

W15-P1: Setup AppProvider_xx for App_xx and get state data

The screenshot shows a VS Code editor with two files open: `Context_26.js` and `index.js`. The `Context_26.js` file defines a context and a provider, while `index.js` uses them in the application root.

Context_26.js

```
1 import React, { useState, useContext, useReducer, useEffect, } from 'react';
2 import Reducer_26 from './Reducer_26';
3
4 import cartItems from './data';
5
6 const AppContext_26 = React.createContext();
7
8 const initialState = {
9   loading: false,
10  cart: cartItems,
11  amount: 0,
12  total: 0
13 };
14
15 const AppProvider_26 = ({children}) => {
16   const [state, dispatch] = useReducer(Reducer_26, initialState);
17   return <AppContext_26.Provider value={...state}>
18     {children}
19   </AppContext_26.Provider>
20 };
21
22 const useGlobalContext_xx = () => {
23   return useContext(AppContext_26);
24 }
25
26 export { AppProvider_26, useGlobalContext_xx };
27
```

index.js

```
1 import React from 'react';
2 import ReactDOM from 'react-dom/client';
3 import './index.scss';
4 import App_26 from './App_26';
5
6 import { AppProvider_26 } from './Context_26';
7
8
9 const root = ReactDOM.createRoot(document.getElementById('root'));
10 root.render(
11   <React.StrictMode>
12     <AppProvider_26>
13       <App_26 />
14     </AppProvider_26>
15   </React.StrictMode>
16 );
17
```

Problems Panel

```
src/data.js
Line 1:1: Assign array to a variable before exporting as module default  import/no-anonymous-default-export

src/index.js
Line 12:5: Imported JSX component AppProvider_26 must be in PascalCase or SCREAMING_SNAKE_CASE  react/jsx-pascal-case
Line 13:5: Imported JSX component App_26 must be in PascalCase or SCREAMING_SNAKE_CASE  react/jsx-pascal-case

webpack compiled with 1 warning
```

W15-P2: replace data in CartContainer using global context

The screenshot displays a web development environment with three main components: a code editor on the left, a browser window in the center, and a component inspector on the right.

Code Editor (Left): The file `CartContainer_26.js` is open. It shows a React component `CartContainer` that uses `useGlobalContext_xx` to access global state. The code includes imports for `React`, `useState`, `CartItem`, and `useGlobalContext_xx`. The `clearCart` function is defined, and the component renders a list of items from the global context. The code is as follows:

```
1 import React, { useState } from 'react';
2 import CartItem from './CartItem_26';
3 // import data from './data';
4
5 import { useGlobalContext_xx } from '../Context_26';
6
7 const CartContainer = () => {
8   // const [cart, setCart] = useState(data);
9   // const [total, setTotal] = useState(0);
10  // const [amount, setAmount] = useState(0);
11  const { cart, total, amount, clearCart } = useGlobalContext_xx();
12  if (cart.length === 0) {
13    return (
14      <section className='cart'>
15        /* cart header */
16        <header>
17          <h2>your bag</h2>
18          <h4 className='empty-cart'>is currently empty</h4>
19        </header>
20      </section>
21    );
22  }
23  return (
24    <section className='cart'>
25      /* cart header */
26      <header>
27        <h2>your bag</h2>
28      </header>
29    </section>
30  );
31 }
```

Browser (Center): The browser shows a page titled "YOUR BAG" at `localhost:3000`. It displays a list of items in a shopping cart:

- Samsung Galaxy S8: \$399.99, 1 item, remove button
- Google Pixel: \$499.99, 1 item, remove button
- Xiaomi Redmi Note 2: \$699.99, 1 item, remove button
- Samsung Galaxy S7: \$599.99, 1 item, remove button

The total amount is displayed as **\$999.9**.

Component Inspector (Right): The component inspector shows the state of the `App_26` component. The `GlobalContext_xx` hook is expanded, showing the following state:

```
{
  Context: {
    amount: 4,
    cart: Array(4),
    loading: false
  },
  cart: [
    { amount: 1, id: "rec1JZlfCIBOPd..." },
    { amount: 1, id: "recB6qcHPxb62Y..." },
    { amount: 1, id: "recdRxBS14Rr2..." },
    { amount: 1, id: "recwTo160XST3P..." }
  ],
  total: 999.9,
  clearCart: f clearCart() {}
}
```

W15-P3: put clear cart function in global context

```
Context_26.js  JS CartItem_26.js  JS index.js M  JS data.js  JS Reducer_26.js  JS CartContainer_26.js

src > JS Reducer_26.js > Reducer_xx
1 const Reducer_xx = (state, action) => {
2   if(action.type === 'CLEAR_CART') {
3     return {...state, cart: [], amount: 0, total: 0 }
4   }
5   if(action.type === 'INCREASE') {
6     let tempCart = state.cart.map( (cartItem) => {
7       if(cartItem.id === action.payload) {
8         return {...cartItem, amount: cartItem.amount + 1}
9       }
10      return cartItem;
11    });
12    return {...state, cart: tempCart}
13  }
14  if(action.type === 'GET_TOTALS') {
15    let {total, amount} = state.cart.reduce((cartTotal, cartItem) =>{
16      const {price, amount} = cartItem;
17      const itemTotal = price * amount;
```

```
JS Context_26.js
src > JS Context_26.js > ...
20 }, (state.cart));
21
22 const clearCart = () => {
23   dispatch({type: 'CLEAR_CART'});
24 }
25
26 const increase = (id) => {
27   dispatch({type: 'INCREASE', payload: id});
28 }
29
30 return <AppContext_xx.Provider value={{...state, clearCart, increase}}>
31   {children}
32 </AppContext_xx.Provider>
33
34
35 const useGlobalContext_xx = () => {
36   return useContext(AppContext_xx);
37 }
```

```
JS CartContainer_26.js
src > components > JS CartContainer_26.js > ...
6
7 const CartContainer = () => {
8   // const [cart, setCart] = useState(data);
9   // const [total, setTotal] = useState(0);
10  // const [amount, setAmount] = useState(0);
11  const { cart, total, clearCart } = useGlobalContext_xx();
12  if (cart.length === 0) {
13    return (
14      <section className='cart'>
15        <div>
16          <h2>your bag</h2>
17          <h4 className='empty-cart'>is currently empty</h4>
18        </div>
19      </section>
20    );
21  }
```

```
JS CartContainer_26.js
src > components > JS CartContainer_26.js > ...
35   <div>
36     <div>
37       <div className='cart-total'>
38         <div>
39           <div>
40             total <span>${total}</span>
41           </div>
42         </div>
43         <div>
44           <div>
45             <button className='btn clear-btn' onClick={clearCart}>
46               clear cart
47             </button>
48           </div>
49         </div>
50       </div>
51     </div>
52   </div>
53 );
54 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Local: http://localhost:3000
On Your Network: http://172.20.10.3:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled successfully

main* 0 0 0

Watch Sass Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

線上課程 - 自由安排時間 | 繁體中文 Notion 教學 | 你所不知道的 c 語言 | 資料分析 實體問答模型 | Monkeytype | iClass | useReducer Cart Con

localhost:3000

UseReducer

YOUR BAG

is currently empty

AppProvider_xx

- Context.Provider
- App_26
- Navbar
- CartContainer

CartContainer

props

new entry: ""

collapse prop value

Context: {amount: 0, cart: Array(0), c...

loading: false

cart: []

amount: 0

total: 0

clearCart: f clearCart() {}

increase: f increase() {}

rendered by

App_26

createRoot()

react-dom@18.2.0

source

Console What's New Issues

Default levels 1 Issue: 1

W15-P4: provide increase and get total function in global context

```
src > JS Context_26.js > [e] AppProvider_xx
2 import Reducer_26 from './Reducer_26';
3
4 import cartItems from './data';
5
6 const AppContext_xx = React.createContext();
7
8 const initialState = {
9   loading: false,
10  cart: cartItems,
11  amount: 4,
12  total: 999.9
13 }
14
15 const AppProvider_xx = ({children}) => {
16   const [state, dispatch] = useReducer(Reducer_26, initialState);
17
18   useEffect(() => {
19     dispatch({type: 'GET_TOTALS'});
20   }, [state.cart]);
21
22   const clearCart = () => {
23     dispatch({type: 'CLEAR_CART'});
24   }
25
26   const increase = (id) => {
27     dispatch({type: 'INCREASE', payload: id});
28   }
29
30   return <AppContext_xx.Provider value={{...state, clearCart, increase}}>
31     {children}
32   </AppContext_xx.Provider>
33 };
34
35 const useGlobalContext_xx = () => {
36   return useContext(AppContext_xx);
37 }
```

```
src > JS Reducer_26.js > ...
1 const Reducer_xx = (state, action) => {
2   if(action.type === 'CLEAR_CART') {
3     return {...state, cart: [], amount: 0, total: 0 }
4   }
5
6   if(action.type === 'INCREASE') {
7     let tempCart = state.cart.map( (cartItem) => {
8       if(cartItem.id === action.payload) {
9         return {...cartItem, amount: cartItem.amount + 1}
10      }
11      return cartItem;
12    });
13    return {...state, cart: tempCart}
14  }
15
16  if(action.type === 'GET_TOTALS') {
17    let {total, amount} = state.cart.reduce((cartTotal, cartItem) =>{
18      const {price, amount} = cartItem;
19      const itemTotal = price * amount;
20      cartTotal.total += itemTotal;
21      cartTotal.amount += amount;
22      return cartTotal;
23    },{
24      total: 0,
25      amount: 0
26    });
27    total = parseFloat(total.toFixed(2));
28    return {...state, total, amount}
29  }
30 }
31
32 export default Reducer_xx;
```

Local: http://localhost:3000
On Your Network: http://172.20.10.3:3000

Note that the development build is not optimized.
To create a production build, use `npm run build`.

webpack compiled **successfully**

main* Watch Sass Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} JavaScript Go Live Prettier


線上課程 - 自由安排時間 × 繁體中文 Notion 教學 × 你所不知道的 c 語言 - × 資料分析 實體關聯模型 × Monkeytype × iClass × useReducer Cart Con ×

localhost:3000

UseReducer

0


YOUR BAG



Samsung Galaxy S8

\$399.99


remove



Google Pixel

\$499.99


remove



Xiaomi Redmi Note 2

\$699.99

remove



Samsung Galaxy S7

\$599.99

remove

2

2

1

2

Total

\$3699.93

CLEAR CART

AppProvider_xx

Context.Provider

App_26

Navbar

CartContainer

CartItem key="rec1JZlfcIB0..."

CartItem key="recB6qcHPxb6..."

CartItem key="recdRxBsE14R..."

CartItem key="recwTo160XST..."

App_26

props

new entry: ""

hooks

Collapse prop value

Context: {amount: 7, cart: Array(4), c...

loading: false

cart: [{}, {}, {}, {}]

amount: 7

total: 3699.93

clearCart: f clearCart() {}

increase: f increase() {}

rendered by

createRoot()

react-dom@18.2.0

source

src/index.js:13

Console

What's New

Issues

Default levels

1 Issue: 1

W15-P5:

851f3c0	WeiYiHuangfelix	Thu Dec 15 15:54:53 2022	+0800	W15-P4: provide increase and get total function
416fbad	WeiYiHuangfelix	Thu Dec 15 15:54:26 2022	+0800	W15-P3: put clear cart function in global conte
45e439e	WeiYiHuangfelix	Thu Dec 15 15:53:52 2022	+0800	W15-P2: replace data in CartContainer using glo
239a35b	WeiYiHuangfelix	Thu Dec 15 15:52:56 2022	+0800	W15-P1: Setup AppProvider_xx for App_xx and get