History

history is a JavaScript library that lets you easily manage session history in any JavaScript environment. History provides a minimal API that lets you manage the history stack, navigate, confirm navigation, and persist state between sessions. — React Training docs

Each router component creates a history object that keeps track of both the current locations (`history.location`) and the previous locations in a stack. When the current location changes, the view is re-rendered and you get a sense of navigation and interaction. How does the current location change? The history object has methods such as `history.push()` and `history.replace()` to take care of that. `history.push()` is invoked when you click on a `<Link>` component, and `history.replace()` is called when you use `<Redirect>`. Other methods — such as `history.goBack()` and `history.goForward()` — are used to navigate through the history stack by going back or forward a page.

Links and Routes

• The `<Route>` component is the most important component in React router. It renders a part of the UI if the current location matches the route's path. Ideally, a `<Route>` component should have a prop named `path` and if the pathname is matched with the current location, it gets rendered.

• The `<Link>` component, on the other hand, is used to navigate between pages. It's comparable to the HTML anchor element. However, using anchor links would result in a browser refresh, which we don't want to happen. So instead, we can use `<Link>` to navigate to a particular URL and have the view re-rendered without a browser refresh.

‹ Previous   next ›