Major Project: Routing And Auth

What is Routing?

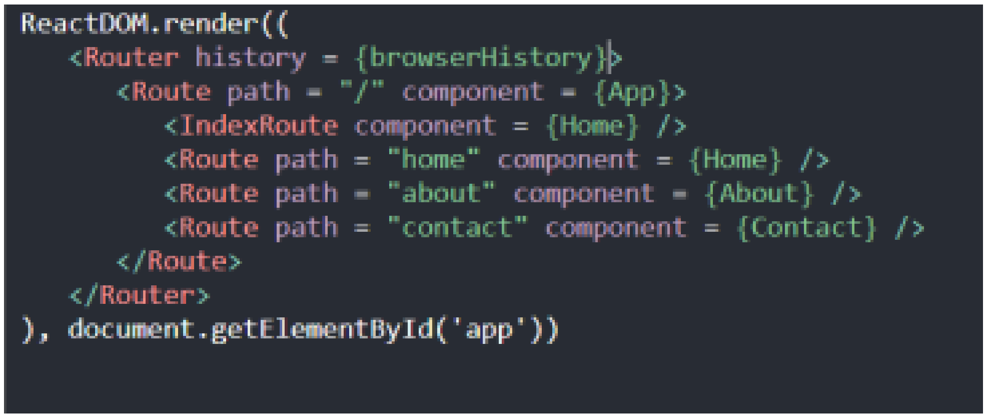
Routing is a mechanism by which requests (specified by URL and HTTP method) are redirected to the code that manages them. Routing is file-based and very simple.

React Router:

React Router is a standard library for routing in React. It allows navigation between views of different parts of the React application, allows the browser to change URLs and syncs the UI with the URL.

To install this library use: *npm install react-router-dom –save* With the Route component, you can set the path of different components that you want to render at different times.

Here is the code snippet for your reference:



```
ReactDOM.render((
   <Router history = {browserHistory}>
      <Route path = "/" component = {App}>
         <IndexRoute component = {Home} />
         <Route path = "home" component = {Home} />
         <Route path = "about" component = {About} />
         <Route path = "contact" component = {Contact} />
      </Route>
   </Router>
), document.getElementById('app'))
```

Note: The route component does not take any props as arguments other than specified in the documentation. To pass non-specified props we can use the ‘render’ argument inside the route component.

Form Handling

- **createRef:** createRef () acquires the basic DOM object as its current asset. When the Ref attribute is used in a custom class object, the ref object receives the component as the current state.

- **preventDefault:** The preventDefault () method is used to prevent the browser from performing the default action for the selected item. It may prevent the user from continuing the request by clicking the link. This event is used to indicate an event or action by a user in response to a function.

- **Target.value:** target.value is used in forms to fetch a particular value of a field.

Logging In

Here in this part, we created a login UI, and then we created hooks for the email, password, and login button. When a user enters the data the state gets updated. Also, we used react-hooks-notifications for receiving notifications when an event is handled.

react-toast-notifications: This is a small browser library that helps to decode jwt token which is encoded.
Command to install: *npm install react-toast-notifications*

Persisting The User

In this part, we learnt about how to make our app remember a particular user. We added this functionality using jwt token and jwt decode. Here, we will store the token in local storage. If the token is found, we will decode the token, and the state will be set to user details and if the token is not present then the state is set to null. Let’s understand them in detail:

Jwt Token: JSON Web Token is a proposed Internet standard for creating data with optional signature and/or optional paid encryption that keeps JSON validating a certain number of claims. Tokens are signed using a private secret or public/private key.

The scenarios when we can use jwt token:

- **Authorization:** This is the most important scenario where jwt token is used. It helps to persist the user and once the user is logged in, the jwt request will include all the resources, accesses that are permitted to the user.
- **Information Exchange:** Jwt tokens is a good way to exchange information between two parties. It also helps us to check whether the content tampered with during transmission or not.

The JWT access token is only valid for a finite period of time. Using an expired JWT will cause operations to fail. This value is normally 1200 seconds or 20 minutes.

Jwt decode: This is a small browser library that helps to decode jwt token which is encoded.

Command to install: *npm install jwt-decode*

Miscellaneous Components

- **Switch:** <Switch> component checks the routes one by one and if the path is matched that route is rendered and it stops checking the other routes after then. It is like an if condition with a break that is as soon as the condition is fulfilled the code breaks.

  For. eg: If some component like PAGE 404 error does not contain any path so it will be rendered at every page in the absence of a switch because then every route will be checked and PAGE 404 error component will satisfy the condition every time and hence will be rendered. With switch condition, this problem will be solved.

- **Link:** It provides declarative and accessible navigation around the application. Internal linking and external linking can be done through this. We can use this to link a string, object, function, components etc.

Summarising It

Let's summarise what we have learnt in this module:

- Learned about routing.
- Learned about form handling in react.
- Learned about how a user logs in and how jwt works.
- Learned about different types of routes.

Some References:

- Routing
 https://reactrouter.com/web/guides/quick-start
- Forms In React
https://reactjs.org/docs/forms.html
- react-toast-notifications
https://jossmac.github.io/react-toast-notifications
- Jwt Authentication
https://jwt.io/introduction
- Routes
https://medium.com/@thanhbinh.tran93/private-route-public-route-and-restricted-route-with-react-router-d50b27c15f5e