changed by updates, it will trigger re-render. The use of Proxy is not new in JavaScript frontend

libraries, but tracking the usage in React render functions is unique to this library.

Next

In the next chapter, we will learn how to make a more realistic ToDo app.