

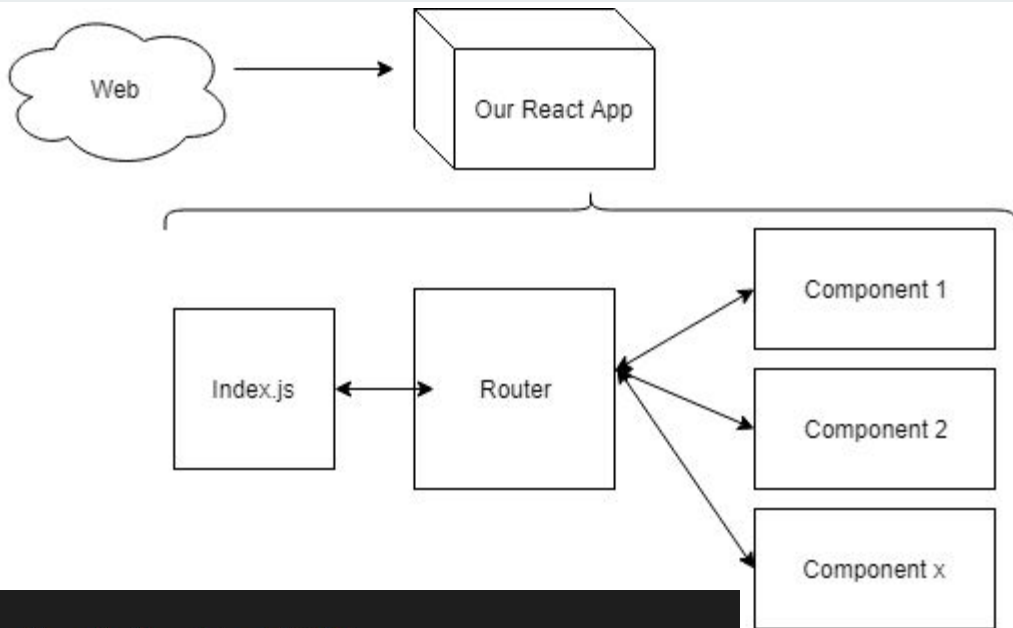


Learn React - Intermediate

Kiran Pachhai

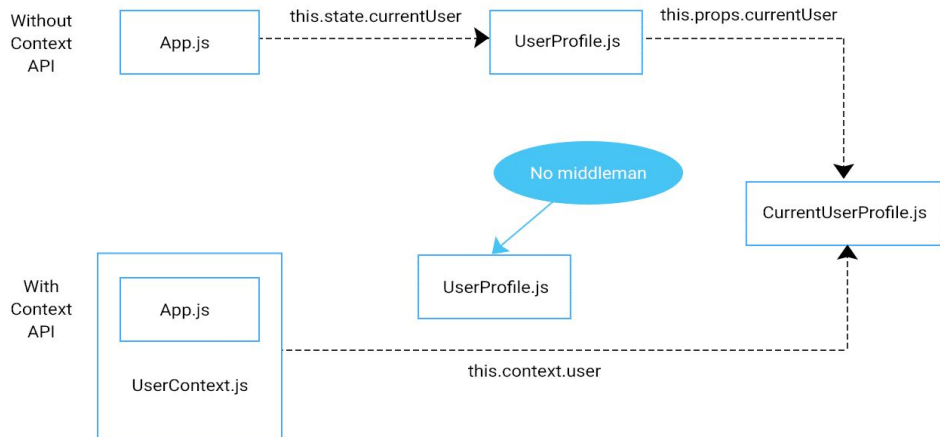
Intro to Routing

- A popular library for handling routing in React applications.
- Allows you to define routes using components
- Provides features like linking to different routes, protecting routes with authentication, etc
- Define your routes using the `<Route>` component.
- Also provides other useful components like `<Link>`, `<Switch>`, and `<Redirect>`



```
<Router>
<div>
  <h2>React Router Step By Step Tutorial</h2>
  <nav>
    <ul>
      <li><Link to={'/'}> Home </Link></li>
      <li><Link to={'/contact'}> Contact</Link></li>
      <li><Link to={'/about'}> About</Link></li>
      <li><Link to={'/services'}> Services</Link></li>
    </ul>
  </nav>
  <Switch>
    <Route path = "/" exact component = {Home}></Route>
    <Route path = "/contact" component = {Contact}></Route>
    <Route path = "/about" component = {About}></Route>
    <Route path = "/services" component = {Services}></Route>
  </Switch>
</div>
```

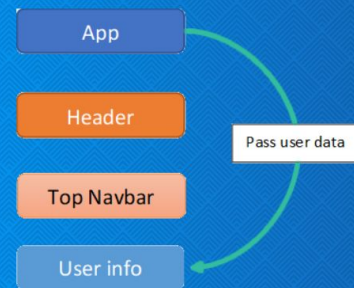
Context



CronJ

- Provides a way to share values like a global variable between components
- No need to explicitly pass a prop through every level of the component tree
- Need to import the `useContext` hook from React
- Use context sparingly and to avoid using it as a replacement for props
- Can make your code more difficult to understand and maintain if overused

Context - ReactJS



Fragments

```
render() {  
  return (  
    <>  
    <td>Hello</td>  
    <td>World</td>  
    </>  
  );  
}
```

**No Parent
Element**

React.Fragments

- A way to group a list of children without adding extra nodes to the DOM
- `div>` element adds an extra node to the DOM and can cause unnecessary re-renders
- Use fragments by adding `<>` at the start of the list and `</>` at the end of the list
- Can improve the performance of your React application
- Supported in React 16.0 and later versions