

createBrowserRouter in react

`createBrowserRouter`

in React Router (v6.4+) **creates a router using a data-driven approach (array of route objects) instead of JSX**, enabling powerful features like data loading (loaders) and mutations (actions), and is the recommended modern way for web apps by managing history via the HTML5 History API for clean URLs. You define routes as objects with `path` and `element` (or `children`) and pass them to `createBrowserRouter`, then wrap your app with the `RouterProvider` component.

How to Use `createBrowserRouter`

1. Install React Router:

bash

2.

```
npm install react-router-dom  
# or  
yarn add react-router-dom
```

3. Define Your Routes (e.g., in `router.js` or `index.js`):

javascript

4.

```
import { createBrowserRouter } from 'react-router-dom';  
import Root from './routes/Root'; // Your main layout component  
import Home from './routes/Home';  
import About from './routes/About';  
import ErrorPage from './routes/ErrorPage';  
  
const router = createBrowserRouter([  
  {  
    path: "/",  
    element: <Root />,  
    errorElement: <ErrorPage />,  
    children: [ // Nested routes  
      { index: true, element: <Home /> }, // Renders <Home> at '/'  
      { path: "about", element: <About /> },  
      { path: "users/:userId", element: <UserProfile /> }, // Dynamic route  
    ],  
  },  
]);
```

```
});
```

```
export default router;
```

- **path**: The URL segment. `index: true` is for the default child route.
- **element**: The component to render for that path.
- **children**: For nested routes, uses `<Outlet />` in the parent.

5. Provide the Router (in `index.js` or `App.js`):

javascript

```
import React from 'react';
import ReactDOM from 'react-dom/client';
import { RouterProvider } from 'react-router-dom';
import router from './router'; // Your router config

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <RouterProvider router={router} />
  </React.StrictMode>,
);
```

Key Advantages (vs. `<BrowserRouter>` /JSX Routes)

- **Data Layer**: Supports `loader` functions for fetching data before rendering, ensuring components get data immediately.
- **Data Mutations**: Supports `action` functions for POST/PUT/DELETE requests, improving data flow.
- **Flexibility**: Configure routes as JS objects (data) instead of JSX, separating concerns.
- **Future-Proof**: Recommended by React Router as the modern direction

Revision #2

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

Updated 9 December 2025 11:52:59 by AI Channel