

30 Days Of React: React Router



LinkedIn



Follow @asabeneh

776

Author: [Asabeneh Yetayeh](#)

October, 2020

<< [Day 16](#) | [Day 18](#) >>



- [React Router](#)
 - [What is React Router ?](#)
 - [BrowserRouter](#)
 - [Route](#)
 - [Switch](#)
 - [NavLink](#)
 - [Nested Routing](#)
 - [Redirect](#)
 - [Prompt](#)
- [Exercises](#)
 - [Exercises: Level 1](#)
 - [Exercises: Level 2](#)
 - [Exercises: Level 3](#)

React Router

What is React Router ?

You may have not heard of the word route or router before and it might be necessary to define it here. The literal meaning of route is a path or a way to get to somewhere. The meaning in React is also similar to the literal meaning. React Router is by itself a React component which allows you to navigate between React components.

In this section, you will get started how to use React router but it may not have plenty of information about it. In case you prefer to learn from the official website of React Router you can get [here](#).

As we have cleared out the very beginning that React is a single page application which has only one index.html page in the entire application. When we implement a React Router the different components get render on the index.html page at same time or different time base on different logic and conditions. React Router has different versions and the latest version is React Router 5. We will use React Router version 4 for this challenge. Let's get started by installing the React Router packages.

```
Asabeneh@DESKTOP-KGC1AKC MINGW64 ~/Desktop/30-days-of-react$ npm install react-router-dom
```

Let's implement a simple routing using the boilerplate codes we have been creating in the previous days. First of all, import the *react-router-dom* and we can extract all the necessary components we need for routing from react-router-dom.

```
import React from 'react'
import {
  BrowserRouter,
  Route,
  NavLink,
  Switch,
  Redirect,
  Prompt,
  withRouter,
} from 'react-router-dom'
```

We may not all these components in every project but it is good to know that it exists.

BrowserRouter

BrowserRouter is a parent component which allows to wrap the application route. Using the BrowserRouter we can access the browser history. Sometimes it can renames as router.

```
import React from 'react'
import { BrowserRouter as Router } from 'react-router-dom'
```

Let's make make use of BrowserRouter to make a navigation for a React application.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router } from 'react-router-dom'

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <h1>React Router DOM</h1>
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

We wrapped our application with BrowserRouter or Router and it works smoothly as it used to be. Let's create Home, About, Contact, Challenge component and route to the different components. In addition to the components, we should import the Route component from react-router-dom.

Route

The Route component allows to navigate between components. It is a pathway from one component to another. The Route component has two required props: the path and component or render. The path props is where the component has to be rendered and the component props is the component which has to be rendered in that specific path. To see your component try to request /home route.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Route path='/home' component={Home} />
        </div>
      </Router>
    )
  }
}
```

```
const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

Let's add some more components to our route.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Route path='/home' component={Home} />
          <Route path='/about' component={About} />
          <Route path='/contact' component={Contact} />
          <Route path='/challenges' component={Challenges} />
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

As you can see the above example, all the routes have slush(/). We usually make the home with just slush(/), then let's use the slush(/) for home.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
```

```

// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Route path='/' component={Home} />
          <Route path='/about' component={About} />
          <Route path='/contact' component={Contact} />
          <Route path='/challenges' component={Challenges} />
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Now if you try to navigate by writing / or /about you will see the home page all the time. The home route has (/) which common to other routes. Since the home is lingering let's find a way to avoid this. We can solve in three ways. One with an attribute exact. If we don't like a URL to have a trailing slash(/about/) we can use strict attribute in addition to exact.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

```

```

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Route exact path='/' component={Home} />
          <Route exact path='/about' component={About} />
          <Route exact path='/contact' component={Contact} />
          <Route exact path='/challenges' component={Challenges} />
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

If we don't like a URL to have a trailing slash, for instance(/about/), we can use strict attribute in addition to exact.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Route exact path='/' component={Home} />
          <Route exact strict path='/about' component={About} />
          <Route exact strict path='/contact' component={Contact} />
          <Route exact strict path='/challenges' component={Challenges} />
        </div>
      </Router>
    )
  }
}

```

```

}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

The other way to avoid the lingering home page is rearranging the routing order and Switch component. Just putting the home route at the bottom.

Switch

The Switch component allows only one component to be rendered.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import { BrowserRouter as Router, Route, Switch } from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Switch>
            <Route exact path='/about' component={About} />
            <Route exact path='/contact' component={Contact} />
            <Route exact path='/challenges' component={Challenges} />
            <Route exact path='/' component={Home} />
          </Switch>
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

The route is a kind of ready but so far we are navigating manually by writing each specific route. Let's make use of the NavLink component to be forwarded to each specific route.

NavLink

The NavLink component allow us to navigate each component. It takes a to required props. The NavLink is a component on top of anchor tag. Clicking on a NavLink does not do a page refresh which is one of the best quality of using a router. See the example below. First, let's implement a navigation for the home page.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <ul>
            <li>
              <NavLink to='/'>Home</NavLink>
            </li>
          </ul>

          <Switch>
            <Route path='/about' component={About} />
            <Route path='/contact' component={Contact} />
            <Route path='/challenges' component={Challenges} />
            <Route path='/' component={Home} />
          </Switch>
        </div>
      </Router>
    )
  }
}
```



```

}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Now, lets' implement navigation for all the components.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)

class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <ul>
            <li>
              <NavLink to='/'>Home</NavLink>
            </li>
            <li>
              <NavLink to='/about'>About</NavLink>
            </li>
            <li>
              <NavLink to='/contact'>Contact</NavLink>
            </li>
            <li>
              <NavLink to='/challenges'>Challenges</NavLink>
            </li>
          </ul>

          <Switch>
            <Route path='/about' component={About} />
            <Route path='/contact' component={Contact} />

```

```

        <Route path='/challenges' component={Challenges} />
        <Route path='/' component={Home} />
      </Switch>
    </div>
  </Router>
)
}
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Our route and navigation works perfectly as long as the route is found. However, if a route is not found it falls to last component. In order to avoid this problem, lets create a separate not found component and put it inside our routing.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)
const NotFound = (props) => <h1>The page your looking for not found</h1>
class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <ul>
            <li>
              <NavLink to='/'>Home</NavLink>
            </li>
            <li>
              <NavLink to='/about'>About</NavLink>
            </li>
            <li>

```

```

        <NavLink to='/contact'>Contact</NavLink>
      </li>
    </li>
    <NavLink to='/challenges'>Challenges</NavLink>
  </li>
</ul>

<Switch>
  <Route path='/about' component={About} />
  <Route path='/contact' component={Contact} />
  <Route path='/challenge' component={Challenges} />
  <Route path='/' component={Home} />
  <Route component={NotFound} />
</Switch>
</div>
</Router>
)
}
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Let's make a separate component which is responsible for the navigation.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component
const Challenges = (props) => (
  <div>
    <h1>30 Days Of React Challenge</h1>
  </div>
)
const NotFound = (props) => <h1>The page your looking for not found</h1>
const Navbar = () => (
  <ul>
    <li>
      <NavLink to='/'>Home</NavLink>
    </li>

```

```

    <li>
      <NavLink to='/about'>About</NavLink>
    </li>
    <li>
      <NavLink to='/contact'>Contact</NavLink>
    </li>
    <li>
      <NavLink to='/challenges'>Challenges</NavLink>
    </li>
  </ul>
)
class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Navbar />
          <Switch>
            <Route component={NotFound} />
            <Route path='/about' component={About} />
            <Route path='/contact' component={Contact} />
            <Route path='/challenge' component={Challenges} />
            <Route exact path='/' component={Home} />
          </Switch>
        </div>
      </Router>
    )
  }
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Nested Routing

We have implemented a simple navigation using React Router. Now, let's see how we can also nest a route. It possible to have a nested route in React.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>

```

```
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component

const challenges = [
  {
    name: '30 Days Of Python',
    description:
      '30 Days of Python challenge is a step by step guide to learn Python in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '20 Nov 2019 - 20 Dec 2019',
    slug: 'pyhton',
    url:
      'https://github.com/https://https://github.com/Asabeneh/30-Days-Of-Python.com/Asabeneh/30-Days-Of-JavaScript/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of JavaScript',
    description:
      '30 Days of JavaScript challenge is a step by step guide to learn JavaScript in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Jan 2020 - 30 Jan 2020',
    slug: 'javascript',
    url: 'https://github.com/Asabeneh/30-Days-Of-JavaScript',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of React',
    description:
      '30 Days of React challenge is a step by step guide to learn React in 30 days.',
    status: 'ongoing',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Oct 2020- 30 Oct 2020',
    slug: 'react',
    url: 'https://github.com/Asabeneh/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
]
```

```
    },
    {
      name: '30 HTML and CSS',
      description:
        '30 Days of HTML and CSS challenge is a step by step guide to learn HTML and CSS in 30 days.',
      status: 'coming',
      days: 30,
      level: 'Beginners to Advanced',
      duration: '',
      slug: 'html-and-css',
      url: '',
      author: {
        firstName: 'Asabeneh',
        lastName: 'Yetayeh',
      },
    },
    {
      name: '30 ReactNative',
      description:
        '30 Days of ReactNative challenge is a step by step guide to learn ReactNative in 30 days.',
      status: 'coming',
      days: 30,
      level: 'Beginners to Advanced',
      duration: '',
      slug: 'reactnative',
      url: '',
      author: {
        firstName: 'Asabeneh',
        lastName: 'Yetayeh',
      },
    },
    {
      name: '30 Data Analysis',
      description:
        '30 Days of Data Analysis challenge is a step by step guide to learn about data, data visualization and data analysis in 30 days.',
      status: 'coming',
      days: 30,
      level: 'Beginners to Advanced',
      duration: '',
      slug: 'data-analysis',
      url: '',
      author: {
        firstName: 'Asabeneh',
        lastName: 'Yetayeh',
      },
    },
    {
      name: '30 Machine Learning',
      description:
        '30 Days of Machine learning challenge is a step by step guide to learn
```

```

data cleaning, machine learning models and predictions in 30 days.',
  status: 'coming',
  days: 30,
  level: 'Beginners to Advanced',
  duration: '',
  slug: 'machine-learning',
  url: '',
  author: {
    firstName: 'Asabeneh',
    lastName: 'Yetayeh',
  },
},
],

const Challenge = ({
  challenge: {
    name,
    description,
    status,
    days,
    level,
    duration,
    author: { firstName, lastName },
  },
}) => (
  <div>
    <h1>{name}</h1>
    <p>{level}</p>
    <p>
      Author: {firstName} {lastName}
    </p>
    {duration && (
      <>
        {' '}
        <small>{duration}</small> <br />
      </>
    )}
    <small>Number of days: {days}</small>

    <p>{description}</p>
  </div>
)

const Challenges = (props) => {
  const path = props.location.pathname
  const slug = path.split('/').slice(path.split('/').length - 1)[0]
  const challenge = challenges.find((challenge) => challenge.slug === slug)

  return (
    <div>
      <h1>30 Days Of React Challenge</h1>
      <ul>
        {challenges.map(({ name, slug }) => (
          <li>

```

```

        <NavLink to={`/${slug}`}>{name}</NavLink>
      </li>
    )}
  </ul>
  <Switch>
    <Route
      exact
      path={`/${challenges}`}
      component={() => <h1>Choose any of the challenges</h1>}
    />
    <Route
      path={path}
      component={(props) => <Challenge challenge={challenge} />}
    />
  </Switch>
</div>
)
}

const NotFound = (props) => <h1>The page your looking for not found</h1>
const Navbar = () => (
  <ul>
    <li>
      <NavLink to={`/`}>Home</NavLink>
    </li>
    <li>
      <NavLink to={`/about`}>About</NavLink>
    </li>
    <li>
      <NavLink to={`/contact`}>Contact</NavLink>
    </li>
    <li>
      <NavLink to={`/challenges`}>Challenges</NavLink>
    </li>
  </ul>
)
class App extends Component {
  render() {
    return (
      <Router>
        <div className='App'>
          <Navbar />
          <Switch>
            <Route path={`/about`} component={About} />
            <Route path={`/contact`} component={Contact} />
            <Route path={`/challenges`} component={Challenges} />
            <Route exact path={`/`} component={Home} />
            <Route component={NotFound} />
          </Switch>
        </div>
      </Router>
    )
  }
}

```



```
const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)
```

In next part will will cover Prompt, Redirect and withRouter component.

Redirect

Redirect can help us to redirect a route to a certain path based some condition. For instance if a user is logged in we redirect it to the dashboard otherwise to the login page. Let's implement a fake login in above snippet of code. If a user logged in it will redirected to the challenges otherwise we suggest the user to login.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
  Redirect,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component

const challenges = [
  {
    name: '30 Days Of Python',
    description:
      '30 Days of Python challenge is a step by step guide to learn Python in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '20 Nov 2019 - 20 Dec 2019',
    slug: 'pyhton',
    url:
      'https://github.com/https://https://github.com/Asabeneh/30-Days-Of-Python.com/Asabeneh/30-Days-Of-JavaScript/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of JavaScript',
```

```

    description:
      '30 Days of JavaScript challenge is a step by step guide to learn JavaScript
in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Jan 2020 - 30 Jan 2020',
    slug: 'javascript',
    url: 'https://github.com/Asabeneh/30-Days-Of-JavaScript',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of React',
    description:
      '30 Days of React challenge is a step by step guide to learn React in 30
days.',
    status: 'ongoing',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Oct 2020- 30 Oct 2020',
    slug: 'react',
    url: 'https://github.com/Asabeneh/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 HTML and CSS',
    description:
      '30 Days of HTML and CSS challenge is a step by step guide to learn HTML and
CSS in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'html-and-css',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 ReactNative',
    description:
      '30 Days of ReactNative challenge is a step by step guide to learn
ReactNative in 30 days.',
    status: 'coming',
    days: 30,

```

```

    level: 'Beginners to Advanced',
    duration: '',
    slug: 'reactnative',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Data Analysis',
    description:
      '30 Days of Data Analysis challenge is a step by step guide to learn about
data, data visualization and data analysis in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'data-analysis',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Machine Learning',
    description:
      '30 Days of Machine learning challenge is a step by step guide to learn
data cleaning, machine learning models and predictions in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'machine-learning',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
]

const Challenge = ({
  challenge: {
    name,
    description,
    status,
    days,
    level,
    duration,
    author: { firstName, lastName },
  },
}) => (

```

```

    <div>
      <h1>{name}</h1>
      <p>{level}</p>
      <p>
        Author: {firstName} {lastName}
      </p>
      {duration && (
        <>
          { ' ' }
          <small>{duration}</small> <br />
        </>
      )}
      <small>Number of days: {days}</small>

      <p>{description}</p>
    </div>
  )

const Challenges = (props) => {
  const path = props.location.pathname
  const slug = path.split('/').slice(path.split('/').length - 1)[0]
  const challenge = challenges.find((challenge) => challenge.slug === slug)

  return (
    <div>
      <h1>30 Days Of React Challenge</h1>
      <ul>
        {challenges.map(({ name, slug }) => (
          <li>
            <NavLink to={`/challenges/${slug}`}>{name}</NavLink>
          </li>
        ))}
      </ul>
      <Switch>
        <Route
          exact
          path={'/challenges'}
          component={() => <h1>Choose any of the challenges</h1>}
        />
        <Route
          path={path}
          component={(props) => <Challenge challenge={challenge} />}
        />
      </Switch>
    </div>
  )
}

const NotFound = (props) => <h1>The page your looking for not found</h1>
const Navbar = ({ username }) => (
  <ul>
    <li>
      <NavLink to="/">Home</NavLink>
    </li>
  </ul>
)

```

```

    <li>
      <NavLink to='/about'>About</NavLink>
    </li>
    <li>
      <NavLink to='/contact'>Contact</NavLink>
    </li>
    <li>
      <NavLink to={` /user/${username}`}>User</NavLink>
    </li>
    <li>
      <NavLink to='/challenges'>Challenges</NavLink>
    </li>
  </ul>
)

const User = ({ match, isLoggedIn, handleLogin }) => {
  const username = match.params.username
  return (
    <div>
      {isLoggedIn ? (
        <>
          <h1>Welcome {username} to the challenge</h1>
          <small>Now, you can navigate through all the challenges</small> <br />
        </>
      ) : (
        <p>Please login in to access the challenges </p>
      )}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

const Welcome = ({ handleLogin, isLoggedIn }) => {
  return (
    <div>
      {isLoggedIn ? 'Welcome to the challenge' : <p>Please login in </p>}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

class App extends Component {
  state = {
    isLoggedIn: false,
    firstName: 'Asabeneh',
  }
  handleLogin = () => {
    this.setState({
      isLoggedIn: !this.state.isLoggedIn,
    })
  }
  render() {
    return (
      <Router>
        <div className='App'>

```

```

    <Navbar username={this.state.firstName} />
    <Switch>
      <Route path='/about' component={About} />
      <Route path='/contact' component={Contact} />
      <Route
        path='/user/:username'
        component={(props) => (
          <User
            {...props}
            isLoggedIn={this.state.isLoggedIn}
            handleLogin={this.handleLogin}
          />
        )}
      />
      <Route
        path='/login'
        component={(props) => (
          <Welcome
            {...props}
            isLoggedIn={this.state.isLoggedIn}
            handleLogin={this.handleLogin}
          />
        )}
      />
      <Route
        path='/challenges'
        component={(props) => {
          return this.state.isLoggedIn ? (
            <Challenges {...props} />
          ) : (
            <Redirect to='/user/asabeneh' />
          )
        }}
      />
      <Route exact path='/' component={Home} />
      <Route component={NotFound} />
    </Switch>
  </div>
</Router>
)
}
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Prompt

Sometimes when a user try to leave a page we may like to inform that he has unfinished task. In order to do that we can use the Prompt component. The Prompt component takes two props which are when and message(<Prompt when = {true ? 'Happy':'Sad'} message = 'When even I am happy' />). Let's implement this in the previous code.

In the following code a Prompt has been implemented without when therefore it will check all the routes.

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
  Redirect,
  Prompt,
} from 'react-router-dom'

// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component

const challenges = [
  {
    name: '30 Days Of Python',
    description:
      '30 Days of Python challenge is a step by step guide to learn Python in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '20 Nov 2019 - 20 Dec 2019',
    slug: 'pyhton',
    url:
      'https://github.com/https://https://github.com/Asabeneh/30-Days-Of-Python.com/Asabeneh/30-Days-Of-JavaScript/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of JavaScript',
    description:
      '30 Days of JavaScript challenge is a step by step guide to learn JavaScript in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Jan 2020 - 30 Jan 2020',
    slug: 'javascript',
    url: 'https://github.com/Asabeneh/30-Days-Of-JavaScript',
    author: {
      firstName: 'Asabeneh',
```

```
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of React',
    description:
      '30 Days of React challenge is a step by step guide to learn React in 30 days.',
    status: 'ongoing',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Oct 2020- 30 Oct 2020',
    slug: 'react',
    url: 'https://github.com/Asabeneh/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 HTML and CSS',
    description:
      '30 Days of HTML and CSS challenge is a step by step guide to learn HTML and CSS in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'html-and-css',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 ReactNative',
    description:
      '30 Days of ReactNative challenge is a step by step guide to learn ReactNative in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'reactnative',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Data Analysis',
```



```

    description:
      '30 Days of Data Analysis challenge is a step by step guide to learn about
data, data visualization and data analysis in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'data-analysis',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Machine Learning',
    description:
      '30 Days of Machine learning challenge is a step by step guide to learn
data cleaning, machine learning models and predictions in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'machine-learning',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
]

```

```

const Challenge = ({
  challenge: {
    name,
    description,
    status,
    days,
    level,
    duration,
    author: { firstName, lastName },
  },
}) => (
  <div>
    <h1>{name}</h1>
    <p>{level}</p>
    <p>
      Author: {firstName} {lastName}
    </p>
    {duration && (
      <>
        {' '}
        <small>{duration}</small> <br />
      </>
    )}
  </div>
)

```

```

    })
    <small>Number of days: {days}</small>

    <p>{description}</p>
  </div>
)

const Challenges = (props) => {
  const path = props.location.pathname
  const slug = path.split('/').slice(path.split('/').length - 1)[0]
  const challenge = challenges.find((challenge) => challenge.slug === slug)

  return (
    <div>
      <h1>30 Days Of React Challenge</h1>
      <ul>
        {challenges.map(({ name, slug }) => (
          <li>
            <NavLink to={`/${challenges}/${slug}`}>{name}</NavLink>
          </li>
        ))}
      </ul>
      <Switch>
        <Route
          exact
          path={`/${challenges}`}
          component={() => <h1>Choose any of the challenges</h1>}
        />
        <Route
          path={path}
          component={(props) => <Challenge challenge={challenge} />}
        />
      </Switch>
    </div>
  )
}

const NotFound = (props) => <h1>The page your looking for not found</h1>
const Navbar = ({ username }) => (
  <ul>
    <li>
      <NavLink to="/">Home</NavLink>
    </li>
    <li>
      <NavLink to="/about">About</NavLink>
    </li>
    <li>
      <NavLink to="/contact">Contact</NavLink>
    </li>
    <li>
      <NavLink to={`/${username}`}>User</NavLink>
    </li>
    <li>
      <NavLink to="/challenges">Challenges</NavLink>

```

```

    </li>
  </ul>
)

const User = ({ match, isLoggedIn, handleLogin }) => {
  const username = match.params.username
  return (
    <div>
      {isLoggedIn ? (
        <>
          <h1>Welcome {username} to the challenge</h1>
          <small>Now, you can navigate through all the challenges</small> <br />
        </>
      ) : (
        <p>Please login in to access the challenges </p>
      )}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

const Welcome = ({ handleLogin, isLoggedIn }) => {
  return (
    <div>
      {isLoggedIn ? 'Welcome to the challenge' : <p>Please login in </p>}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

class App extends Component {
  state = {
    isLoggedIn: false,
    firstName: 'Asabeneh',
  }
  handleLogin = () => {
    this.setState({
      isLoggedIn: !this.state.isLoggedIn,
    })
  }
  render() {
    return (
      <Router>
        <div className='App'>
          <Navbar username={this.state.firstName} />
          <Prompt message='Are you sure you want to leave?' />

          <Switch>
            <Route path='/about' component={About} />
            <Route path='/contact' component={Contact} />
            <Route
              path='/user/:username'
              component={(props) => (
                <User
                  {...props}

```

```

        isLoggedIn={this.state.isLoggedIn}
        handleLogin={this.handleLogin}
      />
    )}
  />
  <Route
    path='/login'
    component={(props) => (
      <Welcome
        {...props}
        isLoggedIn={this.state.isLoggedIn}
        handleLogin={this.handleLogin}
      />
    )}
  />
  <Route
    path='/challenges'
    component={(props) => {
      return this.state.isLoggedIn ? (
        <Challenges {...props} />
      ) : (
        <Redirect to='/user/asabeneh' />
      )
    }}
  />
  <Route exact path='/' component={Home} />
  <Route component={NotFound} />
</Switch>
</div>
</Router>
)
}
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Instead of without condition, let's inform the user if he really wants to log out by adding checking some condition using a call back function inside the message.

```

import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import {
  BrowserRouter as Router,
  Route,
  Switch,
  NavLink,
  Redirect,
  Prompt,
} from 'react-router-dom'

```

```
// Home component
const Home = (props) => <h1>Welcome Home</h1>
// About component
const About = (props) => <h1>About Us</h1>
// Contact component
const Contact = (props) => <h1>Contact us</h1>
// Challenge component

const challenges = [
  {
    name: '30 Days Of Python',
    description:
      '30 Days of Python challenge is a step by step guide to learn Python in 30
days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '20 Nov 2019 - 20 Dec 2019',
    slug: 'pyhton',
    url:
      'https://github.com/https://https://github.com/Asabeneh/30-Days-Of-
Python.com/Asabeneh/30-Days-Of-JavaScript/30-Days-Of-React',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of JavaScript',
    description:
      '30 Days of JavaScript challenge is a step by step guide to learn JavaScript
in 30 days.',
    status: 'completed',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Jan 2020 - 30 Jan 2020',
    slug: 'javascript',
    url: 'https://github.com/Asabeneh/30-Days-Of-JavaScript',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Days Of React',
    description:
      '30 Days of React challenge is a step by step guide to learn React in 30
days.',
    status: 'ongoing',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '1 Oct 2020- 30 Oct 2020',
    slug: 'react',
    url: 'https://github.com/Asabeneh/30-Days-Of-React',
  },
]
```

```
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 HTML and CSS',
    description:
      '30 Days of HTML and CSS challenge is a step by step guide to learn HTML and CSS in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'html-and-css',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 ReactNative',
    description:
      '30 Days of ReactNative challenge is a step by step guide to learn ReactNative in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'reactnative',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
  {
    name: '30 Data Analysis',
    description:
      '30 Days of Data Analysis challenge is a step by step guide to learn about data, data visualization and data analysis in 30 days.',
    status: 'coming',
    days: 30,
    level: 'Beginners to Advanced',
    duration: '',
    slug: 'data-analysis',
    url: '',
    author: {
      firstName: 'Asabeneh',
      lastName: 'Yetayeh',
    },
  },
},
```

```

{
  name: '30 Machine Learning',
  description:
    '30 Days of Machine learning challenge is a step by step guide to learn
    data cleaning, machine learning models and predictions in 30 days.',
  status: 'coming',
  days: 30,
  level: 'Beginners to Advanced',
  duration: '',
  slug: 'machine-learning',
  url: '',
  author: {
    firstName: 'Asabeneh',
    lastName: 'Yetayeh',
  },
},
],

```

```

const Challenge = ({
  challenge: {
    name,
    description,
    status,
    days,
    level,
    duration,
    author: { firstName, lastName },
  },
}) => (
  <div>
    <h1>{name}</h1>
    <p>{level}</p>
    <p>
      Author: {firstName} {lastName}
    </p>
    {duration && (
      <>
        {' '}
        <small>{duration}</small> <br />
      </>
    )}
    <small>Number of days: {days}</small>

    <p>{description}</p>
  </div>
)

```

```

const Challenges = (props) => {
  const path = props.location.pathname
  const slug = path.split('/').slice(path.split('/').length - 1)[0]
  const challenge = challenges.find((challenge) => challenge.slug === slug)

  return (
    <div>

```

```

    <h1>30 Days Of React Challenge</h1>
    <ul>
      {challenges.map(({ name, slug }) => (
        <li>
          <NavLink to={` /challenges/${slug}`}>{name}</NavLink>
        </li>
      ))}
    </ul>
    <Switch>
      <Route
        exact
        path={` /challenges`}
        component={() => <h1>Choose any of the challenges</h1>}
      />
      <Route
        path={path}
        component={(props) => <Challenge challenge={challenge} />}
      />
    </Switch>
  </div>
)
}

const NotFound = (props) => <h1>The page your looking for not found</h1>
const Navbar = ({ username }) => (
  <ul>
    <li>
      <NavLink to="/">Home</NavLink>
    </li>
    <li>
      <NavLink to="/about">About</NavLink>
    </li>
    <li>
      <NavLink to="/contact">Contact</NavLink>
    </li>
    <li>
      <NavLink to={` /user/${username}`}>User</NavLink>
    </li>
    <li>
      <NavLink to="/challenges">Challenges</NavLink>
    </li>
  </ul>
)

const User = ({ match, isLoggedIn, handleLogin }) => {
  const username = match.params.username
  return (
    <div>
      {isLoggedIn ? (
        <>
          <h1>Welcome {username} to the challenge</h1>
          <small>Now, you can navigate through all the challenges</small> <br />
        </>
      ) : (

```



```

        <p>Please login in to access the challenges </p>
      )}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

const Welcome = ({ handleLogin, isLoggedIn }) => {
  return (
    <div>
      {isLoggedIn ? 'Welcome to the challenge' : <p>Please login in </p>}
      <button onClick={handleLogin}>{isLoggedIn ? 'Logout' : 'Login'}</button>
    </div>
  )
}

class App extends Component {
  state = {
    isLoggedIn: false,
    firstName: 'Asabeneh',
  }
  handleLogin = () => {
    this.setState({
      isLoggedIn: !this.state.isLoggedIn,
    })
  }
  render() {
    return (
      <Router>
        <div className='App'>
          <Navbar username={this.state.firstName} />

          <Prompt
            message={({ pathname }) => {
              return this.state.isLoggedIn &&
                pathname.includes('/user/Asabeneh')
                ? 'Are you sure you want to logout?'
                : true
            }}
          />

          <Switch>
            <Route path='/about' component={About} />
            <Route path='/contact' component={Contact} />
            <Route
              path='/user/:username'
              component={(props) => (
                <User
                  {...props}
                  isLoggedIn={this.state.isLoggedIn}
                  handleLogin={this.handleLogin}
                />
              )}
            />
            <Route

```

```

        path='/login'
        component={ (props) => (
          <Welcome
            {...props}
            isLoggedIn={this.state.isLoggedIn}
            handleLogin={this.handleLogin}
          />
        )}
      />
    <Route
      path='/challenges'
      component={ (props) => {
        return this.state.isLoggedIn ? (
          <Challenges {...props} />
        ) : (
          <Redirect to='/user/asabeneh' />
        )
      }}
    />
    <Route exact path='/' component={Home} />
    <Route component={NotFound} />
  </Switch>
</div>
</Router>
)
}
}

const rootElement = document.getElementById('root')
ReactDOM.render(<App />, rootElement)

```

Exercises

Exercises: Level 1

1. What package do you use to implement routing in react?
2. What is the default export in react-router-dom?
3. What is the use of the following Components(Route, NavLink, Switch, Redirect, Prompt)

Exercises: Level 2

Now, you know about React router. Build your portfolio with React and implement React router for navigation.

Exercises: Level 3

coming



CONGRATULATIONS ! 🎉

[<< Day 16](#) | [Day 18 >>](#)