Hema Coding School

YouTube Link: https://www.youtube.com/@HemaCodingSchool

React Router DOM is a library used in React applications to handle client-side routing. It enables the creation of Single Page Applications (SPAs) where navigation does not cause a full page reload, improving the user experience.

Key Concepts

1. **Routing**: Refers to the ability to map URLs to specific components in the application.
2. **Single Page Application (SPA)**: An application that loads a single HTML page and dynamically updates content as the user interacts with the app.
3. **Declarative Routing**: Routes are defined using JSX, making the code easy to understand and maintain.

Installation

To install React Router DOM, use the following npm command:

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

Main Components

1. **BrowserRouter**
   - The primary router component for web applications.
   - Uses the HTML5 history API to keep the UI in sync with the URL.
   - Wrapped parent component inside BrowserRouter

```
import { BrowserRouter } from 'react-router-dom';
  <BrowserRouter>
    <App />
   </BrowserRouter>
```

Routes and Route

- Routes component defines the routing rules.
- Route component maps a URL path to a specific component

```
import { Routes, Route } from 'react-router-dom';

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

Link

- Link: Used for navigation without reloading the page.

```
import { Link } from 'react-router-dom';

function Navigation() {
  return (
    <nav>
      <Link to="/">Home</Link>
    </nav>
  );
}
```

Outlet

- A component that renders the matching child route.
- Useful for nested routing structures.

```
import { Outlet } from 'react-router-dom';

function Layout() {
  return (
    <div>
      <h1>My Website</h1>
      <Outlet />
    </div>
  );
}
```

Example Usage

A simple example to demonstrate the usage of React Router DOM:

```
import React from 'react';
import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';
import Home from './Home';
import About from './About';
import './App.css';

function App() {
  return (
    <Router>
      <div>
        <nav>
          <Link to="/">Home</Link>
          <Link to="/about">About</Link>
        </nav>
```

```
      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
      </Routes>
    </div>
  </Router>
 );
}

export default App;
```

Interview Questions:

BrowserRouter

Question 1: What is BrowserRouter in React Router DOM?

Answer: BrowserRouter is a component in React Router DOM that uses the HTML5 history API (pushState, replaceState, and the popstate event) to keep your UI in sync with the URL. It is the primary router component for web applications.

Question 2: How does BrowserRouter differ from HashRouter?

Answer: BrowserRouter uses the HTML5 history API to manipulate the URL, providing clean URLs without the hash symbol (#). HashRouter, on the other hand, uses the URL hash (#) to keep track of the routing, which can be useful for older browsers or environments where you cannot configure the server to handle the URL paths correctly.

HashRouter Example:

```
import React from 'react';
import { HashRouter as Router, Routes, Route, Link } from 'react-router-dom';
import Home from './Home';
import About from './About';
import Contact from './Contact';
import './App.css';

function App() {
 return (
  <Router>
   <div>
    <nav className="navbar">
     <Link to="/" className="nav-link">Home</Link>
     <Link to="/about" className="nav-link">About</Link>
     <Link to="/contact" className="nav-link">Contact</Link>
```

```
      </nav>

      <Routes>
        <Route path="/" element={<Home />} />
        <Route path="/about" element={<About />} />
        <Route path="/contact" element={<Contact />} />
      </Routes>
    </div>
  </Router>
 );
}

export default App;
```

Routes and Route

Question 3: What is the purpose of the Routes component in React Router DOM?

Answer: The Routes component is used to define a set of routes in a React application. It replaces the older Switch component and ensures that only the first matching Route element is rendered.

Question 4: How do you use the Route component to define a route?

Answer: The Route component is used to define a single route in your application. You specify the path prop to match the URL and the element prop to specify the component to render.

```
import { BrowserRouter as Router, Routes, Route } from 'react-router-dom';
import Home from './Home';
import About from './About';

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

Link

Question 5: What is the Link component used for in React Router DOM?

Answer: The Link component is used to create navigation links in your application. It allows you to navigate between different routes without causing a full page reload.

```
import { Link } from 'react-router-dom';

function Navigation() {
  return (
    <nav>
      <Link to="/">Home</Link>
      <Link to="/about">About</Link>
    </nav>
  );
}
```

Question 6: How does NavLink differ from Link?

Answer: NavLink is a special type of Link that allows you to apply styling to the link when it is active. It accepts an activeClassName or activeStyle prop to style the active link.

```
import { NavLink } from 'react-router-dom';

function Navigation() {
  return (
    <nav>
      <NavLink to="/" activeClassName="active">Home</NavLink>
      <NavLink to="/about" activeClassName="active">About</NavLink>
    </nav>
  );
}
```

Outlet

Question 7: What is the Outlet component used for in React Router DOM?

Answer: The Outlet component is used to render the matched child route components in nested routing scenarios. It acts as a placeholder in the parent component where the child routes will be rendered.

```
import { BrowserRouter as Router, Routes, Route, Outlet } from 'react-router-dom';

function Layout() {
  return (
    <div>
      <h1>My Website</h1>
      <Outlet />
    </div>
  );
```

```
}

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route path="home" element={<Home />} />
          <Route path="about" element={<About />} />
        </Route>
      </Routes>
    </Router>
  );
}
```

Question 8: How do you set up nested routes using the Outlet component?

Answer: To set up nested routes, you define a parent route with an element that includes the Outlet component. Then, define child routes as nested routes within the parent route.

```
import { BrowserRouter as Router, Routes, Route, Outlet } from 'react-router-dom';
import Home from './Home';
import About from './About';

function Layout() {
  return (
    <div>
      <h1>My Website</h1>
      <Outlet />
    </div>
  );
}

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Layout />}>
          <Route path="home" element={<Home />} />
          <Route path="about" element={<About />} />
        </Route>
      </Routes>
    </Router>
  );
}
```