**Full Stack IV– Lab 5**

* React Router

**Developer Note:**

* Create a new **create-react-app** application for this lab.

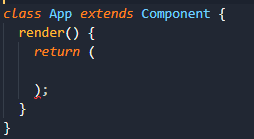
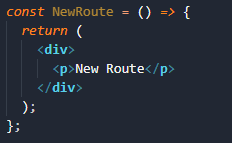
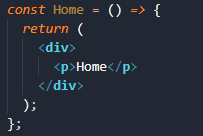
**References:**

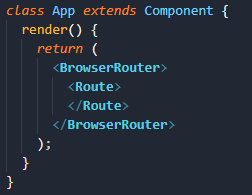
<https://www.npmjs.com/package/react-router>

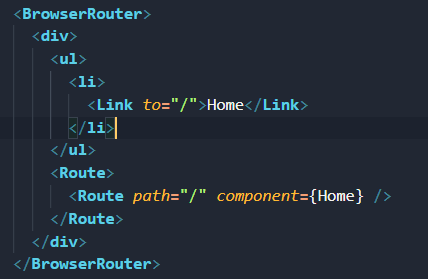
**Exercise 1 – Basic Routes**

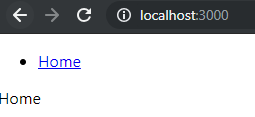
1. Create a new React application using **create-react-app**
2. Install ***React Router*** using ***npm install*** command.

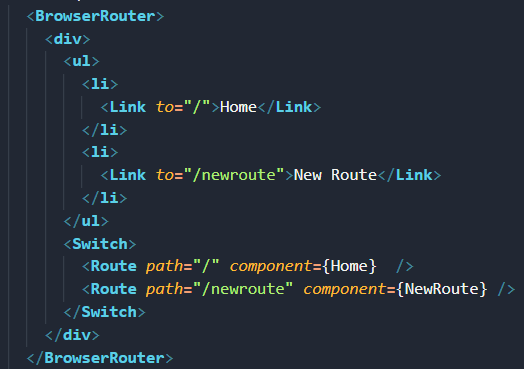


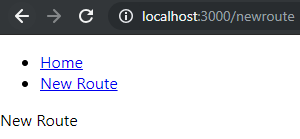
1. Navigate to the ***App.js*** file and remove the default JSX in the App component render method.  
   
2. In the ***App.js***, add an import ***BrowserRouter*** and ***Route*** from ***react-router-dom***  
     
   
3. In the ***App.js***, above the App component add the following function component***.***
4. Create a home component in the same ***App.js***  
     
   
5. In the ***App.js*** component start setting up the Routes by adding the ***BrowserRouter*** and ***Route*** to the render method.



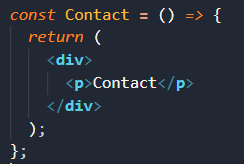
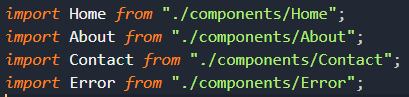
1. To setup our routes we need to add Link and ***Switch*** to the import statement from react-router-dom  
     
   
2. Add the first link and default path to load the ***Home*** component.  
     
     
     
   The browser should output as follows



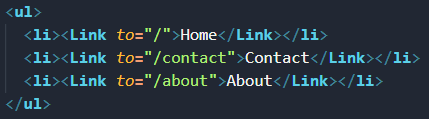
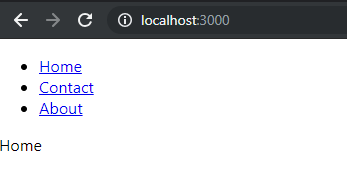
1. We need to add more Routes and Link to the Browser router. Use the ***Switch*** statement in this way to allow for multiple route paths. Test out the new routing links in the browser.  
     
   
2. Testing the application the New Route will not working, since the default route “/” is being matched first when “/newroute” is requested. To solve this add the keyword exact to the default Home route.  
   



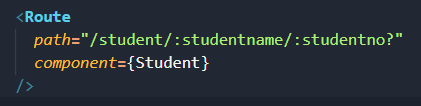
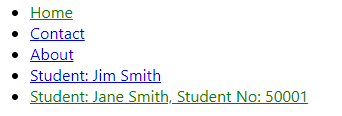
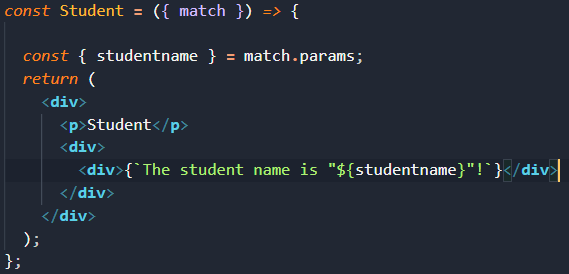
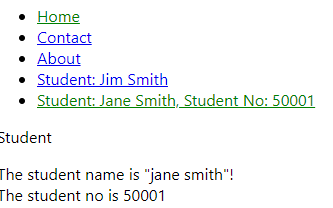
**Exercise 2 – Multiple Routes**

1. Create a component folder and then three new functional components: About, Contact, Error. Move the code for Home component in a file and place in the components folder.
2. In each of the component files add similar code to the following. In render the name of the component in the JSX will be displayed  
     
   
3. Import all of the components into the ***App.js*** file.  
   
4. Add the new routes to the ***BrowserRouter*** as follows:

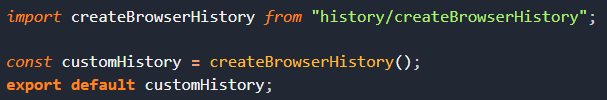
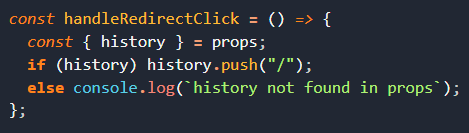


1. Add the new navigation links to the ***JSX*** unordered list to trigger the routes.  
     
   
2. Create a new ***Navigation*** component and move all the navigation links into the file. Import it into the ***App.js*** and include the component in the render in place of the <ul></ul> links.

**Exercise 3 – Router Parmeters**

1. Create a new component named ***Student*** and a new Link in the ***Navigation*** component.
2. Add a colon for a route parameter for studentname, followed by another optional route parameter for studentno  
   
3. Update the navigation links to display and pass the following route parameters.  
   
4. Add the following code in the ***Student*** component to output the ***mandatory*** ***route parameter*** for ***Student name***.  
   
5. Update the ***Student*** component to display the ***optional route parameter*** for ***Student no.*** *The output when navigating to this route should be as follows.*

**Exercise 4 – Redirecting with History API**

1. We need to create a ***History*** module and export it, so that we can use the History API in the ***BrowserRouter***. To do this, we need to create a ***History.js*** file with the following code.  
     
   
2. Next, import the history module in the ***App.js*** and include the history object in the ***BrowserRouter*** to make it available to the entire application   
     
   
3. Create a new navigation link, route and functional component named ***Redirect***.
4. The ***history object*** will be available in the component props parameter. Add the following event handler in the ***Redirect*** component.  
   
5. Add a button with a click handler to call the ***handleRedirectClick*** method and redirect the route to the home page.  
   