

Instrukcja użycia biblioteki React Router w aplikacji Cars

1. Zainstaluj bibliotekę React Router wpisując w konsoli: `npm install react-router-dom`
2. W osobnym pliku (np. `Routes.tsx`) zdefiniuj tablicę obiektów typu `RouteObject[]`, która będzie przechowywać ścieżki aplikacji i dopasowywać do nich odpowiednie komponenty (widoki). Zdefiniuj również ścieżkę główną (tzw. korzeń). Utwórz obiekt router przy użyciu instrukcji `createBrowserRouter`.

```
9  export const routes: RouteObject[] = [  
10    {  
11      path: "/",  
12      element: <App />,  
13      children: [  
14        {path: 'cars', element: <CarList />},  
15        // :id jest tzw 'root parameter', który należy odebrać w <CarDetails />  
16        // używamy do tego useParams  
17        {path: 'cars/:id', element: <CarDetails />},  
18        {path: 'edit/:id', element: <CarForm />},  
19        {path: 'not-found', element: <NotFound />},  
20        // za każdym razem jak będzie niepoprawny adres url odeśle nas do <NotFound />  
21        {path: '*', element: <Navigate replace to='/not-found' />}  
22      ],  
23    },  
24  ],  
25  
26  export const router = createBrowserRouter(routes);
```

3. W celu użycia zdefiniowanego routera, należy go dostarczyć w całej aplikacji (pliku `index.tsx`). `<RouterProvider router={router} />` służy do uruchomienia systemu routingu (trasowania) w aplikacji React, przy użyciu konfiguracji tras dostarczonej w obiekcie `router`.

```
9  const root = ReactDOM.createRoot(  
10    document.getElementById('root') as HTMLElement  
11  );  
12  root.render(  
13    <RouterProvider router={router} />  
14  );
```

4. Do głównego komponentu `<App>` dodaj komponent `<Outlet>`. Komponent ten będzie wyświetlał inne komponenty w zależności od ścieżki. `<Outlet>` działa jako „miejsce na renderowanie” dla zagnieżdżonych tras wewnątrz trasy nadrzędnej. Jest to specjalne miejsce, gdzie React Router automatycznie wstawia komponent odpowiadający aktualnie dopasowanej zagnieżdżonej trasie.

```

1  import React from 'react';
2  import './App.css';
3  import { Outlet } from 'react-router-dom';
4  import NavBar from './NavBar';
5
6  function App() {
7    return (
8      <>
9        <NavBar />
10       <Outlet />
11     </>
12   );
13 }
14
15 export default App;

```

5. Stwórz menu w celu nawigacji po aplikacji. Dodaj elementy w pliku Routes.tsx jeśli zajdzie taka potrzeba. W nawigacji skorzystaj z któregoś z elementów: <Link>, <NavLink>, <Navigate>.