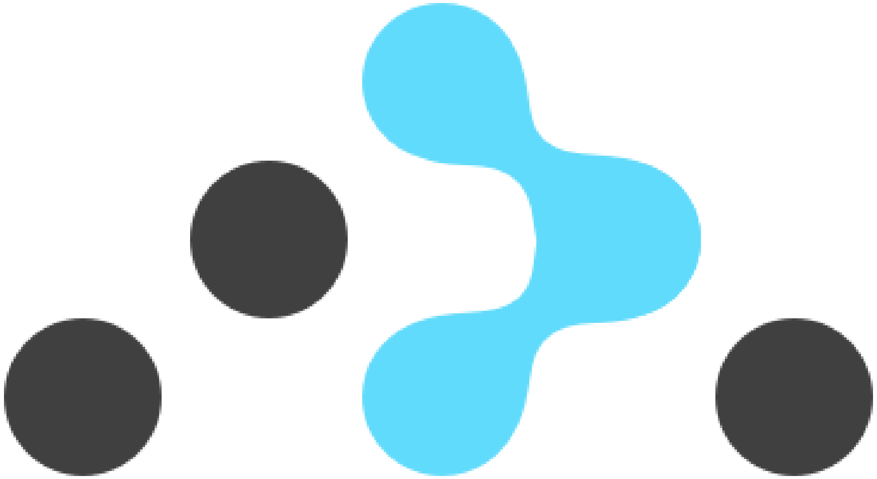


REACT ROUTER

https://reactrouter.com/

Routing

The act of moving between pages/components of an app to complete tasks

Designing effective navigation = Simplify the user journey

Routing

- Component Router ease loading components as the user interacts with the page

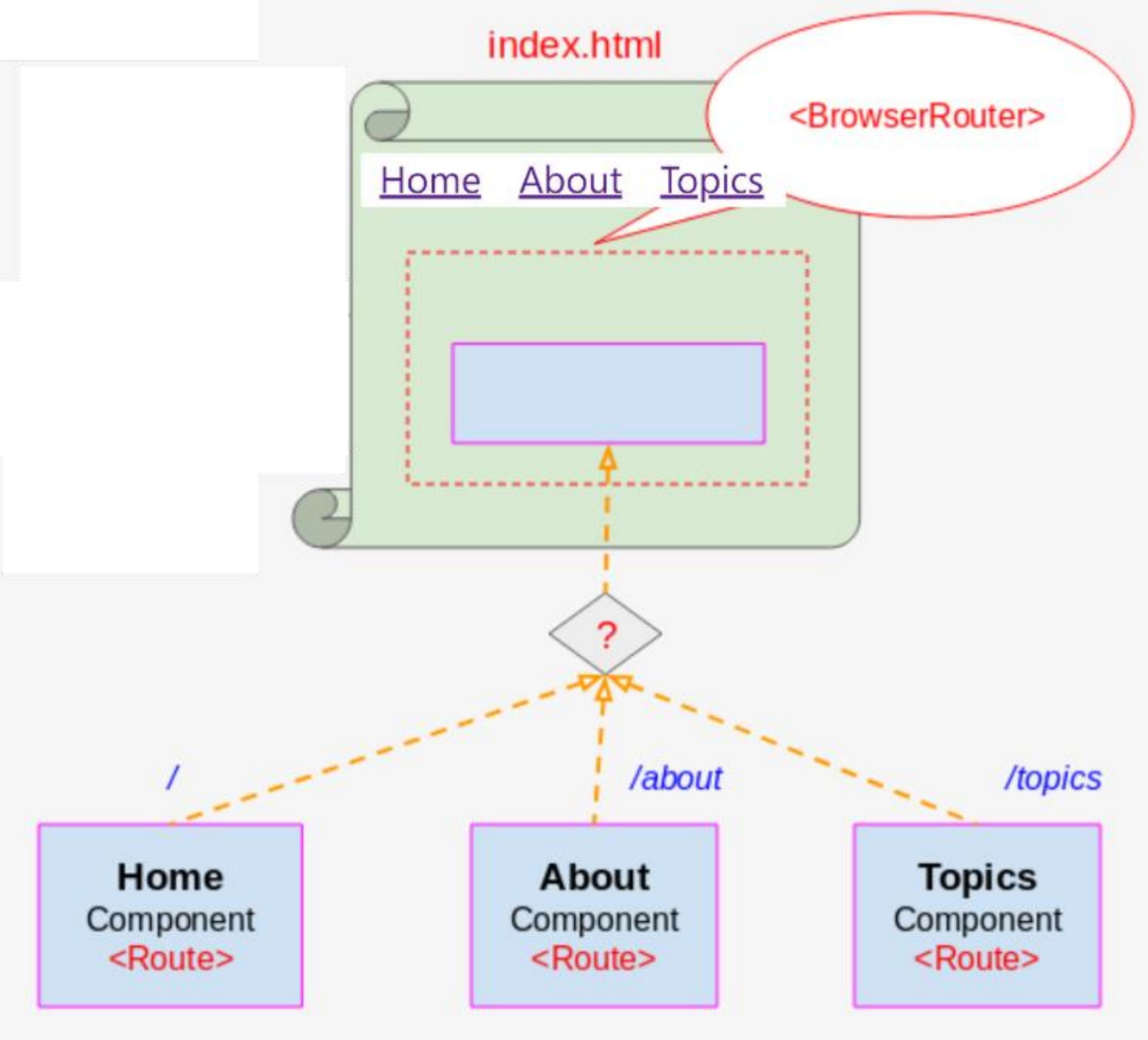
- Routing implements client-side navigation for SPA:

  - Configure routes: map relative Url to the corresponding components in a declarative way

  - On URL change the router loads the associated component

- Install

```
npm install react-router-dom@6
```


index.html
<BrowserRouter>
Home
About
Topics
?
/
/about
/topics
Home
Component
<Route>
About
Component
<Route>
Topics
Component
<Route>

Routing with React Router

```
import React from "react";
import { BrowserRouter as Router, Route, Link } from "react-router-dom";
function RouterBasicExample() {
  return (
    <Router>
        <div>
            <ul>
                <li> <Link to="/">Home</Link> </li>
                <li> <Link to="/about">About</Link> </li>
                <li> <Link to="/topics">Topics</Link> </li>
            </ul>
            <hr />
            <Routes>
              <Route exact path="/" element={Home} />
              <Route path="/about" element={About} />
              <Route path="/topics" element={Topics} />
            </Routes>
        </div>
    </Router>
  ); }
```

Configuring Routes

- React-Router allows you to declaratively define routes using the `<Route>` component
- **<Route>** component renders the component mentioned in the `element` prop when the path value mentioned in the `path` prop matches the browser's URL location

Route Parameter

- A <Route> can be configured to accept URL parameters
  - E.g, to display user info for a given userID, the URL path could look like '/user/1' for a user with
    a userID of '1', and '/user/123' for a user with
    a userID of '123’
  - E.g., Twitter's profile page can accept twitterID and displays the feed for the given user
- A <Route> component can be configured to accept the dynamic portion in the URL prefixed with a colon (:)

```
<Route path='products/:productId' element={<SingleProduct />}
/>
```

Router programmatic access

- Request the router to navigate to a Url programmatically

```
const navigate = useNavigate();
navigate('/dashboard');
```

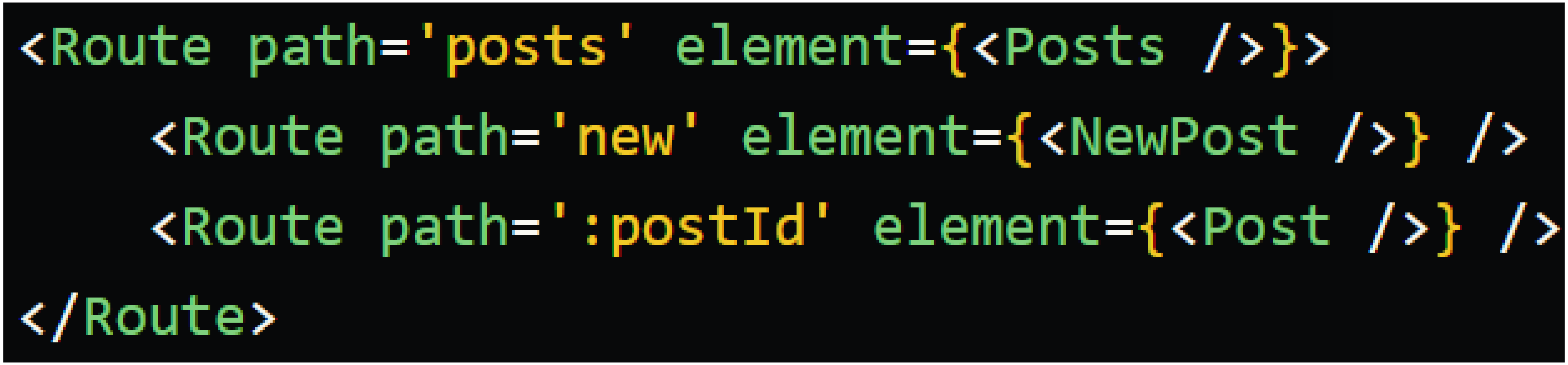
- Get route parameter

```
const { heroId } = useParams();
```

Route configured in the router:

```
<Route path="/hero/:heroId" component={SingleHero} />
```

Nested Routes



```
<Route path='posts' element={<Posts />}>
    <Route path='new' element={<NewPost />} />
    <Route path=':postId' element={<Post />} />
</Route>
```