ARYATECHNO

React - Router

Topics : React JS
Written on January 03, 2024

React Router is a library for handling navigation and routing in React applications. It enables developers to create a single-page application with navigation and views that are dynamically rendered based on the URL. React Router helps manage the application's UI state and keeps the URL in sync with the components being displayed.

Here's a basic overview of React Router and its key components:

1. **Installation:** To use React Router, you need to install it as a dependency in your project. You can do this using npm or yarn:

   ```
   npm install react-router-dom
   # or
   yarn add react-router-dom
   ```

2. **Router Component:** React Router provides different types of routers for different environments. The most commonly used one is `BrowserRouter`, which uses HTML5 history API for navigation. It should be placed around the root of your application:

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

   function App() {
   return (
   <Router>
   {/* Routes go here */}
   </Router>
   );
   }
   ```

3. **Route Component:** The `Route` component is used to define the mapping between a URL path and the component that should be rendered when the path matches. It can be used in different ways, such as rendering a component inline or using the `render` prop:

   ```
   <Route path="/home" component={Home} />
   // or
   <Route path="/about" render={() => <About />} />
   ```

4. **Switch Component:** The `Switch` component is used to render the first `Route` or `Redirect`

that matches the current location. It's handy when you want to exclusively render one route at a time:

```
<Switch>
<Route path="/home" component={Home} />
<Route path="/about" component={About} />
<Route path="/contact" component={Contact} />
<Route component={NotFound} />
</Switch>
```

5. **Link Component:** The `Link` component is used to navigate between different views in your application. It renders an anchor (`<a>`) tag but prevents the browser from doing a full page refresh:

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

   // ...

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

6. **Navigation with useHistory:** You can also use the `useHistory` hook to programmatically navigate in response to user actions or events:

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

   function MyComponent() {
   const history = useHistory();

   const handleClick = () => {
   history.push('/new-route');
   };

   return (
   <button onClick={handleClick}>Go to New Route</button>
   );
   }
   ```