



Introduction to ReactJS

A Student-Run Short Course (SRC) conducted by the Student Academic Council in collaboration with the Technical Council.

Instructors:

- Reuben Devanesan
- Ananthu JP
- Kanishk Singhal

Logistics

- ❑ 16th Oct to 4th Nov, 10 PM to 11 PM. AB 1/101, Learning Theatre.
- ❑ All important updates will be communicated through Google Classroom.
- ❑ All codes will be uploaded to the GitHub repository. Link: <https://github.com/Reuben27/ReactJS-SRC>
- ❑ Part I: <https://github.com/Reuben27/Web-Development-SRC>
- ❑ P/F course. Attendance compulsory.





React Router

- ❑ Routing is the process of redirecting a user to different pages based on their action or request (as specified by a URL and HTTP method).
- ❑ In React, for routing you need to use an external library called React Router.
- ❑ **URL:** The URL in the address bar.
- ❑ **Location:** This is a React Router specific object that is based on the built-in browser's `window.location` object. It represents "where the user is at". It's mostly an object representation of the URL but has a bit more to it than that.
- ❑ **Segment:** The parts of a URL or path pattern between the `/` characters. For example, `"/users/123"` has two segments.
- ❑ **Path Pattern:** These look like URLs but can have special characters for matching URLs to routes, like dynamic segments (`"/users/:userId"`) or star segments (`"/docs/*"`). They aren't URLs, they're patterns that React Router will match.



React Router

- ❑ **Dynamic Segment:** A segment of a path pattern that is dynamic, meaning it can match any values in the segment. For example the pattern `/users/:userId` will match URLs like `/users/123`
- ❑ **URL Params:** The parsed values from the URL that matched a dynamic segment.
- ❑ **Router:** Stateful, top-level component that makes all the other components and hooks work.
- ❑ **Route:** An object or Route Element typically with a shape of `{ path, element }` or `<Route path element>`. The path is a path pattern. When the path pattern matches the current URL, the element will be rendered.
- ❑ **Route Element:** Or `<Route>`, This element's props are read to create a route by `<Routes>`, but otherwise does nothing.
- ❑ **History Stack:** As the user navigates, the browser keeps track of each location in a stack. If you click and hold the back button in a browser you can see the browser's history stack right there.



Installation & Setup

```
// index.js
import React from 'react';
import ReactDOM from 'react-dom/client';
import { BrowserRouter } from 'react-router-dom';
import App from './App';

const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <BrowserRouter>
      <App />
    </BrowserRouter>
  </React.StrictMode>
);
```

```
npm install react-router-dom@6
```

```
const App = () => {
  return (
    <>
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/products" element={<Products />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </>
  );
};

export default App;
```



Router, Routes, Route

- ❑ **<BrowserRouter>**: BrowserRouter is a parent component in react-router-dom that stores all the other route components. Allowing the declaration of individual routes is the main functionality of using BrowserRouter in the application.
- ❑ **<Routes>**: Routes is a new component introduced in v6 that replaces the switch component. (Switch renders a route exclusively as it displays the first child route that matches the current URL).
- ❑ **<Route>**: Route is the child component that renders a specific UI component when the URL matches the specified path. The path attribute specifies the path name we assign to the component and the element attribute refers to the component to render when the URL matches.



Dynamic Segments, Params

- ❑ If a path segment starts with : then it becomes a "**dynamic segment**". When the route matches the URL, the dynamic segment will be parsed from the URL and provided as params to other router APIs.
- ❑ The **useParams** hook returns an object of key/value pairs of the dynamic params from the current URL that were matched by the <Route path>. Child routes inherit all params from their parent routes.



Thank you!