



# Front-End Deployment

VSCHOOL

FSJS

# Objectives



Understand the diff



Learn the common syntax  
for routing in React



Implement basic routing  
functionality in a React  
application

# Benefits of using React Router DOM

- ▶ Improved user experience with smooth transitions between pages
- ▶ Better SEO by managing URL structure
- ▶ Easier application maintenance and code organization
- ▶ Enhanced accessibility for screen readers and assistive technologies.

# Routes

- **BrowserRouter:**

- Sets up client-side routing infrastructure.
- Uses HTML5 History API.
- Should be the root component for routing.

- **Routes:**

- Container for defining route configurations.
- Encloses multiple `Route` components.
- Promotes organized and centralized route definitions.

- **Route:**

- Associates a specific path with a React component.
- Uses `path` prop for URL pattern.
- Uses `element` prop for the corresponding component.

# Routes

```
<BrowserRouter>

  <Navbar />
  <Routes>

    <Route path="/" element={<Home />} />
    <Route path="/products" element={<Products />} />
    <Route path="/products/:productId" element={<ProductDetails />} />
  </Routes>

</BrowserRouter>;
```

# Browser Router as Router

- ▶ Index.js or main.js
- ▶ Common practice to wrap `<App />` in Router tags

```
import {BrowserRouter as Router} from 'react-router-dom'  
  
<Router>  
|   <App />  
</Router>
```

# Routes and Route

```
<Routes>  
  <Route />  
  <Route />  
</Routes>
```

# Route

```
<Route path="/" element={<Home />} />
```

```
<Route path="/profile" element={<Profile />} />
```



## Link

- ▶ Facilitates declarative navigation in React Router DOM.
- ▶ Utilizes the `to` prop to specify the target URL or path.
- ▶ Prevents full page reloads, contributing to a smoother user experience.
- ▶ Integrates seamlessly with other React Router components for dynamic routing.
- ▶ Used within components to create clickable links for navigation.

# Link

```
<Link to="/">  
|   Home  
</Link >
```

```
<Link to="/profile">  
|   Profile  
</Link >
```

# The Relationship between Route and Link

- ▶ Link components are used within your application to create navigation links.
- ▶ Corresponding **<Route>** components define what content to render based on the URL path.

# Navigate

- ▶ **useNavigate** is a React Router hook designed for programmatic navigation in React applications.
- ▶ Enables seamless transitions between different views without triggering a full page reload, providing a smoother user experience.
- ▶ For instance, in a button click event, invoking **navigate("/products")** can redirect the user to the "/products" route, updating the UI accordingly.

# Navigate

```
import { useNavigate } from 'react-router-dom'

const navigate = useNavigate()

✓ function handleClick() {
  |   -----
  |       navigate('/products')
  |
  }
}
```

# Params

- ▶ The **useParams** hook in React Router is a utility that allows developers to access and extract parameters from the URL in a React component.
- ▶ Allows for dynamic components based off route parameters.
- ▶ Example -- Details page of a product on an e-commerce site.

# Params

```
<Route path='/products/:productid' element={<ProductDetails />} />  
.....  
  
function productDetails() {  
  const { productid } = useParams()  
  
  return (  
    <div>Product ID: {productid}</div>  
  )  
}
```