

React 19 Routers

React 19 Routers (React Router v6+ Overview)

In **React 19**, routing is handled using **React Router v6+**, a powerful library that enables navigation between components and pages. It supports:

- Declarative routing using `<Route />`
 - Nested routes
 - Dynamic route parameters (`/user/:id`)
 - Redirects using `useNavigate`
 - Access to route info via hooks like `useParams`, `useLocation`, and `useNavigate`
-

Feature	Description
<code>BrowserRouter</code>	Top-level component that wraps all routes using browser history
<code>Routes</code>	Container for all <code><Route></code> definitions
<code>Route path="..."</code>	Declares a route and which component it renders
<code>useNavigate()</code>	Hook to programmatically redirect to another route
<code>useParams()</code>	Hook to read dynamic URL segments (e.g., <code>/user/:id</code>)
<code>useLocation()</code>	Gives access to location object (path, search, state, etc.)

Types of Routes

- **Static route:**

```
<Link to="/signup">Sign Up</Link>
```

- **Dynamic route:**

```
<Link to={` /user/${user.id}`}>{user.name}</Link>
```

- **Passing state (optional):**

```
<Link to="/profile" state={{ from: "dashboard" }}>Profile</Link>
```

You can then access that state with `useLocation()` in the destination component.

React Routers Application showcasing dynamic route parameters

Install the following:

```
npm install bootstrap
```

```
npm install react-router-dom
```

-----App.js-----

```
import React from "react";
import { BrowserRouter as Router, Routes, Route } from "react-router-dom";
import UserList from "../components/UserList";
import UserDetails from "../components/UserDetails";

const App = () => {
  return (
    <Router>
      <div className="container mt-4">
        <Routes>
          <Route path="/" element={ <UserList /> } />
          <Route path="/user/:id" element={ <UserDetails /> } />
        </Routes>
      </div>
    </Router>
  );
};

export default App;
```

-----UserList.js-----

```
import React from "react";
import { Link } from "react-router-dom";

const users = [
  { id: "1", name: "Alice" },
  { id: "2", name: "Bob" },
  { id: "3", name: "Charlie" },
];

const UserList = () => {
  return (
    <div>
      <h2>User List</h2>
      <ul className="list-group">
        {users.map((user) => (
          <li key={user.id} className="list-group-item">
            <Link to={`/user/${user.id}`}>{user.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
};

export default UserList;
```

Note:

`<Link>` is a **React component** provided by **React Router DOM** allows you to **navigate between routes without reloading the page**.

It's like an HTML `<a>` tag, **but better** for Single Page Applications (SPA), because it avoids full-page reloads and keeps the app fast and smooth.

Use `<Link>` instead of `<a>`

<code></code>	<code><Link to="/home"></code>
Causes full page reload	Keeps SPA behavior, no reload
Breaks React state/context	Preserves app state
Not suitable for SPA	Designed for React Router apps

-----UserDetails.js-----

```
import React from "react";
import { useParams } from "react-router-dom";

const UserDetails = () => {
  const { id } = useParams(); // Extract :id from route

  return (
    <div>
      <h2>User Details</h2>
      <p>User ID from URL: <strong>{id}</strong></p>
    </div>
  );
};

export default UserDetails;
```

Note:

- `UserList` shows a list of users with links like `/user/1`, `/user/2`, etc.
- When a link is clicked, the router matches the path `/user/:id`.
- `UserDetails` extracts the ID via `useParams()` and renders details for that user.

Another Application showcasing React Routers

Login and Signup page

Install the following:

```
npm install bootstrap
npm install react-router-dom
```

-----App.js-----

```
import React from "react";
import { BrowserRouter as Router, Routes, Route, } from "react-router-dom";
import Login from "../components/Login";
import Signup from "../components/Signup";
import 'bootstrap/dist/css/bootstrap.min.css';
import { useNavigate, useLocation } from "react-router-dom";
import { useEffect } from "react";
```

```
const RedirectToLogin = () => {
  const navigate = useNavigate();
  const location = useLocation();
```

```
  useEffect(() => {
    navigate("/login", { replace: true, state: { from: location } });
  }, [navigate, location]);
```

```
  return null;
};
```

```
const App = () => {
  return (
    <Router>
      <div className="container mt-5">
        <Routes>
          <Route path="/login" element={<Login />} />
          <Route path="/signup" element={<Signup />} />
          <Route path="*" element={<RedirectToLogin/>} />
        </Routes>
      </div>
    </Router>
  );
};
```

```
export default App;
```

-----Login.js-----

```
import React, { useState } from "react";

const Login = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    alert(`Logged in as ${email}`);
    setEmail("");
    setPassword("");
  };

  return (
    <div className="card p-4">
      <h2 className="mb-3">Login</h2>
      <form onSubmit={handleSubmit}>
        <div className="mb-3">
          <label htmlFor="email" className="form-label">Email address</label>
          <input type="email" className="form-control" id="email" value={email} onChange={(e) =>
setEmail(e.target.value)} required />
        </div>
        <div className="mb-3">
          <label htmlFor="password" className="form-label">Password</label>
          <input type="password" className="form-control" id="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
        </div>
        <button type="submit" className="btn btn-primary">Login</button>
      </form>
    </div>
  );
};

export default Login;
```

-----Signup.js-----

```
import React, { useState } from "react";

const Signup = () => {
  const [email, setEmail] = useState("");
  const [password, setPassword] = useState("");
  const [confirmPassword, setConfirmPassword] = useState("");

  const handleSubmit = (e) => {
    e.preventDefault();
    if (password !== confirmPassword) {
      alert("Passwords do not match!");
      return;
    }
    alert(`Signed up with email: ${email}`);
    setEmail("");
    setPassword("");
    setConfirmPassword("");
  };

  return (
    <div className="card p-4">
      <h2 className="mb-3">Sign Up</h2>
      <form onSubmit={handleSubmit}>
        <div className="mb-3">
          <label htmlFor="email" className="form-label">Email address</label>
          <input type="email" className="form-control" id="email" value={email} onChange={(e) =>
setEmail(e.target.value)} required />
        </div>
        <div className="mb-3">
          <label htmlFor="password" className="form-label">Password</label>
          <input type="password" className="form-control" id="password" value={password} onChange={(e) =>
setPassword(e.target.value)} required />
        </div>
        <div className="mb-3">
          <label htmlFor="confirmPassword" className="form-label">Confirm Password</label>
          <input type="password" className="form-control" id="confirmPassword" value={confirmPassword}
onChange={(e) => setConfirmPassword(e.target.value)} required />
        </div>
        <button type="submit" className="btn btn-success">Sign Up</button>
      </form>
    </div>
  );
};

export default Signup;
```
