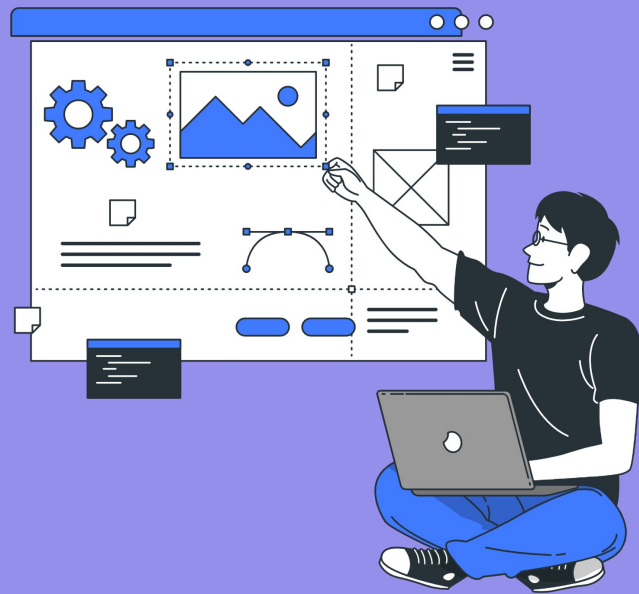


useRefs , usecontext, custom hooks

Relevel
by Unacademy



App we will be building today - Ecommerce Application

- Today, we will make our Ecommerce Application using React hooks useContext, useRef, and Custom hooks.
- App will have three screens, i.e., Login page, Home/Product Listing Page, and a Product Detail page.
- It is a complete frontend-based application made using the concepts of react.



Ecommerce Application - Login Screen

Ekart Login 


Get access to your Orders, Wishlist and Recommendations

Name

Phone Number


LOGIN

Ecommerce Application - Home/Product Listing Screen

Ekart 


Hello Tanay Tapanshu

LOGOUT




Fjallraven - Foldsack No. 1 Backpack, Fits 15 Laptops

₹ 109.95




Mens Casual Premium Slim Fit T-Shirts

₹ 22.3




Mens Cotton Jacket

₹ 55.99



Mens Casual Slim Fit

₹ 15.99



John Hardy Women's Legends Naga Gold & Silver Dragon Station Chain Bracelet

₹ 695

Ecommerce Application - Product Detail Screen



Mens Casual Slim Fit

men's clothing

₹15.99

Description

The color could be slightly different between on the screen and in practice. / Please note that body builds vary by person, therefore, detailed size information should be reviewed below on the product description.

Add to Cart

Buy Now

What are we going to learn today

The knowledge of the following concepts is necessary to build the E-commerce React app today -

- useRef with examples
- useContext definition with examples
- custom hook along with its implementation
- Implementation of the above concepts in our project



But Before we start, let's recall react hooks and useState hook which we have learned in previous classes



React Hooks and useState



- React hooks are the new addition to the react.
- Hooks allow to use of state and other react features inside a functional component, which was earlier only possible for class-based components.
- Using **useState** we can use a state variable inside a functional component
- We use a state variable when DOM has something which dynamically manipulated
- Example -

```
const [data, setData] = useState(null);
```


useRef

- It is used to create a reference to the DOM element
- uses of ref :
 - I> Used in managing the focus or text selection.
 - II> Triggering the animations.
 - III> Integrating the third-party DOM libraries.
- Example of useRef -

```
function App() {  
  const textElement = useRef()  
  return (  
    <div className="App">  
      <h1 ref={textElement}>Relevel  
Learning</h1>  
    </div>  
  );  
}
```

.current property

- Once a reference is created for an element, it can access the properties of the object using the “.current” property
- Using “.current” we can access and manipulate innerHTML, innerText, offsetHeight , offsetWidth, clientHeight , clientWidth, draggable and many other properties values



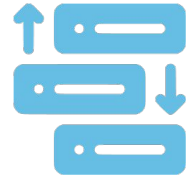
Example

In the above example we are changing the inner text of h1 using “.current”, similarly, we can change other properties as well

```
function App() {  
  const textElement = useRef()  
  useEffect(() => {  
    console.log(  
      "This our ref element", textElement  
    );  
    textElement.current.innerText = "Hi, I am learning useRef"  
  }, []);  
  return ( <div className="App">  
    <h1 ref={textElement}>Let's learn useRef </h1>  
    </div>  
  );  
}
```

useContext

- useContext is used for creating common data that can be accessed by components hierarchy without passing the props down to each level.
- It provides the data passage through the component tree without passing it from one component to another.



React.createContext

- Creates a context object, when react renders a component that has subscribed to this object/context object then it will read the current value from the closest matching provider.

```
const ThemeContext = React.createContext();
```

Context.Provider

- Allows consuming components to subscribe to context changes.

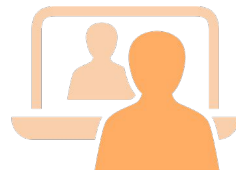
```
<ThemeContext.Provider value={{theme, toggleTheme}}>

  {children}

</ThemeContext.Provider >
```

Context.Consumer

- Consumer is defined as a component that subscribes to context changes. It subscribes to a context within a function component.



```
<MyContext.Consumer>

  {value => /* render something based on the context value */}

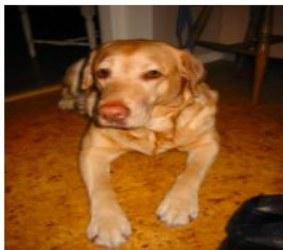
</MyContext.Consumer>
```

Example of Context API

- In this example we are managing the theme using Context.

Code - <https://codesandbox.io/s/dawn-moon-r9p8p?file=/src/index.js>

Hi I am learning Custom Hooks



India

Toggle the theme

Hi I am learning Custom Hooks



India

Toggle the theme

Custom Hooks

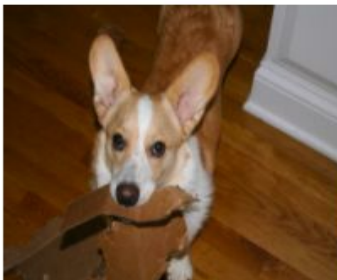
- The custom hook is a javascript function.
- Custom hook uses multiple hooks inside the function and starts with “use.”
- It is mainly used for code reusability.



Example of Custom Hooks

- Code - <https://codesandbox.io/s/dark-snow-8dlxq?file=/src/App.js>

Hi I am learning Custom Hooks



United States

E-commerce App Project code

- Code - <https://github.com/TANAYTAPANSHU/ecommerce-app-react>

Ekart Login 

Get access to your Orders, Wishlist and Recommendations

Name

Phone Number

LOGIN



Fjallraven - Foldsack No.
1 Backpack, Fits 15
Laptops

₹ 109.95



Mens Casual Premium
Slim Fit T-Shirts

₹ 22.3



Mens Cotton Jacket

₹ 55.99



Mens Casual Slim Fit

₹ 15.99



John Hardy Women's
Legends Naga Gold &
Silver Dragon Station
Chain Bracelet

₹ 695



Mens Casual Slim Fit

men's clothing

₹15.99

Description

The color could be slightly different between on the screen and in practice. / Please note that body build vary by person, therefore, detailed size information should be reviewed below on the product description

Add to Cart

Buy Now

Assignment

- Create a Counter application with a custom hook that will update the count and title. For example, the count should be updated after each click on the button, and the webpage/document title should be updated like this : 'You clicked {no of count} times'.

Also try adding a theme manager for the application



You clicked 9 times

Click me

Upcoming Class Teasers

- Making API Requests with React - Get/Post
- Building Lists
- Intro to rest APIs



Thank you!