COMP 431/531: Web Development

Lecture 12: React Routing

Mack Joyner (mjoyner@rice.edu)

https://www.clear.rice.edu/comp431

Announcements & Reminders

HW #4 (Draft Front-end) due Tue. Oct. 21st at 11:59pm

classroom hw4 repo: https://classroom.github.com/a/hfLD_40C

Deploying HW #4 Front-End to Surge

- Deploying app using Surge:

  >> ng build (or npm run build)

Fix TypeScript build errors

- Project name folder appears in dist

  >> *cd dist/*

- Check dir for index.html (avoid page not found error for surge url)

  >> *surge*

- Client side only app (SPA) for React, may need to add in package.json:

  "build": "tsc -b && vite build && cp build/index.html build/200.html"

- Data persistence not a requirement for this assignment
  - page refresh or navigating away from page restores original data

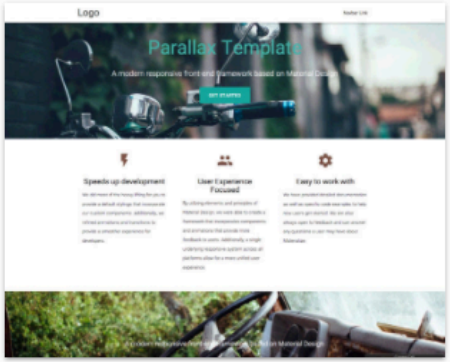
HW #4 Demo

Libraries

- bootstrap
- material
- font awesome
- material-ui/core
- react-icons
- forms-react (sign-in, registration)
- react-define (define actions)
- axios (http requests)
- tailwind, shadcn

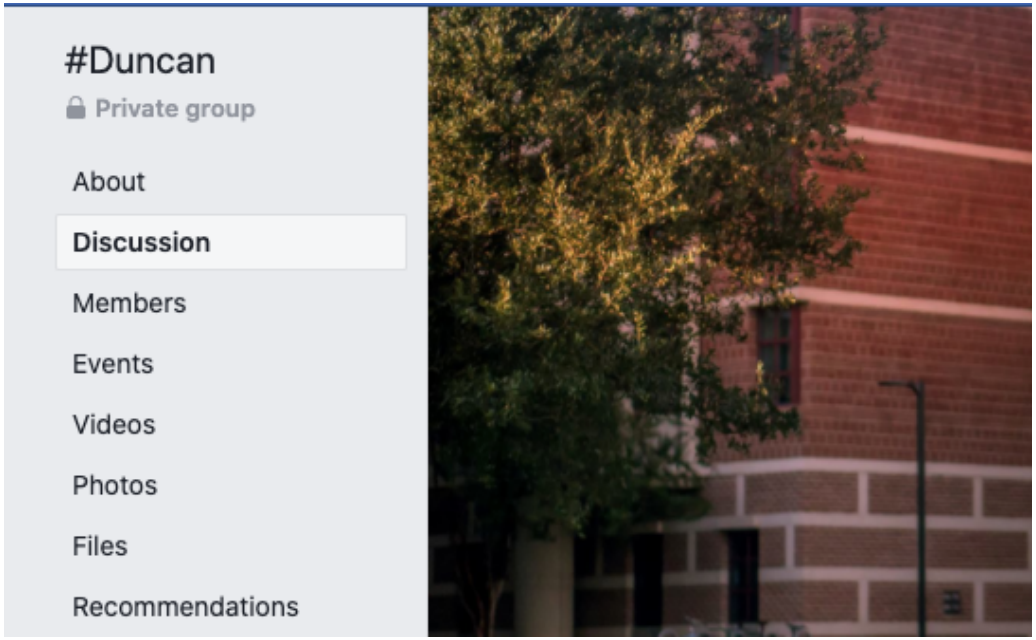
Starter Template

Parallax Template



React Routing

Navigate between different pages of application with authenticated users



React Routing

- >> *npm install —save react-router-dom*
- BrowserRouter for dynamic web app
- Import:
  - — BrowserRouter as Router
  - — Routes to bundle together multiple routes.
  - — Route employs path for navigation

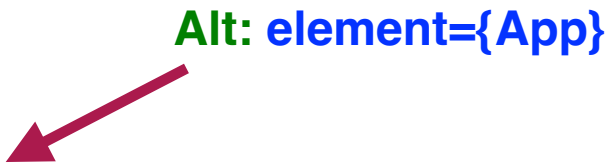
```
12    createRoot(document.getElementById('root')!).render(
13        <StrictMode>
14            <Provider store={store}>
15                <Router>
16                    <Routes>
17                        <Route exact path="/" element={<App />}/>
18
19                    </Routes>
20                </Router>
21            </Provider>
22        </StrictMode>
23    )
```

No longer use exact when matching paths in Routes

React Route with Path

- Route employs path for navigation
- Router renders element for first path match

```
13    createRoot(document.getElementById('root')!).render(
14        <StrictMode>
15            <Provider store={store}>
16                <Router>
17                    <Routes>
18                        <Route path="/" element={<App />}/>
19                        <Route path="/players" element={<Players />}/>
20                    </Routes>
21                </Router>
22            </Provider>
23        </StrictMode>
24    )
```



React Route with Path

- Alternatively use render={() => (…)} instead of element={Game}
- May use path parameters (“user/:userId”)
- Component match prop has params, url

```
<Router>
    <Routes>
        <Route exact path="/">
            <Login />
        </Route>
        <Route path="/home" render={(props) => <Main {...props}/>}/>
        <Route path="/profile" render = {(props) => <Profile {...props}/>}/>
    </Routes>
</Router>
```


Links

- Link enables navigation with hyperlinks
- Specify place to navigate (to=)
- Link can accept identifier /players/${id} where id is a defined variable
- May also find alternative react hook useNavigate() to be useful

```
15  class Game extends React.Component {
16      render() {
17          return (
18              <div className="game">
19                  <div className="game-board">
20                      <Board/>
21                  </div>
22                  <div className="game-info">
23                      <div>{/* status */}</div>
24                      <ol>{/* TODO */}</ol>
25                  </div>
26                  <div>
27                      <Link to={"/players"}>Players</Link>
28                  </div>
29              </div>
30          );
31      }
32  }
```


Router Hook: useHistory

- Access router for navigation
- Support navigating forward/back

```
import {useHistory} from "react-router-dom";

const history = useHistory();

function handleSubmit(event) {
    event.preventDefault();
    history.push({
        pathname: "/home"
    });
}

<form onSubmit={handleSubmit}>
```

Component Lifecycle Methods

- componentWillMount
  - invoked once, on client and server, before rendering occurs
- componentDidMount
  - invoked once, on client, after rendering occurs
- componentWillUpdate
  - preprocessing, before render called
- componentDidUpdate
  - postprocessing, after render called
- componentWillUnmount
  - invoked once, before unmounting component

React Hook: useEffect

- Synchronize component with external system
- Setup runs when component connects to DOM
- Dependencies list values referenced in setup

```
useEffect(setup, dependencies?)
```

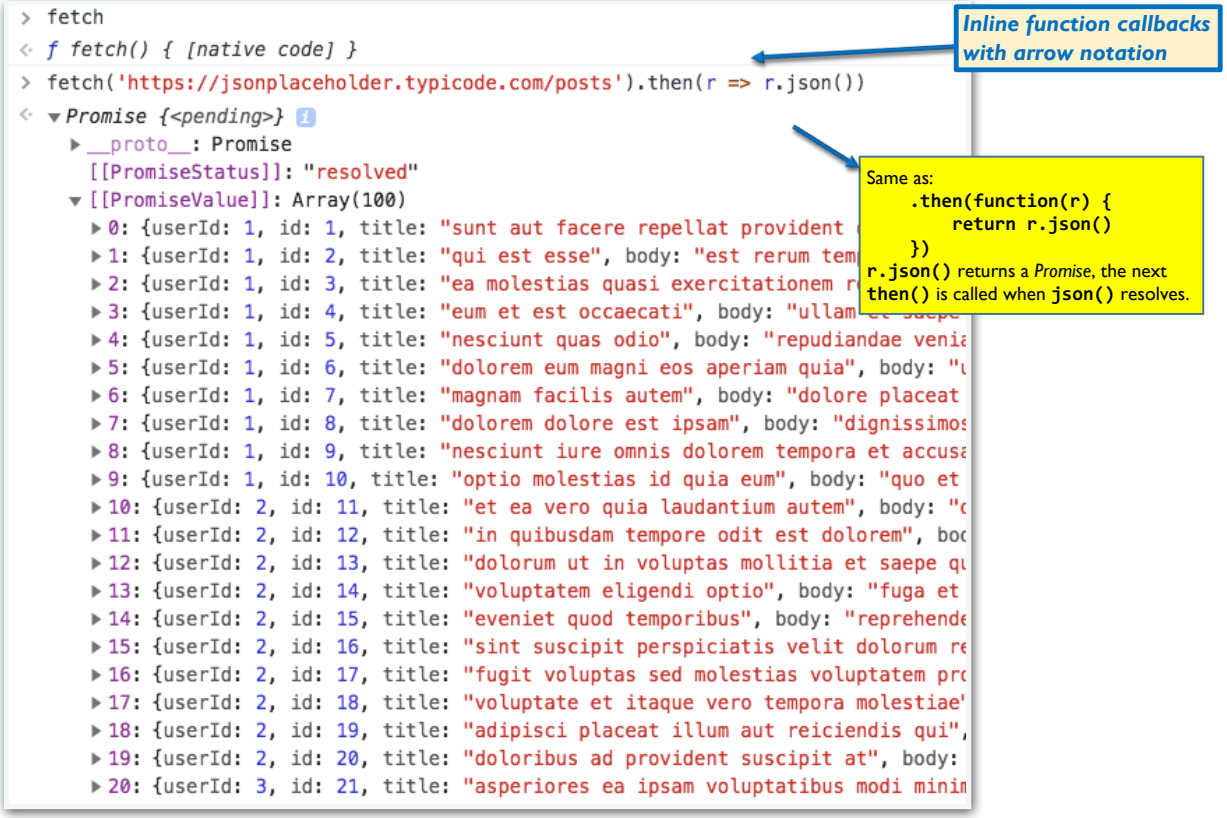
```
const [serverUrl, setServerUrl] = useState('https://localhost:1234');

useEffect(() => {
  const connection = createConnection(serverUrl, roomId);
  connection.connect();
  return () => {
    connection.disconnect();
  };
}, [serverUrl, roomId]);
```

Rerun when serverUrl or roomId changes

https://react.dev/reference/react/useEffect

AJAX Requests in React

- Components lifecycle method: requires a component
- useEffect (hook) doesn't require component class



Async/Await

- Async functions always return Promise
- Await can only be inside async

```
async function addNewArticles( ) {
   let newArts = await fetch(…)
   newArts = await newArts.json()
   newArts = newArts.articles
   articles.push(...newArts)
}
```

Expression after await is like using Promise then

Repeating Elements in Virtual DOM

```
<li><span className="fancy">thing1</span></li>
<li>thing2</li>
{ Array(5).fill(1).map((x, i) => <li key={i}>thing{3 + i}</li>) }
```

React uses the key to identify separate repeated elements.

- thing1
- thing2
- thing3
- thing4
- thing5
- thing6
- thing7

React Developer Tools (Redux)

Display when visiting players page after clicking on Players link in Board

Players

1. Bret
2. Antonette
3. Samantha
4. Karianne
5. Kamren
6. Leopoldo_Corkery
7. Elwyn.Skiles
8. Maxime_Nienow
9. Delphine
10. Moriah.Stanton

Home