Hands-on 6

Objectives

- Explain the need and benefits of React Router
- Identify the Components in React Router
- List the types of Router Components
- Parameter passing via url

In this hands-on lab, you will learn how to:

- Implement a Simple Navigation Menu
- Add Basic Routes (install, configure)
- Use Routes in React Applications

Prerequisites

The following is required to complete this hands-on lab:

- Node.js
- NPM
- Visual Studio Code

Notes

Estimated time to complete this lab: **60 minutes.**

Cognizant Academy teams want to maintain a list of trainers along with their expertise in a SPA using React as the technology. You are assigned the task of creating this React app.
The following trainers' data application will deal.

1. T-ID
2. Name
3. Phone
4. Email
5. Stream
6. Skills

1. Create a new React app using *create-react-app* tool with the as "TrainersApp"
2. Open the application using the VS Code
3. Add a new file called *trainer.js* inside the src folder and define a class named as "Trainer" with the following properties
    a. TrainerId
    b. Name
    c. Email
    d. Phone
    e. Technology
    f. Skills

```
trainer.js U
1  class Trainer {
2      constructor(trainerId, name, email, phone, technology, skills) {
3          this.trainerId=trainerId;
4          this.name=name;
5          this.email=email;
6          this.phone=phone;
7          this.technology=technology;
8          this.skills=skills;
9      }
10 }
11 export default Trainer;
```

Figure 1: Trainer.js

4. Create a new TrainersMock.js file which will contain the mock trainer data. Refer the following screenshot for mock data

```
trainersmock.js U
1  import Trainer from "./trainer";
2  const trainersMock = [
3      new Trainer('t-syed8',
4                  'Syed Khaleelullah',
5                  'khaleelullah@cognizant.com',
6                  '97676516962',
7                  '.NET',
8                  ['C#','SQL Server','React','.NET Core']),
9      new Trainer('t-jojo',
10                 'Jojo Jose',
11                 'jojo@cognizant.com',
12                 '9897199231',
13                 'Java',
14                 ['Java','JSP','Angular','Spring']),
15     new Trainer('t-elisa',
16                 'Elisa Jones',
17                 'elisa@cognizant.com',
18                 '9871212235',
19                 'Python',
20                 ['Python','Django','Angular'])
21 ]
22 export default trainersMock;
```

Figure 2: TrainersMock.js

5. Install the support for React router for the dom. Execute the following command.

```
C:\Windows\System32\cmd.exe

C:\CTS-NewHandsOns\ReactHandsOns\trainersapp>npm install react-router-dom@6
```

Figure 3: Install React Router

6. Create new component named as TrainersList inside *Trainerlist.js* file. The component should accept the trainer's data as parameter and render it as a list. The list should display names of each trainers which must be clickable like a hyper link. Refer the following screenshot for the component layout.

Figure 4: TrainersList Component

7. Create a new component named as Home inside Home.js which will be responsible for displaying the following

Figure 5: Home Component

8. Modify the App component to add support for routing and defining the navigation links to Home component and TrainersList component. Use BrowserRouter, Routes, Route and Link components from the react-router-dom library.
   Define the following URL
   1. / - must redirect to home component
   2. /trainers – must redirect to trainers list component.

   The layout of the page must be similar to the following

Figure 6: App Component

9. Create a new component named TrainerDetail in *TrainerDetails.js* file.
   The component should retrieve a parameter named id from the URL with the help of "useParams" hook from the React router DOM library.
   It should query the mock trainer data using the id and display the trainer details as show in screenshot.
   Modify the TrainersList component to add Links to TrainerDetail component while passing the ID.
   Define a route in App component for the same.

Figure 7: Trainers Detail Component

10. Build and run the application. The complete layout of the application will look as follows.

Figure 8: Home

Figure 9: Trainers List

Figure 10: Trainer Details

Implementation

App.js

```
import { BrowserRouter, Routes, Route, Link } from 'react-router-dom';
import Home from './Home';
import TrainerList from './TrainerList';
import TrainerDetail from './TrainerDetails';
import trainers from './TrainersMock';
import './App.css';

function App() {
  return (
    <BrowserRouter>
      <div className="container">
        <h1>My Academy Trainers App</h1>
        <nav>
          <Link to="/">Home</Link> | <Link to="/trainers">Show Trainers</Link>
        </nav>
        <Routes>
          <Route path="/" element={<Home />} />
          <Route path="/trainers" element={<TrainerList data={trainers} />} />
          <Route path="/trainer/:id" element={<TrainerDetail />} />
        </Routes>
      </div>
    </BrowserRouter>
  );
}

export default App;
```

App.css

```
.container {
  margin: 20px;
  font-family: Arial, sans-serif;
}

nav {
  margin-bottom: 15px;
}

nav a {
  margin-right: 10px;
  text-decoration: none;
  color: purple;
}
```

CountPeople.js

```
import React, { Component } from 'react';
class CountPeople extends Component {
  constructor(props) {
    super(props);
    this.state = {
      entrycount: 0,
      exitcount: 0
    };
  }
  UpdateEntry = () => {
    this.setState(prevState => ({
      entrycount: prevState.entrycount + 1
    }));
  };
  UpdateExit = () => {
    this.setState(prevState => ({
      exitcount: prevState.exitcount + 1
    }));
  };
  render() {
    return (
      <div style={{ textAlign: 'center', marginTop: '50px' }}>
        <h1>Mall Entry/Exit Counter</h1>
        <h2>People Entered: {this.state.entrycount}</h2>
        <h2>People Exited: {this.state.exitcount}</h2>
        <button onClick={this.UpdateEntry} style={{ marginRight: '10px', padding: '10px'
}}>Login</button>
        <button onClick={this.UpdateExit} style={{ padding: '10px' }}>Exit</button>
      </div>
    );
  }
}
export default CountPeople;
```

Home.js

```
function Home() {
  return (
    <div>
      <h3>Welcome to My Academy trainers page</h3>
    </div>
  );
}

export default Home;
```

Trainer.js

```
class Trainer {
  constructor(trainerId, name, email, phone, technology, skills) {
    this.trainerId = trainerId;
    this.name = name;
    this.email = email;
    this.phone = phone;
    this.technology = technology;
    this.skills = skills;
  }
}

export default Trainer;
```

TrainerDetails.js

```
import { useParams } from 'react-router-dom';
import trainers from './TrainersMock';

function TrainerDetail() {
  const { id } = useParams();
  const trainer = trainers.find(t => t.trainerId === parseInt(id));

  return (
    <div>
      <h3>Trainers Details</h3>
      {trainer ? (
        <>
          <h4>{trainer.name} ({trainer.technology})</h4>
          <p>{trainer.email}</p>
          <p>{trainer.phone}</p>
          <ul>
            {trainer.skills.map((skill, index) => (
              <li key={index}>{skill}</li>
            ))}
          </ul>
        </>
      ) : (
        <p>Trainer not found.</p>
      )}
    </div>
  );
}

export default TrainerDetail;
```

TrainerList.js

```
import { Link } from 'react-router-dom';

function TrainerList({ data }) {
  return (
    <div>
      <h3>Trainers List</h3>
      <ul>
        {data.map(trainer => (
          <li key={trainer.trainerId}>
            <Link to={`/trainer/${trainer.trainerId}`}>{trainer.name}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
}

export default TrainerList;
```

TrainersMock.js

```
import Trainer from './Trainer';

const trainers = [
  new Trainer(
    1,
    'Syed Khaleelullah',
    'khaleelullah@cognizant.com',
    '97676516962',
    '.NET',
    ['C#', 'SQL Server', 'React', '.NET Core']
  ),
  new Trainer(
    2,
    'Jojo Jose',
    'jojo@cognizant.com',
    '9876543210',
    'Java',
    ['Java', 'Spring Boot', 'Hibernate']
  ),
  new Trainer(
    3,
    'Elisa Jones',
    'elisa@cognizant.com',
    '9123456789',
```

```
    'Frontend',
    ['HTML', 'CSS', 'JavaScript', 'React']
  )
];

export default trainers;
```

Output: