DEFAULT REDIRECTION WHEN NO VALID PATH IS FOUND –

<Router>

        <Route path="/" exact>

          <Users />

        </Route>

        <Redirect path="/" />

</Router>

npm install --save react-router-dom@4.2.2

USE SWITCH TO AVOID AUTOMATIC REDIRECTION –

<Router>

      <Switch>

      <Route path="/" exact>

          <Users />

      </Route>

      <Route path="/places" exact>

          <Places />

      </Route>

      <Redirect path="/" />

      </Switch>

</Router>

**Links in React**

In regular apps without a library like React, links are created with an anchor tag, as shown below.

<a href="https://pluralsight.com">Visit Pluralsight</a>

Bear in mind that this also works in React. This method of routing causes a full page refresh, when in reality the only thing that changed on the new route might be just an image and some text. In some cases, a full page refresh is useful, but in most cases, it's not needed.

So use Link

**Presentational and Stateful components –**

Now what does this difference mean for you?How does this impact the way you build your components?

Well it's not really impacting the way you build your components,it's just important to be aware that there are these two types of components in React - smart and dumb ones, stateful and presentational ones and it shouldn't impact how you build your components but you should be aware of the fact that it's totally normal to have way more presentational components, dumb components than stateful components.

Typically you have a few components where you manage your data, where you fetch data and so on (stateful) and then you have way more components that simply control how things are presented and how things are laid out.(presentational).

**4) Organization –**

Keep react libraries files on top and user-defined libraries below them.

**5) props.children / this.props.children –**

What even is ‘children’?

The React docs say that you can use props.children on components that represent ‘generic boxes’ and that ‘don’t know their children ahead of time’.

My simple explanation of what this.props.children does is that it is used to display whatever you include between the opening and closing tags when invoking a component.

.

**6) Use react portals –**

React Portal is a first-class way to render child components into a DOM node outside of the parent DOM hierarchy (root element) defined by the React. The Portal's most common use cases are when the child components need to visually break out of the parent container. (like a screen overlay). Semantically this is much better , rather then nesting it in one single root

**7)**