**React**

**Day 65th**

**React Routing**

Routing is an essential technique for navigation among pages on a website based on user request and action. This is the most important part in any application. It comes in client side routing as well as server side.

React does not have any routing mechanism but it can achieve it using a library called `**React router dom**` which is a third party library and enables react to do routing between pages and allows defining multiple routes. This library can be used in frontend as well as backend as well as cloud side rendering.

**What is React Router**

React Router is a JavaScript framework that lets us handle client and server side routing in react applications. It enables the creation of single page web or mobile apps that allow navigating without refreshing the page. It allows us to use browser history features while preserving the right application view.

React router enables client side routing.

In traditional websites, the browser requests a document from a web server, downloads and evaluates CSS and JS assets, and renders the HTML sent from the server. When the user clicks a link it starts the process all over again for a new page.

Client side routing allows your app to update the URL from a link click without making another request for another document from the server. Instead, your app can immediately render some new UI and make data request with fetch to update the page with new information.

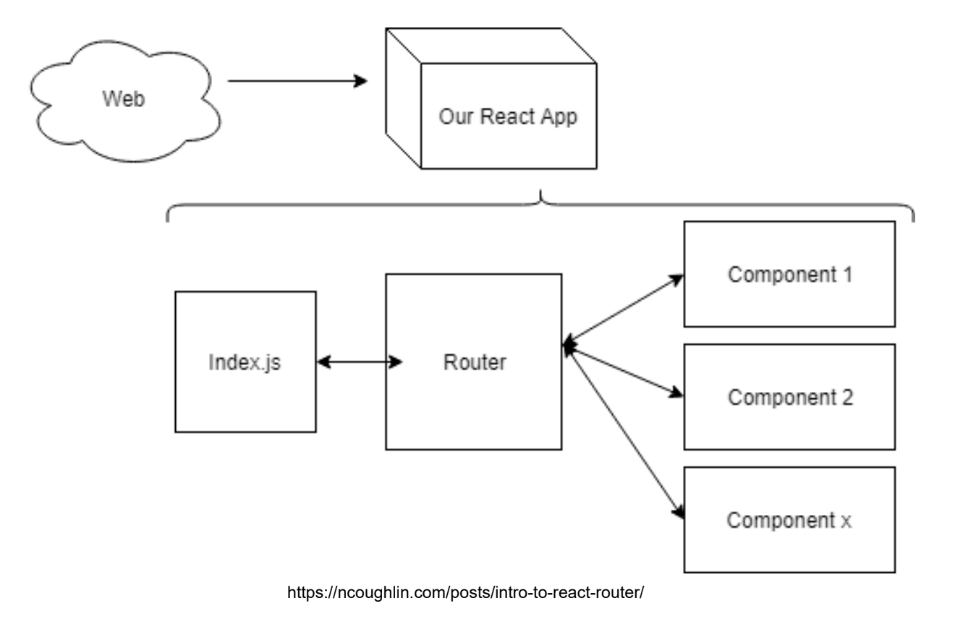
This enables faster user experiences becaurs the browser does not need to request an entirely new document or re evaluates CSS and JS assets for the next page. It also enables more dynamic user experiences with things like animations.

**What is client side routing in react**

Client side routing is one of the most important features to make use of while developing single-page web applications. In web development, routing describes the process by which a user clicks a link on a webpage and is directed to a new page with different content and a unique URL.

**What is routing in React**

React Router is a JS framework that lets us handle client and server side routing in react applications. It enables the creation of single page web or mobile apps that allow navigating without refreshing the page. It allows us to use browser history features while preserving the right application view.



React Router is a popular standard library for routing among various view components in react applications. It helps to keep the user Interface in Sync with the URL. It allows defining which view to display for the specified URL.

**How a component travels in websites**

Let’s take an example just consider you are having 5 main components for your website like (home, about, services, why choose us, login) and so on. If we talk about local URL if you are dealing with react then you will be having like this `**http://localhost:5143**` so this is our base URL and it will be taking you to the first landing page for your website or we can say it as Home page. Now if you want to move to `**about**` section your URL will be looking like this `**http://localhost:5143/about**` means here we will be having path for about component and same goes to other components.

**How it is handled**

So basically we have **React router** which we can get from a third party library `**react router dom**` and it monitors all the thing. We got three things after installing the library (React-router, React-router-DOM and React-router-Native). Where React Router contains the core functionality of react router including route matching algorithm. It’s not used any more.

React-router-DOM includes everything in react router and adds a few DOM specific APIs.

React-router native includes everything in react router adds a few React Native APIs.

**What is React Router DOM**

The primary functionality of react-router-dom is implementing dynamic routing in web applications based on the platform and the requirement of the applications. It supports component based routing which is the ideal solution for routing in the react application running on the browser.

**How to implement in code**

So for the very first time you need to move to the root component i.e. App.jsx and here you need to import three things from the react-router-DOM and these are `**BrowserRouter, Routes, Route**`.

**🤷🏼‍♀️ Browser Router**

Browser router works with HTML file history API. Now already in React router Dom dynamic routing is there.

Browser Router is a router implementation that uses the HTML5 history API (pushState, replaceState, and popState event) to keep your UI in sync with the URL. It is the parent component that is used to store all of the other components.

These three things (pushState, replaceState and popState) are part of HTML5 history API and are used to manipulate the browser’s history and URL without reloading the page – which is very useful for single page applications.

1. History.pushState: It adds a new entry to the Browser’s session history stack. Adds the new state to the history, so the user can press back to return.
2. History.replaceState: It replaces the current history entry with a new one. It does not add a new entry to the browser’s history. The back button will not go to the previous replaced URL.
3. popState event: It is fired when the active history entry changes i.e. when user clicks back and forward button. It is useful for detecting when users navigate through browsers button.

While moving from a component to another component it passes some data. But why understand with an example here. Suppose there is a website let’s take an example of Instagram only without logging in you can’t scroll reels, post, even you can’t see anything related to account right. But while logging in how the website understands that we have logged in. Here we are passing data you can say a kind of token that will reside in the browser after logging in. So we can say here when we were on login page we didn’t have any data over there right but as soon as we logged in we got data in the form of token and moved to our dashboard.

🤷🏼‍♀️ **Routes**

It is nothing but a kind of component introduced after react-router-DOM version 6 onwards and it replaces the switch components. It renders a branch of <Routes> that best matches the current location. Note that these routes do not participate in data loading, actions, code splitting or any other route module features.

In react router DOM v6+, the <Routes/> replaces the older <Switch/> component. It is used to define all available route paths..

🤷🏼‍♀️**Route**

It is child component which renders a specific UI component when URL is matches the specified path using path attribute.

📝Page not found component should be at the bottom of the route.