**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
7. Create a new React app using *create-react-app* tool with the as “TrainersApp”
8. Open the application using the VS Code
9. Add a new file called *trainer.js* inside the **src folder** and define a class named as “Trainer” with the following properties
   1. TrainerId
   2. Name
   3. Email
   4. Phone
   5. Technology
   6. Skills

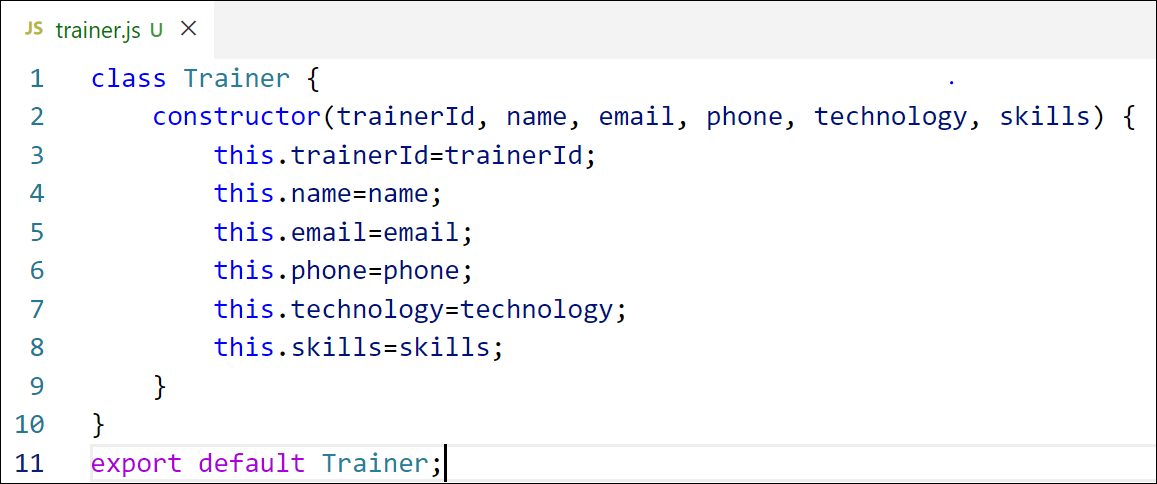


Figure 2: Trainer.js

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

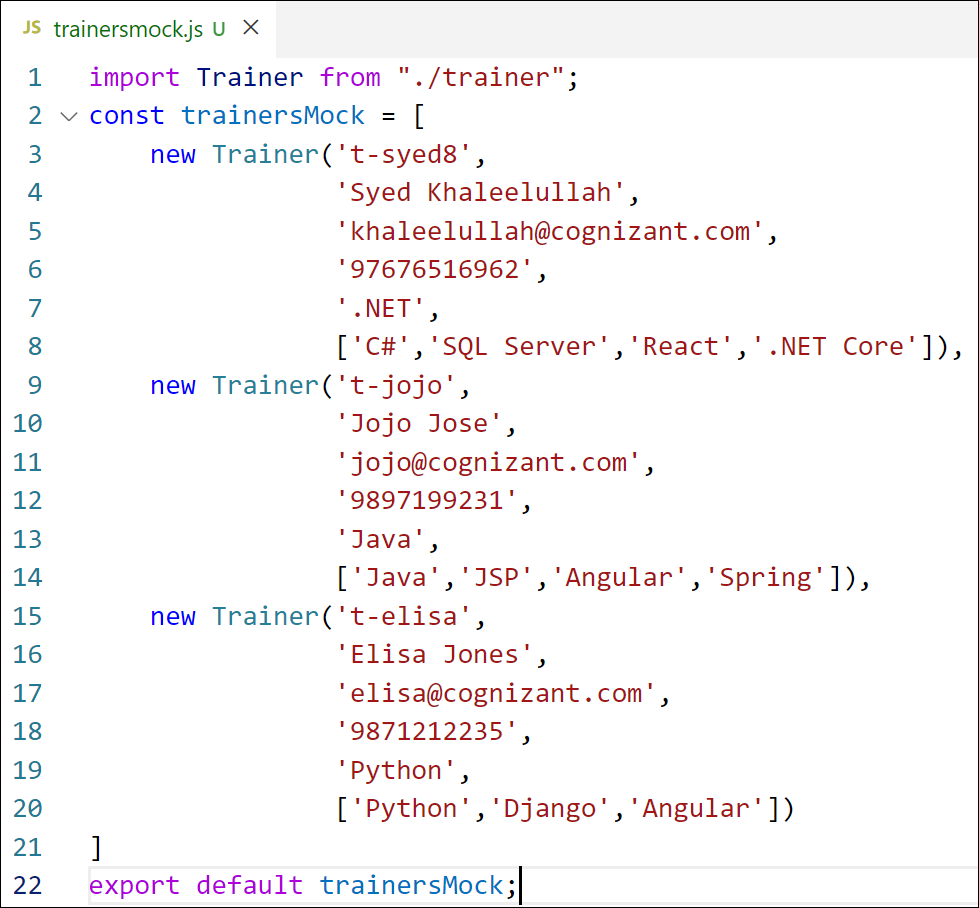


Figure 3: TrainersMock.js

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

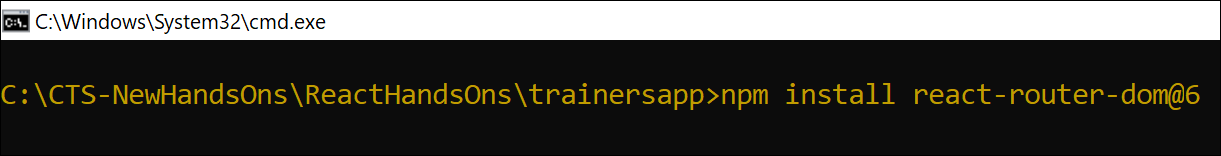


Figure 4: Install React Router

1. 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 5: TrainersList Component

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



Figure 6: Home Component

1. 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 7: App Component

1. 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.

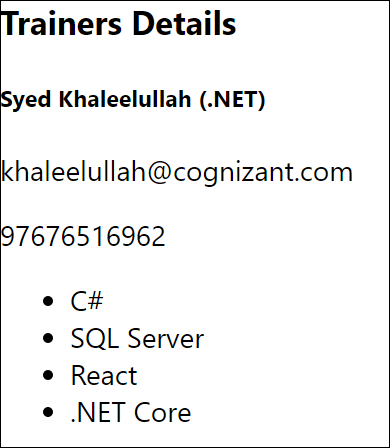


Figure 8: Trainers Detail Component

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



Figure 9: Home

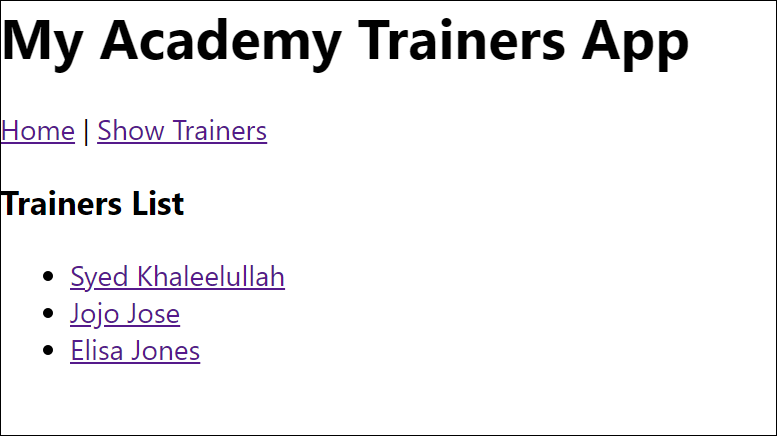


Figure 10: Trainers List

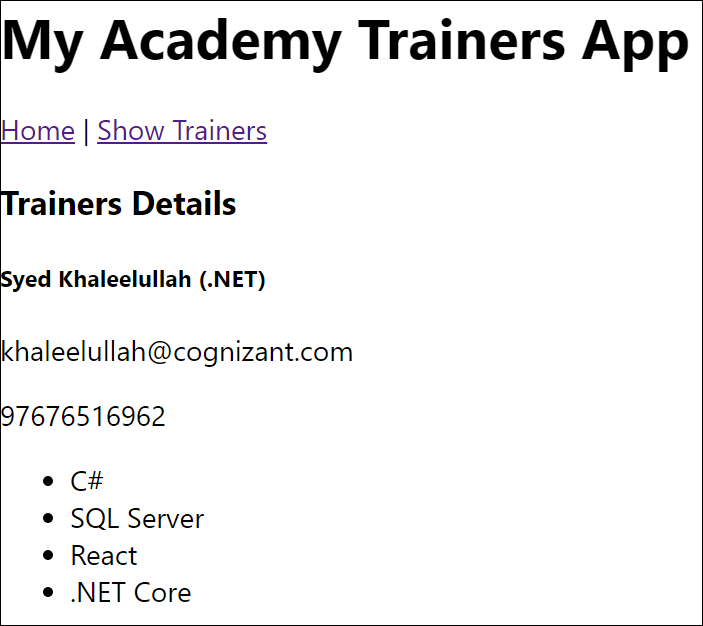


Figure 11: Trainer Details

**Solution :**

**1. Create React App**

npx create-react-app TrainersApp

cd TrainersApp

code .

2.Define Trainer Class Create src/Trainer.js:

export 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;

}

}

**3. Create Mock Data**

Create src/TrainersMock.js:

import { Trainer } from './Trainer';

const Trainers = [

new Trainer(1, "Arun Kumar", "arun@abc.com", "9876543210", "Java", ["Spring", "Hibernate"]),

new Trainer(2, "Meera Sharma", "meera@abc.com", "9999999999", "Python", ["Django", "Flask"]),

new Trainer(3, "John Doe", "john@abc.com", "8888888888", "React", ["Hooks", "Redux"]),

];

export default Trainers;

Install React Router DOM

npm install react-router-dom

Create TrainerList Component src/TrainerList.js:

import React from 'react';

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

function TrainerList({ trainers }) {

return (

<div>

<h2>Trainer List</h2>

<ul>

{trainers.map((t) => (

<li key={t.trainerId}>

<Link to={`/trainer/${t.trainerId}`}>{t.name}</Link>

</li>

))}

</ul>

</div>

);

}

export default TrainerList;

**. Create Home Component**

src/Home.js:

import React from 'react';

function Home() {

return (

<div>

<h2>Welcome to Cognizant Academy</h2>

<p>This portal manages trainer data and expertise mapping.</p>

</div>

);

}

export default Home;

**Create Trainer Detail Component**

src/TrainerDetails.js:

import React from 'react';

import { useParams } from 'react-router-dom';

import Trainers from './TrainersMock';

function TrainerDetails() {

const { id } = useParams();

const trainer = Trainers.find((t) => t.trainerId === parseInt(id));

if (!trainer) return <h3>Trainer Not Found</h3>;

return (

<div>

<h2>{trainer.name}</h2>

<p><strong>Email:</strong> {trainer.email}</p>

<p><strong>Phone:</strong> {trainer.phone}</p>

<p><strong>Technology:</strong> {trainer.technology}</p>

<p><strong>Skills:</strong> {trainer.skills.join(', ')}</p>

</div>

);

}

export default TrainerDetails;

**Update App.js to Add Routing**

src/App.js:

import React from 'react';

import { BrowserRouter as Router, Routes, Route, Link } from 'react-router-dom';

import Home from './Home';

import TrainerList from './TrainerList';

import TrainerDetails from './TrainerDetails';

import Trainers from './TrainersMock';

function App() {

return (

<Router>

<div>

<nav>

<Link to="/">Home</Link> | <Link to="/trainers">Trainers</Link>

</nav>

<hr />

<Routes>

<Route path="/" element={<Home />} />

<Route path="/trainers" element={<TrainerList trainers={Trainers} />} />

<Route path="/trainer/:id" element={<TrainerDetails />} />

</Routes>

</div>

</Router>

);

}

export default App;

Run and Test the App

npm start