Routing and Higher Order Components

Note: This document is a brief introduction to Routing and Higher Order Components, they will be covered again in the Redux Lectures.

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: 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.

Miscellaneous Components

- **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 conditions, 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.

- **useParams:** useParams is a hook that returns an object of key/value pairs of URL parameters. It is used to access params of the current <Route>.

Form Handling

- **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.

Higher Order Components

Higher order components are JavaScript functions used to add additional functionality to an existing component. These functions are pure, meaning they receive data and return values according to that data. If the data changes, the high order functions are re-enabled with different data inputs. If we want to update our returning part, we do not need to change the HOC. We need to change the data that uses our function.
The High Order Component (HOC) **wraps** around the "normal" component and provides additional data input. It actually returns a function and another component that wraps around the source.

```
import React from 'react';

var Data = {
  title: 'Higher Order Component',
}

var MyHOC = ComposedComponent => class extends React.Component {
  componentDidMount() {
    this.setState({
      title: Data.title
    });
  }
  render() {
    return <ComposedComponent {...this.props} {...this.state} />;
  }
};
class MyComponent extends React.Component {
  render() {
    return (
```

```
      <div>
        <h1>{this.props.title}</h1>
      </div>
    )
  }
}

export default MyHOC(MyComponent);
```

Note: High order components are **pure functions** used for different functionalities. As you start using these functions you will feel that your app is becoming easier to maintain.

Some References:

- **Routing**
  https://reactrouter.com/web/guides/quick-start
- **Forms In React**
  https://reactjs.org/docs/forms.html
- **Routes**
  https://medium.com/@thanhbinh.tran93/private-route-public-route-and-rest ricted-route-with-react-router-d50b27c15f5e
- **High Order Components**
  https://reactjs.org/docs/higher-order-components.html