

import {createBrowserRouter, Navigate} from "react-router-dom";

This JavaScript line imports `createBrowserRouter` for setting up application-wide routing and `Navigate` for programmatic redirection within React Router (v6+), allowing you to define routes and easily redirect users based on conditions or actions in your web application. It's used to configure your app's navigation structure and conditionally send users to different pages or paths.

`createBrowserRouter`: Setting Up Routes

- **Purpose:** Creates a history object for typical browser navigation (using the History API) and enables powerful data APIs (loaders, actions).
- **Usage:** You pass an array of route objects (defining paths and elements to render) to it, then wrap your app with `<RouterProvider router={router}>..`

javascript

```
import { createBrowserRouter, RouterProvider } from 'react-router-dom';
import { HomePage } from './pages/HomePage';
import { AboutPage } from './pages/AboutPage';

const router = createBrowserRouter([
  { path: '/', element: <HomePage /> },
  { path: '/about', element: <AboutPage /> },
]);

function App() {
  return <RouterProvider router={router} />;
}
```

`Navigate`: Programmatic Redirection

- **Purpose:** A component that renders a <100ms 301 redirect when mounted, useful for authenticated routes or after form submissions.
- **Usage:** Place it inside a component to redirect to a specific `to` path; use `replace` to avoid adding to history (e.g., for auth).

javascript

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

function ProtectedRoute({ children }) {
  const isAuthenticated = checkAuthStatus(); // Your auth logic
  if (!isAuthenticated) {
    return <Navigate to="/login" replace />; // Redirect if not logged in
  }
  return children;
}

// In your router:
{ path: '/dashboard', element: <ProtectedRoute><Dashboard /></ProtectedRoute> }
```

Revision #4

Created 29 October 2025 02:43:39 by AI API

Updated 9 December 2025 11:42:07 by AI Channel