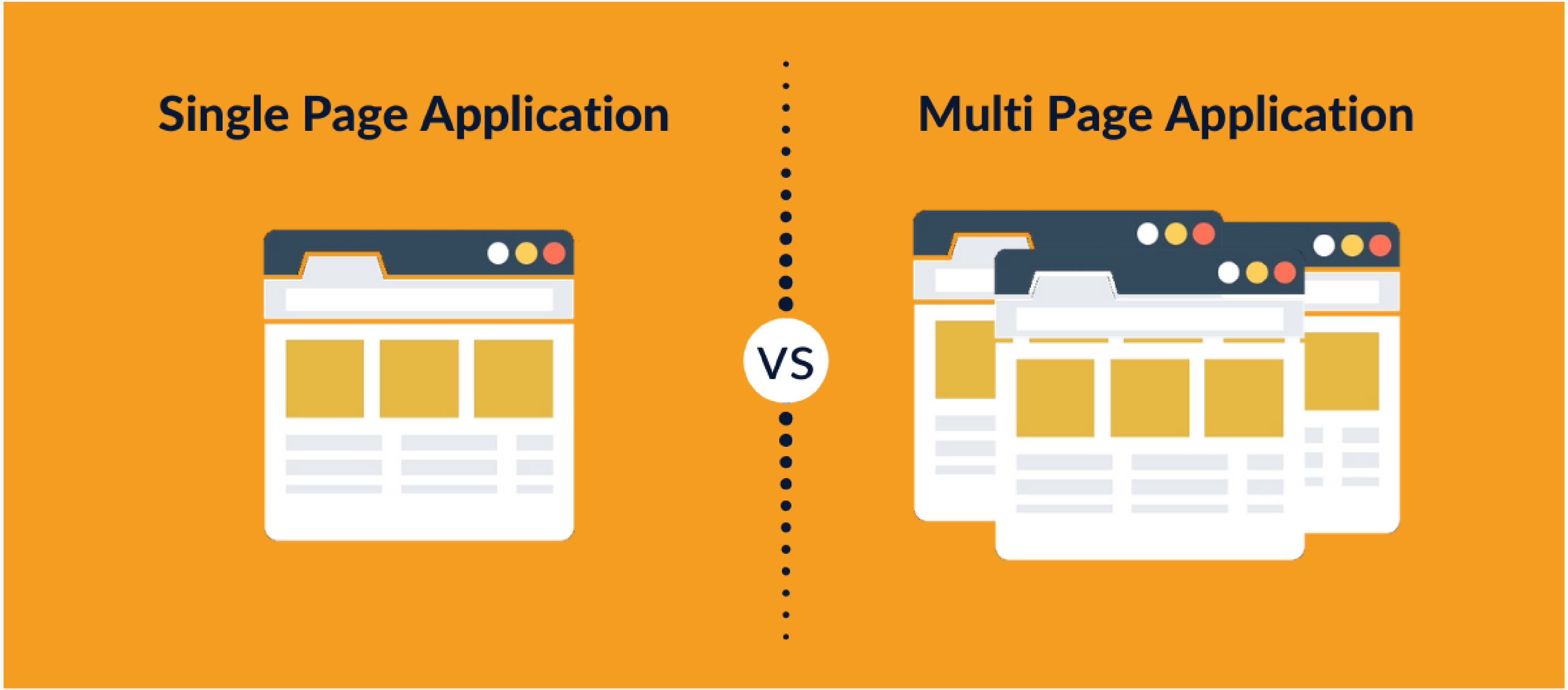
Școala
informală
de IT

React Router

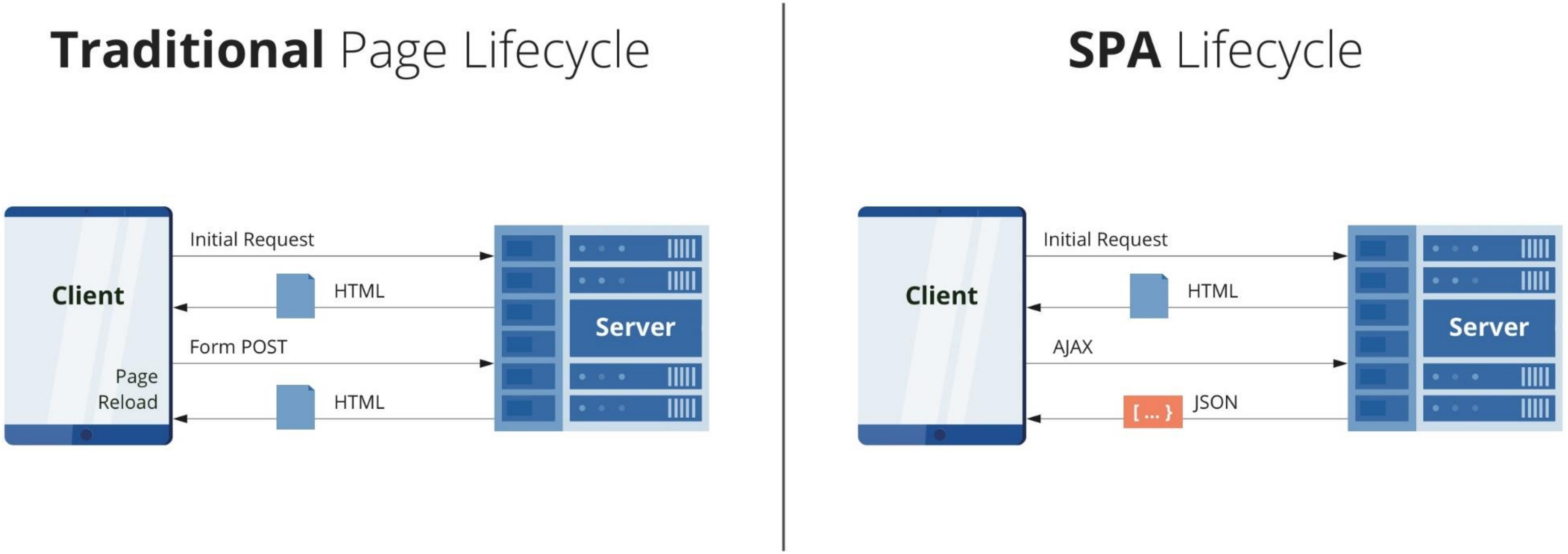
SPA – Single page application

- Multiple views – pages
- Unlike conventional pages – not all content is being rendered
- Each view should have an unique URL(https://www.socialin.com/posts)
- The back and forward button should work
- Nested views: https://www.socialin.com/posts/24343332df

SPA – Single page application

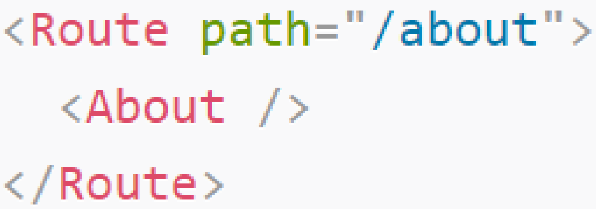


SPA – Single page application



What is routing?

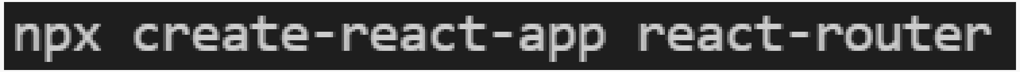
- Process of keeping the browser URL in sync with the content of the page
- React router – declarative approach
- Route is just a component
- Can be place anywhere in code



```
<Route path="/about">
  <About />
</Route>
```

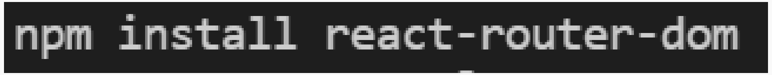
React and React Router Setup

- http://reactrouter.com – official documentation
- Create a new react project



```
npx create-react-app react-router
```

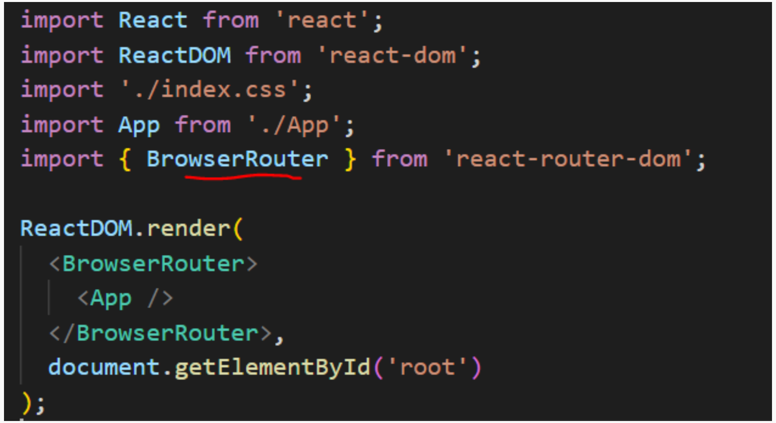
- Install react router dom package



```
npm install react-router-dom
```

React-Router Basics

- Router Component – provided by react router package
- First step – wrap <App> component into a <Router> component
- <BrowserRouter> - uses HTML history
- <HashRouter> - uses window.location.hash
- https://stackoverflow.com/questions/51974369/what-is-the-difference-between-hashrouter-and-browserrouter-in-react



```
import React from 'react';
import ReactDOM from 'react-dom';
import './index.css';
import App from './App';
import { BrowserRouter } from 'react-router-dom';

ReactDOM.render(
  <BrowserRouter>
    <App />
  </BrowserRouter>,
  document.getElementById('root')
);
```

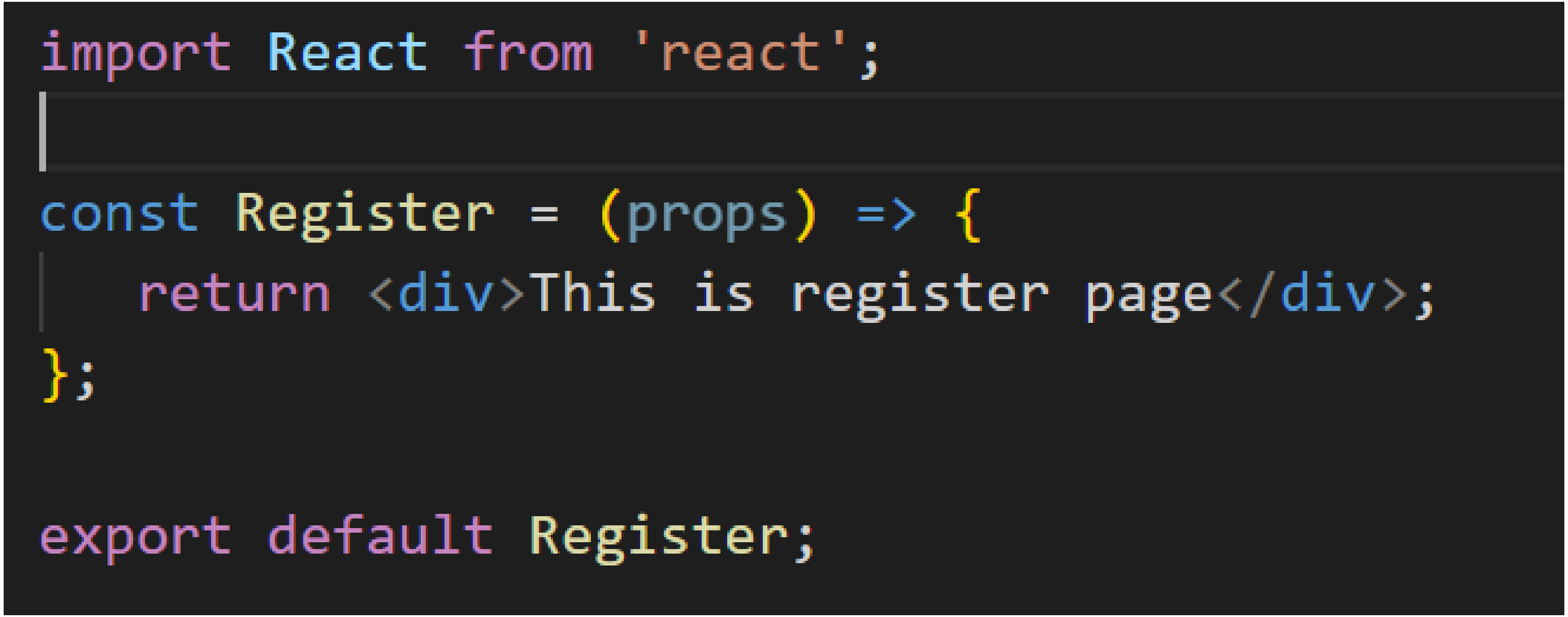
Link and Route components

- <Route> - most used component from React Router
  - Defines a path to a component
  - The component will be rendered when the path will be the location from URL

- <Link> - used to navigate between pages.
  - comparable with HTML anchor element (a)
  - Anchor links – full page refresh (entire new page)
  - <Link> - navigate to a particular URL and some parts of the UI are rerendered

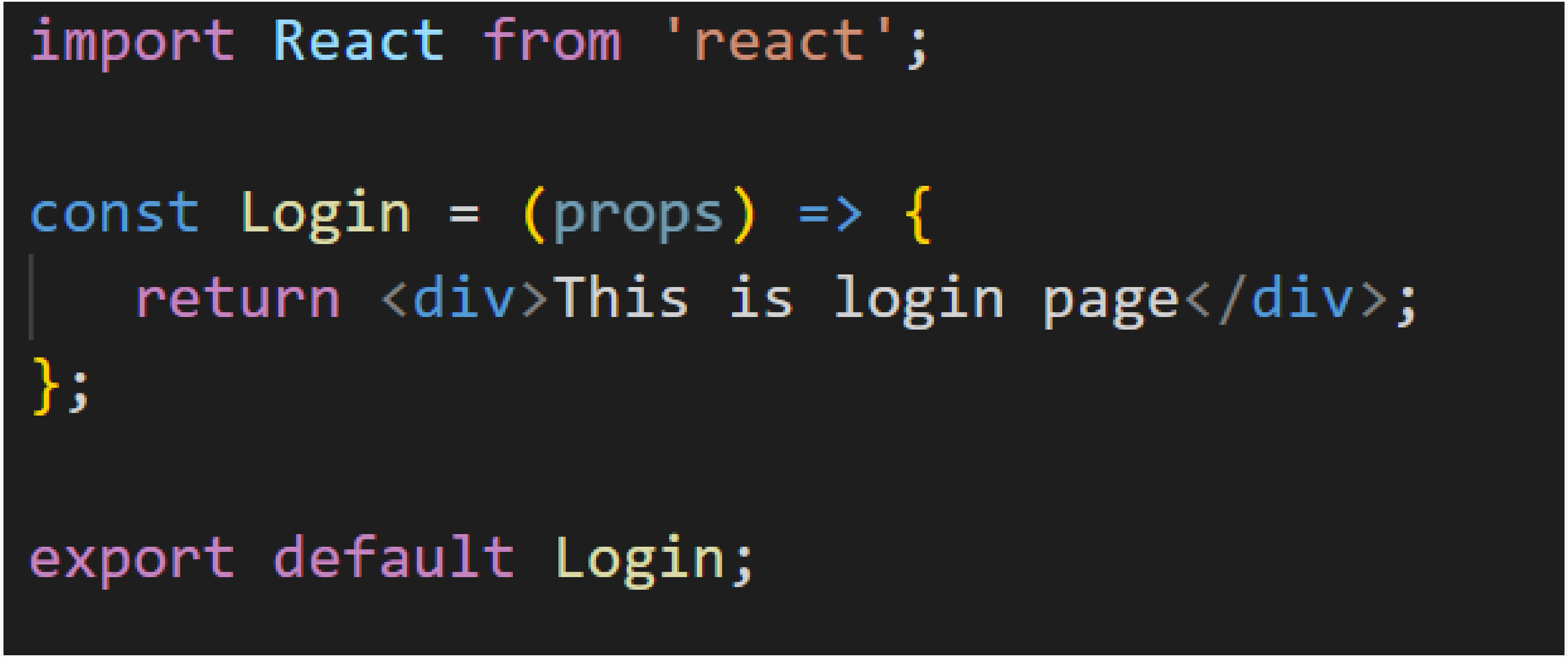
Create 3 new simple components – Register, Login, Posts

Register Component



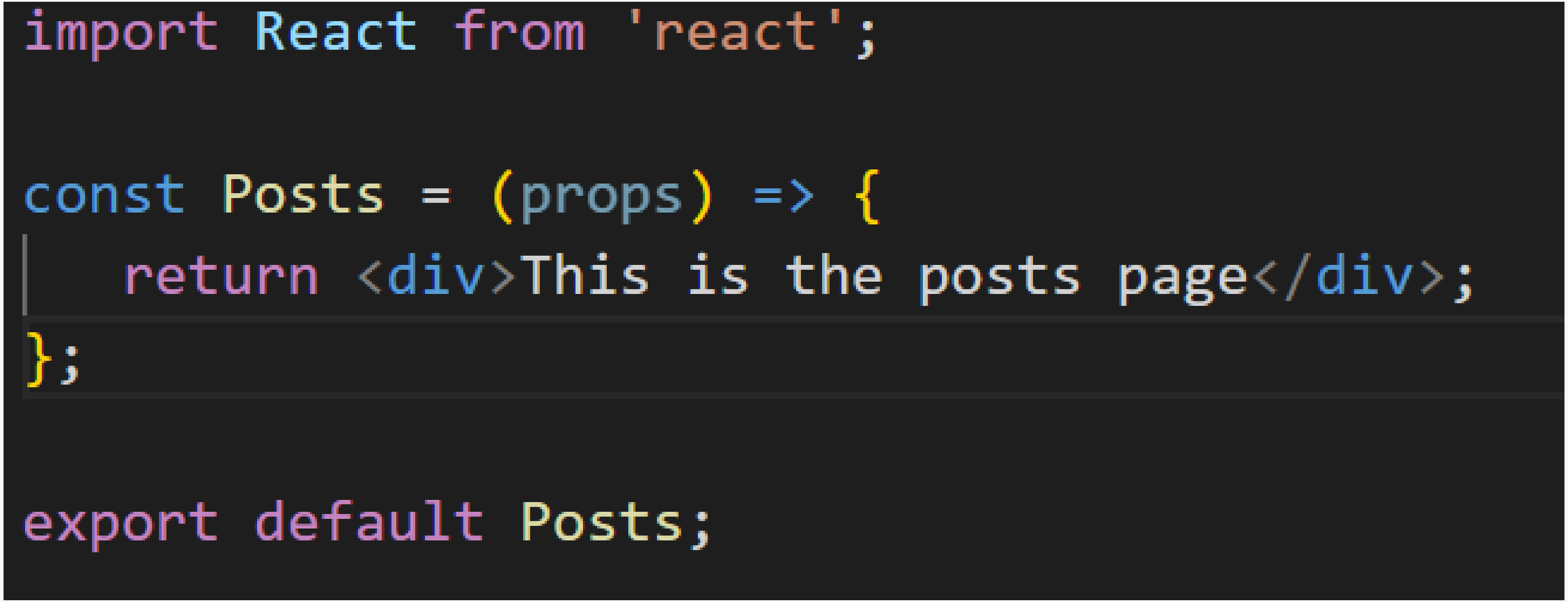
```
import React from 'react';

const Register = (props) => {
    return <div>This is register page</div>;
};


export default Register;
```

Login Component



```
import React from 'react';

const Login = (props) => {
    return <div>This is login page</div>;
};

export default Login;
```

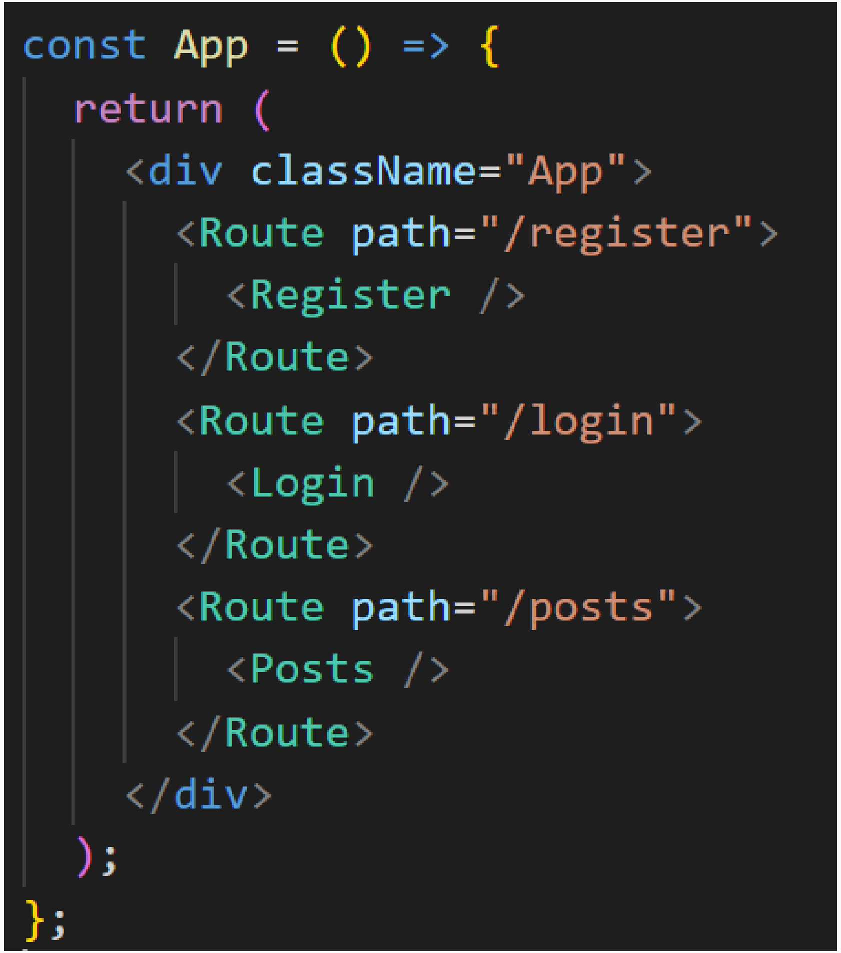
Posts Component



```
import React from 'react';

const Posts = (props) => {
    return <div>This is the posts page</div>;
};

export default Posts;
```

Route component

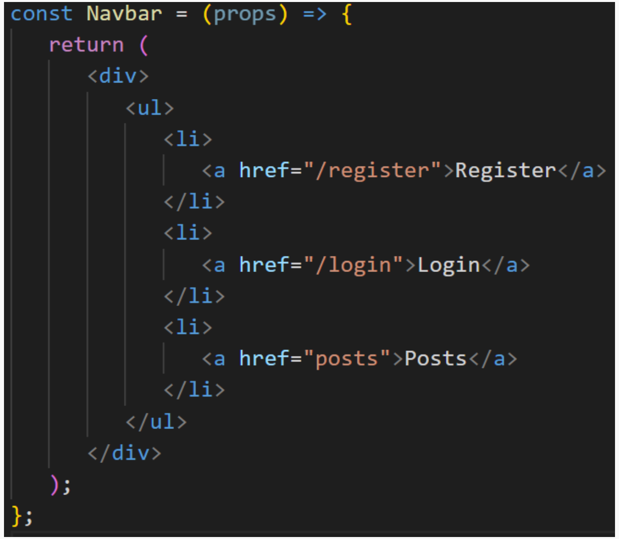
- Add routing mechanism to newly created components
- Test using
  - /register
  - /login
  - /posts
  - /dummy



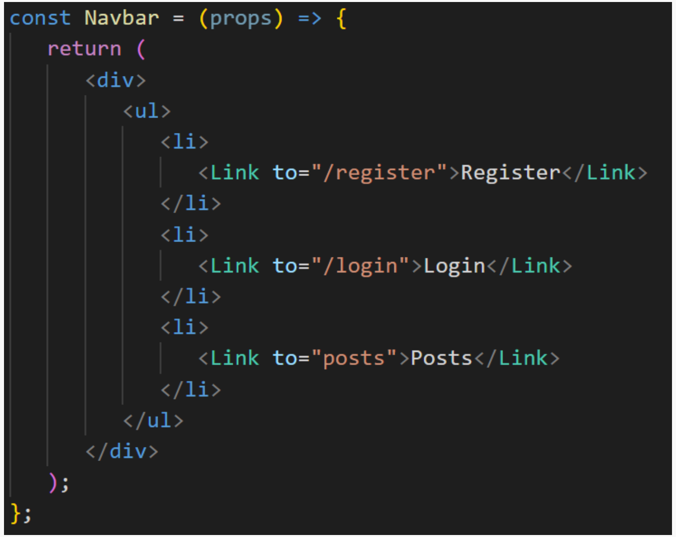
```
const App = () => {
  return (
    <div className="App">
      <Route path="/register">
        <Register />
      </Route>
      <Route path="/login">
        <Login />
      </Route>
      <Route path="/posts">
        <Posts />
      </Route>
    </div>
  );
};
```

Link component vs HTML <a> tags

- Manually type the routes → Route component
- Add links to the page → Link component or <a>
- Using <a> tags a new request is sent each time – NO SPA



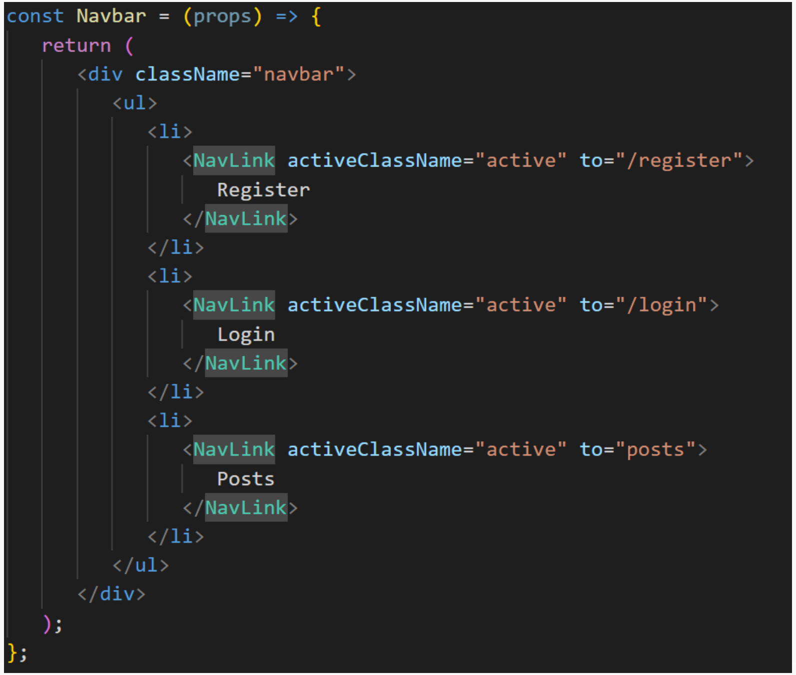
```
const Navbar = (props) => {
    return (
        <div>
            <ul>
                <li>
                    <a href="/register">Register</a>
                </li>
                <li>
                    <a href="/login">Login</a>
                </li>
                <li>
                    <a href="posts">Posts</a>
                </li>
            </ul>
        </div>
    );
};
```



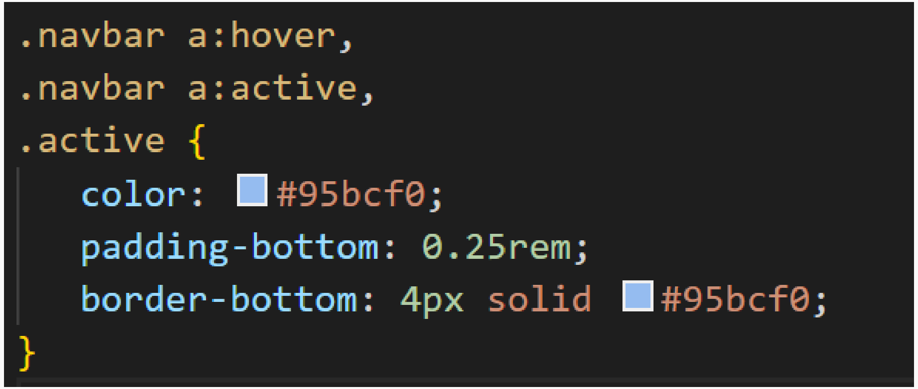
```
const Navbar = (props) => {
    return (
        <div>
            <ul>
                <li>
                    <Link to="/register">Register</Link>
                </li>
                <li>
                    <Link to="/login">Login</Link>
                </li>
                <li>
                    <Link to="posts">Posts</Link>
                </li>
            </ul>
        </div>
    );
};
```

NavLink

- Same as Link but can enable a css class to show the current Route in the UI



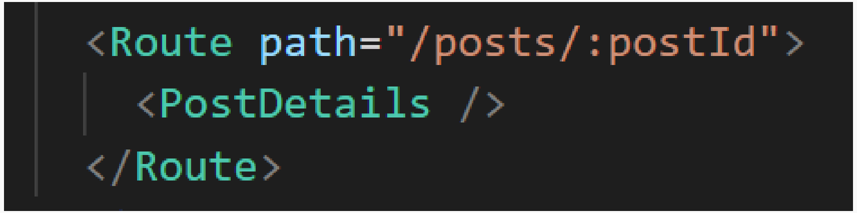
```
const Navbar = (props) => {
    return (
        <div className="navbar">
            <ul>
                <li>
                    <NavLink activeClassName="active" to="/register">
                        Register
                    </NavLink>
                </li>
                <li>
                    <NavLink activeClassName="active" to="/login">
                        Login
                    </NavLink>
                </li>
                <li>
                    <NavLink activeClassName="active" to="posts">
                        Posts
                    </NavLink>
                </li>
            </ul>
        </div>
    );
};
```



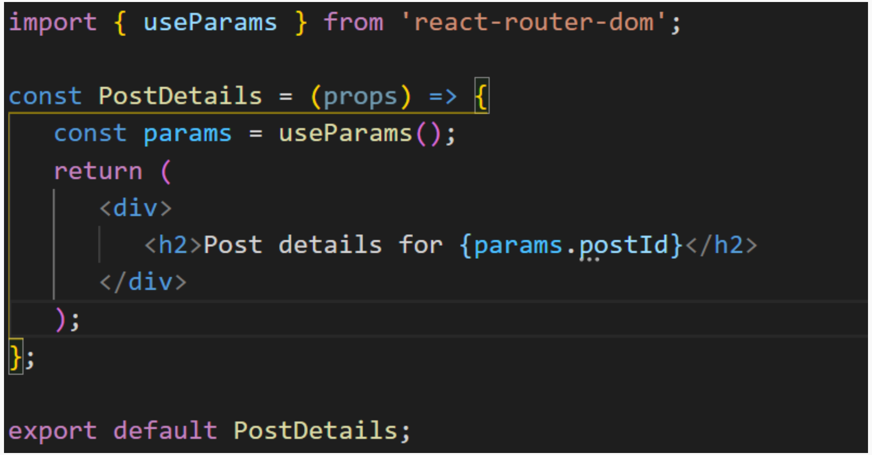
```
.navbar a:hover,
.navbar a:active,
.active {
    color: #95bcf0;
    padding-bottom: 0.25rem;
    border-bottom: 4px solid #95bcf0;
}
```

Dynamic Routes | useParams hook

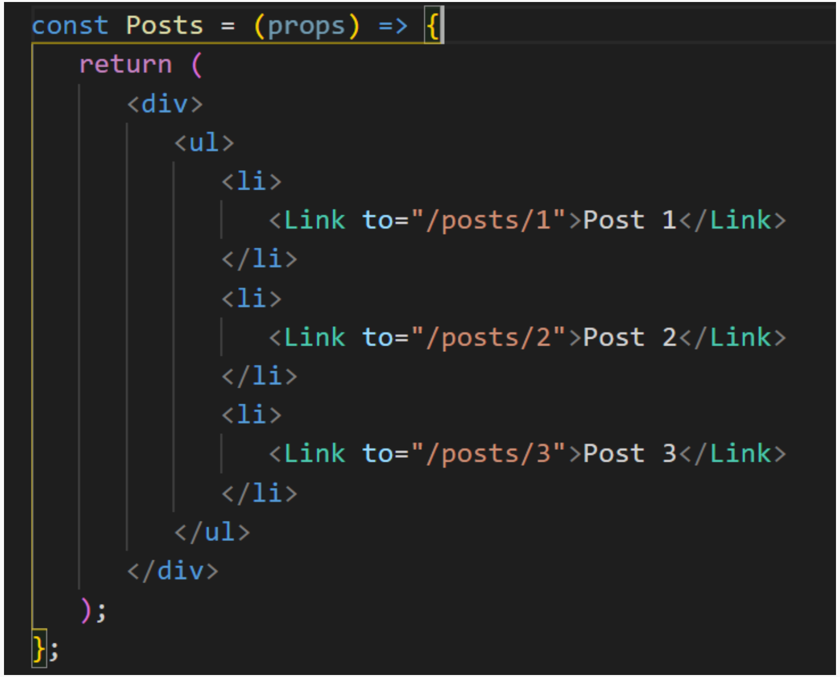
- Used to define “details” pages



```
<Route path="/posts/:postId">
  <PostDetails />
</Route>
```



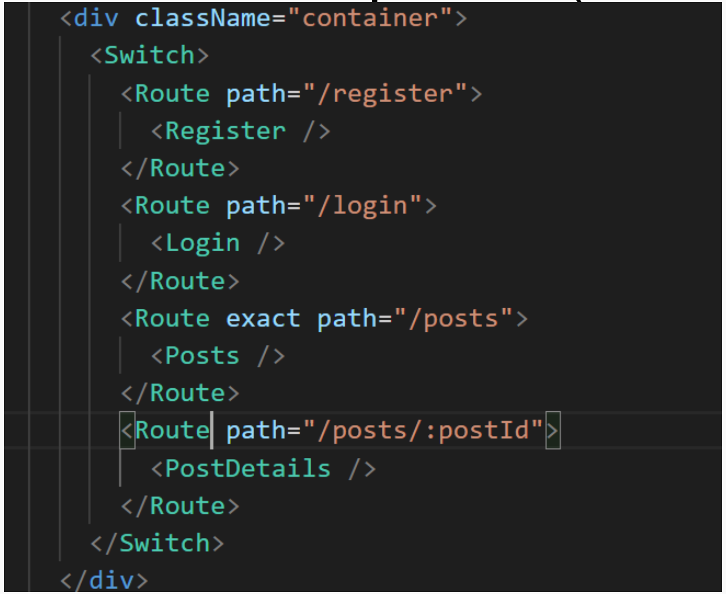
```
import { useParams } from 'react-router-dom';

const PostDetails = (props) => {
  const params = useParams();
  return (
    <div>
      <h2>Post details for {params.postId}</h2>
    </div>
  );
};

export default PostDetails;
```



```
const Posts = (props) => {
  return (
    <div>
      <ul>
        <li>
          <Link to="/posts/1">Post 1</Link>
        </li>
        <li>
          <Link to="/posts/2">Post 2</Link>
        </li>
        <li>
          <Link to="/posts/3">Post 3</Link>
        </li>
      </ul>
    </div>
  );
};
```

Switch

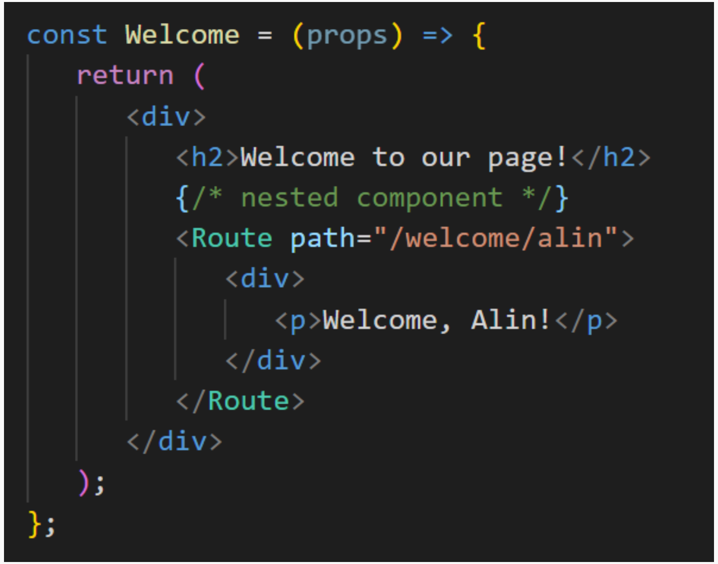
- Used to structures similar Routes
- Only one route is active when Routes and defined inside Switch component (“exact” attribute !!!)



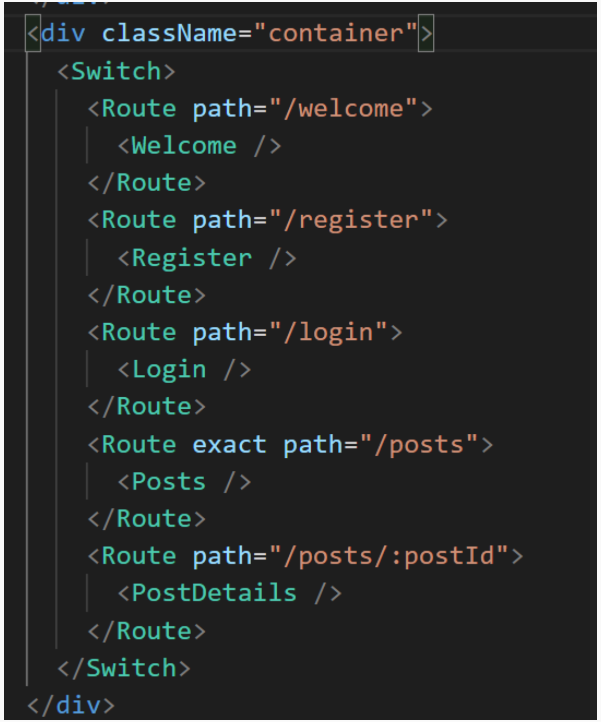
```
<div className="container">
  <Switch>
    <Route path="/register">
      <Register />
    </Route>
    <Route path="/login">
      <Login />
    </Route>
    <Route exact path="/posts">
      <Posts />
    </Route>
    <Route path="/posts/:postId">
      <PostDetails />
    </Route>
  </Switch>
</div>
```

Nested Routes

- Route component can be added inside a Routed component



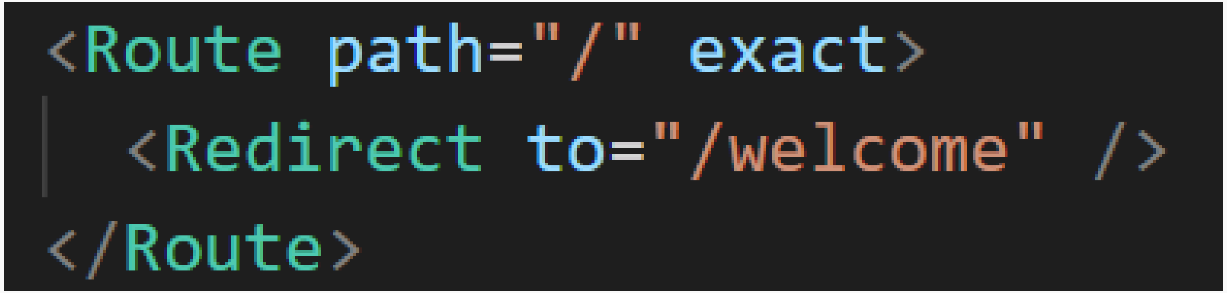
```
const Welcome = (props) => {
    return (
        <div>
            <h2>Welcome to our page!</h2>
            {/* nested component */}
            <Route path="/welcome/alin">
                <div>
                    <p>Welcome, Alin!</p>
                </div>
            </Route>
        </div>
    );
};
```



```
<div className="container">
  <Switch>
    <Route path="/welcome">
      <Welcome />
    </Route>
    <Route path="/register">
      <Register />
    </Route>
    <Route path="/login">
      <Login />
    </Route>
    <Route exact path="/posts">
      <Posts />
    </Route>
    <Route path="/posts/:postId">
      <PostDetails />
    </Route>
  </Switch>
</div>
```

Redirect

- **Used to redirect a route to another path(route)**



```
<Route path="/" exact>
  <Redirect to="/welcome" />
</Route>
```

Resources

- https://reactrouter.com/
- https://www.freecodecamp.org/news/react-router-in-5-minutes/
- https://www.javatpoint.com/react-router
- https://github.com/alingrig87/react-router-demo