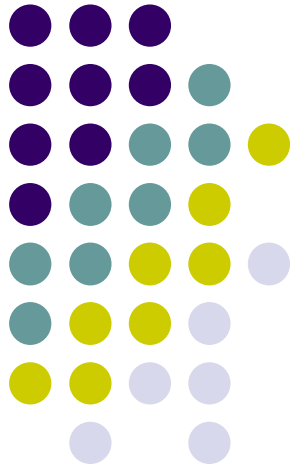
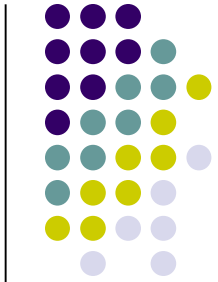
React Router

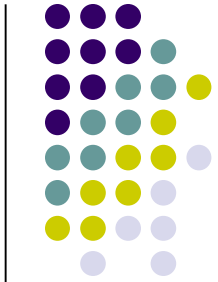


React Router



- React Router is mainly used for developing a Single Page Web Applications. It plays an important role to display multiple views in a single page application.
- Routing is a process in which a user is directed to different pages based on their action or request.
- When a user types a specific URL into the browser, and if the URL path matches any 'route' inside the router file, the user will be redirected to that particular route.
- Most of the social media websites like Facebook, Instagram uses React Router for rendering multiple views.

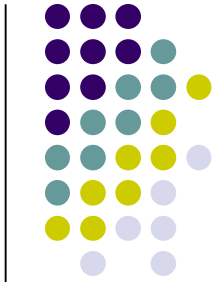
React Router Installation



- React Router is a standard library system. If you want to create routing in the React application using React Router Package.
- It is not possible to install react-router directly in your application. To use react routing, first, you need to install react-router-dom modules in your application.
- The below command is used to install react router dom.

  $npm install react-router-dom--save

- react-router-dom: It is used for web applications design.



Route

It is the conditionally shown component based on matching a path to a URL.

Routes

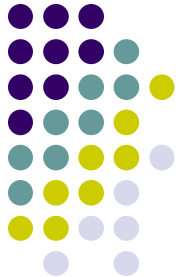
With Routes we can form a tree of Route components in a ordered segments.

We can order our route according to our static segment and dynamic segment.

BrowserRouter

- It is the router implementation It uses HTML5 history API (i.e. pushState, replaceState and popState API) to keep your Ul in sync with the URL.

import{BrowserRouter as Router, Route, Link, NavLink, Switch}

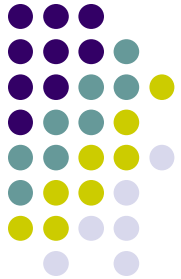


Memory Router

- Memory router keeps the URL changes in memory not in the user browsers.
- It does not read or write to the address bar so the user can not use the bowser's back button as well as the forward button.
- It is very useful for testing and non-browser environments like React Native.

Hash Router

- HashRouter uses a hash symbol in the URL, which has the effect of all subsequent URL path content being ignored in the server request
- When we have small client side applications which doesn't need backend we can use HashRouter because when we use hashes in the URL/location bar browser doesn't make a server request.

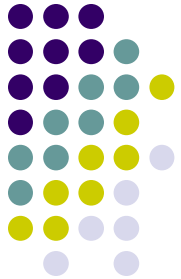


Link

- It is your replacement for anchor tags. Sometimes, we want to need multiple links on a single page.
- This <Link> component is used to create links which allow to navigate on different URLS and render its content without reloading the webpage.

NavLink

- component is used to add styles to the active routes and add properties activeStyle.
- The activeStyle properties mean when we click on the Link, it should have a specific style so that we can differentiate which one is currently active.



Switch

- The <Switch> component is used to render components only when the path will be matched. Otherwise, it returns to the not found component.
- For the latest version of router we are not using it instead we are using **‘*’** operator for path mismatch’s.

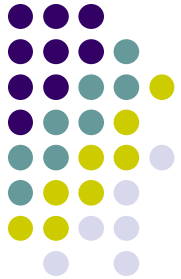
Element

- After we have defined our routes and set up data loaders for them, we can finally tell React Location what to render when those routes are matched.
  - •element: A React element to render when the route is matched.
  - •errorElement: A React element to render when the route is matched but an error occurs.
  - •pendingElement: A React element to render when the route is matched and enters a pending state.

useNavigate

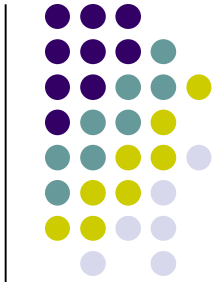
- The **useNavigate() hook** is introduced in the React Router v6 to replace the useHistory() hook.
- In the earlier version, the useHistory() hook accesses the React Router history object and navigates to the other routers using the push or replace methods.
- It helps to go to the specific URL, forward or backward pages
- In the updated version, the React Router's new navigation API provides a useNavigate() hook which is an imperative version to perform the navigation actions with better compatibility.

useParams



- In our React app sometimes we want to access the parameters of the current route in this case useParams hook comes into action.
- The react-router-dom package has useParams hooks that let you access the parameters of the current route.

Sensitive vs Non-Sensitive



- The sensitive prop ensures that the path prop's case is taken into consideration when matching it with the browser's URL path.
- By adding the sensitive prop, one can define routes with the same pathname, but do so using a different case.

Dynamic route

- In Dynamic Routing we can pass params in the url, based on the params the page will be loaded.
- It will be much useful when we use API call for the page, based on the params.
- The Dynamic route can also be accessible in Child or nested route.

- React Router
  - Installation
  - Router Types
  - Route
  - Routes
  - Switch
  - Link
  - NavLink
  - Element


QUERIES