# Routing

## Terms

Index routes
Layout routes
Route
Route parameters
Query string parameters

## Summary

- React Router is a popular routing library for React that allows us to handle navigation and rendering of different components based on the URL.

- To create a router, we use the **createBrowserRouter()** function and provide an array of route objects.

- Each route has two properties: **path** and **element**, where path is a string that defines the URL pattern and element is the component to be rendered when the path matches the current URL.

- Route parameters are defined using a colon (e.g., /users/:userId) in the path of a route object, allowing us to capture dynamic values from the URL and pass them to our components.

- The **useParams()** hook is used to retrieve the route parameters in the current route.

- The **useSearchParams()** hook is used to retrieve and update the query string parameters of the current URL.

- The **<Link>** component is used to create links between different routes.

- The **<NavLink>** component works like the **<Link>** component, but it adds an active class to the current link by default. The name of this class can be customized by passing a function to the **className** prop.

- The **useNavigate()** hook is used to redirect the user programmatically.

- The **errorElement** property of the root route can be used to specify a component to render when a route is not found or an error occurs during rendering.

- The **<Outlet>** component is used as a placeholder to render child component when working with nested routes.

- Layout routes are useful to enforce a common layout or business rules across a group of routes. A layout route is a route without a path.

- An index route represents the default component to render in the outlet of a parent route when the current URL matches the path of the parent route.

## SETTING UP ROUTES

```
const router = createBrowserRouter([
  { path: '/', element: <HomePage /> },
  { path: '/users', element: <UserListPage /> },
]);
```

```
<React.StrictMode>
  <RouterProvider router={router} />
</React.StrictMode>
```

## NAVIGATION

```
<Link to='/users'>Users</Link>
```

```
// Adds the "active" class if the route
// matches the current URL.
<NavLink to='/users'>Users</NavLink>
```

## PROGRAMMATIC NAVIGATION

```
const navigate = useNavigate();
```

```
const handleSubmit = () => {
  navigate('/');
}
```

## ROUTE PARAMETERS

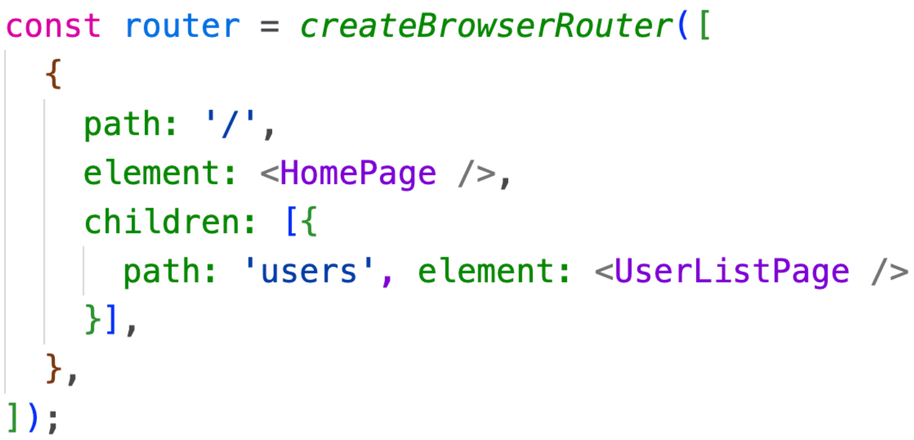
```
const router = createBrowserRouter([
  { path: '/', element: <HomePage /> },
  { path: '/users', element: <UserListPage /> },
  { path: '/users/:id', element: <UserDetailPage /> },
]);
```

```
// Get route parameters
const { id } = useParams();

// Get/update query string parameters
const [searchParams, setSearchParams] = useSearchParams();

// Get the current location
const location = useLocation();
```

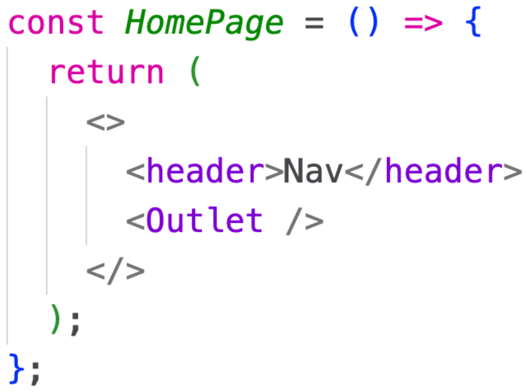
## HANDLING ERRORS

```
const router = createBrowserRouter([
  {
    path: '/',
    element: <HomePage />,
    errorElement: <ErrorPage />,
  },
]);
```

## NESTED ROUTES

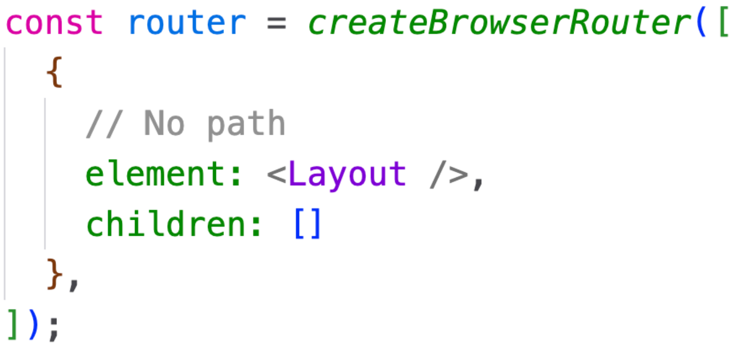


```
const router = createBrowserRouter([
  {
    path: '/',
    element: <HomePage />,
    children: [{
      path: 'users', element: <UserListPage />
    }],
  },
]);
```



```
const HomePage = () => {
  return (
    <>
      <header>Nav</header>
      <Outlet />
    </>
  );
};
```

## LAYOUT ROUTES



```
const router = createBrowserRouter([
  {
    // No path
    element: <Layout />,
    children: []
  },
]);
```