**Theory Chapter 08**

**Q: Different types of router in react-router-dom**

**A: 1. createBrowserRouter :**  It uses DOM API history to manage and update the history stack . It also enables the v6.4 data APIs like [loaders](https://reactrouter.com/en/main/route/loader), [actions](https://reactrouter.com/en/main/route/action), [fetchers](https://reactrouter.com/en/main/hooks/use-fetcher) and more. Mostly web application uses this

**2. createHashRouter :** It uses the hash portion of the url to manage the application url

**3.. createMemoryRouter :** It does not uses the browser history api instead it creates the own history stack in the memory . Can also be used in non – browser environment

**Q: how to create nested Routes**

**A:**

const router = createBrowserHistory([

  {

    path:'/'

    element :<ComponentNAme/>,

    children:[

      {

        path:'/about',

        element:<About />, // in this component there should be Outlet to render children

        children:[

          {

            path:'info', // path will be /about/info

            element : <Info/>

          },

          {

            path:'/info', // path will be /info

            element : <Info/>

          }

        ]

      }

    ]

  }

])

**Q: Order of lifecycle method calls in class based components**

1. Constructor
2. Render
3. componentDidMount // if there is any state is being updated then after 3 point ll happen
4. shouldComponentUpdate // if it return true then after 4 ll happen
5. render
6. componentDidUpdate
7. compoentWillUnmount

**Q: Why one use componentDidMount?**

**A:** componentDidMount to make any api call or any other side effect.It is a good user experience that the user sees some initial ui being render rather than a blank screen and after that suddenly the data is coming in the ui . so as componentDidMount ll call only after initial render , so user can see the initial ui being render in the screen and after that the data from the api , the DOM is being updated according to the new state

**Good place to take data or load data from a remote end point**

Can use setState in that , one extra rendering ll happen but this rendering ll happen before it is been updated to the DOM

**Q: use of componentWillUnmount?**

**A: This** method Is called before removing the component from the DOM. It is basically used for cleaning purpose. Like one has started an interval in componentDidMount then that interval should be cleaned when the component is being unmounted from the DOM . so for cleaning the interval it should happen in componentWillUnMount. Or removing any eventListerner which is attached in componentDidMount

**Q: Use of super(props) in constructor?**

**A:**  one does not constructor if there is not state initialization in the component

**If one does not call super(props) before any other statement in the constructor then this.props will be undefined which can lead to bugs in the system**

One should not use setState in constructor , set the initial state to that particular state only.

Don’t copy the props into the state of the class based component

**Q: Why cannot have async callback function in useEffect?**

**A:**  we know that async functiom returns promises

useEffect expects a callback to return either nothing or code clean up function

NOTE: callbacks are \_only\_ allowed to return either void, or a destructor.