

# React Router

In this lab we'll implement routing for our bookstore. We'll create a navigation menu and make the links work to bring up different components based on what the user requests when clicking.

First, we'll need some other components to navigate to. Let's create those.

1. Create a new component called `NotFound`. It should just tell the user they've asked for something we don't have. Look at one of these for inspiration if you like: <http://bit.ly/404Pages>
2. Create another new one called `Login`. It should have textboxes for username and password and a button labelled "log in".
3. Create a third called `HomePage` with welcome text and other landing page things.
4. And lastly create an `AboutUs` component with lots and lots of details about our company. Just kidding, make it a placeholder page.

## Adding a Router

Routing is limited to the areas of our app that is surrounded by a Router. We want to be able to get to those pages (components) so let's put the whole App inside a `BrowserRouter`.

5. Open `index.js` and find where you're rendering `<App />`. Wrap that in a `<BrowserRouter>..`
6. You'll need to import `{ BrowserRouter }` from `'react-router-dom'`.
7. Run and test. Make sure you haven't broken anything yet.

When you do, you'll see that we're still seeing our product list and cart. We really only want the user to go there when they browse to that component. So let's not make it the main component.

8. If it isn't the main component, it shouldn't be called `"App.js"`. Refactor it to be named `BookStore.js`. Don't forget that you'll want to change the name of the file and the component itself.

Much better! The component is named well and won't come up by default. But we really do need a main component and `App` is an appropriate name for it. Besides, we needed a place to hold our Routes. And since this is the page that never goes away, we can put a static navigation bar (aka menu) here.

9. Create a new `App.js`. Inside the returned JSX, put a new `<div with a className="container">` as your root element. Inside of it, put a `<h1>I'm the main component</h1>`
10. Run and test. You should be able to see your new App component. If not, work on it until you can.

## The start of Routing

Now the fun begins! Let's add Routing.

11. Edit `App.js` and add a `<Switch></Switch>` to the JSX. This will tell React Router to take the first route that matches the URL.
12. Inside that `<Switch>`, put in Route components for each of the components you created above. Each should have a `path` attribute and a `component` attribute. Here's an example for `AboutUs`.

```
<Route path="/about" component={AboutUs} />
```

13. Go ahead and add one more for the BookStore component we just created.
14. This will not work yet because we haven't imported these things from their library. Add an import statement at the top for Switch and Route. These are found in the react-router-dom.
15. Also you've referenced five other components that you've written. import those as well.
16. Run and test again. Do you see your app component? Not too magic yet, right? Well, change the URL adding "/about" to it. Do you now see your AboutUs component? Keep trying for all of the Routes you just added.

## Creating a menu

17. We've got Routing working and have created some cool Routes. Now let's add a way to get to them.
18. Open App.js in your IDE again.
19. At the start of the JSX, put a <nav></nav>. (This just behaves like a <div> but is more appropriate for a list of links).
20. Inside your new <nav>, add a Link for each of the pages you just created. Here's an example for AboutUs:  

```
<Link to="/about">About the bookstore</Link>
```
21. Ah! There's another component that needs to be imported. Link is found in react-router-dom alongside Switch and Route. Add it please.
22. Run and test. You should see the Links rendered as clickable links on the page. When you click each, it should show that component below where the Switch is.

Congratulations, you implemented some basic React Routing.

23. Bonus! If you have extra time, try this challenge. Since we're already using Bootstrap, style your list of Links to look more like a menu. Here's a start:

```
<nav className="navbar navbar-expand-lg">
  <Link className="navbar-brand" to="/home">Our bookstore</Link>
  <ul className="navbar-nav mr-auto">
    <li className="nav-item">
      <Link to="/bookstore" className="nav-link">Products</a>
    </li>
    <li className="nav-item">
      <Link to="/about">About the bookstore</Link>
    </li>
    ...
  </ul>
</nav>
```