React Context

* Context API provide a way to pass data throught component tree without having to pass props down manually at every level.
* To create context :

import React from 'react';

const AuthContext = React.createContext(

    {

        isLoggedIn : false

    }

);

export default AuthContext;

* Context.Provider : to provide context to components

So all component inside AuthContext.Provider have access to context data.

Also all the childrens of that components have access to context

      <AuthContext.Provider>

        <MainHeader isAuthenticated={isLoggedIn} onLogout={logoutHandler} />

        <main>

          {!isLoggedIn && <Login onLogin={loginHandler} />}

          {isLoggedIn && <Home onLogout={logoutHandler} />}

        </main>

      </AuthContext.Provider>

Note: techniqually if we provide default value then provider is not required.

Provider is used to provide value to context

<AuthContext.Provider

      value={{

        isLoggedIn: false, //We Can provide here dynamic value

      }}

    >

      <MainHeader onLogout={logoutHandler} />

      <main>

        {!isLoggedIn && <Login onLogin={loginHandler} />}

        {isLoggedIn && <Home onLogout={logoutHandler} />}

      </main>

    </AuthContext.Provider>

* Context.Consumer : to consume value of context
  + There are two method to consume data from Context

Method 1: using Context.Consumer

<AuthContext.Consumer>

      {(context) =>{

        return(

          <nav className={classes.nav}>

        <ul>

          {context.isLoggedIn && (

            <li>

              <a href="/">Users</a>

            </li>

          )}

        </ul>

      </nav>

        )

      }}

    </AuthContext.Consumer>

Method 2: using useContext( ) hook

useContext hook return context value.

import React, {useContext} from "react";

import AuthContext from "../../store/auth-context";

const Navigation = (props) => {

  const context = useContext(AuthContext);

  return (

    <nav className={classes.nav}>

      <ul>

        {context.isLoggedIn && (

          <li>

            <a href="/">Users</a>

          </li>

        )}

      </ul>

    </nav>

  );

};

export default Navigation;

\*Note: We can also provide function to Context

* Limitations:
  + React context is not optimized for high frequency changes.
  + React Context also shouldn’t be to replace All component communication and props
* Note :
  + If want to store some value to context and update it on some condition the use following technique
  + <https://mindsers.blog/en/updating-react-context-from-consumer/>

import React from "react";

const CountContext = React.createContext({

  value: 0,

  increment: () => {},

  decrement: () => {},

});

export default CountContext;

import React, { useState } from "react";

import Counter from "./components/Counter";

import CountContext from "./store/count-context";

const App = () => {

  const [count, setCount] = useState(0);

  const value = {

    value: count,

    increment: () => {

      setCount(count + 1);

    },

    decrement: () => {

      setCount(count - 1);

    },

  };

  return (

    <CountContext.Provider value={value}>

      <Counter />

    </CountContext.Provider>

  );

};

export default App;

import React, { useContext } from "react";

import CountContext from "../store/count-context";

const Counter = () => {

  const counterContext = useContext(CountContext);

  return (

    <React.Fragment>

      <button onClick={counterContext.increment}>+</button>

      <span>

        {counterContext.value}

      </span>

      <button onClick={counterContext.decrement}>-</button>

    </React.Fragment>

  );

};

export default Counter;